

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



MEMORIA DEL TRABAJO FINAL

**Módulo de búsqueda, seguimiento y  
decorrelación para un sistema GPS sobre  
FPGA**

**Autor:**  
**Ing. Facundo Larosa**

Director:  
Ing. Nicolás Álvarez (FIUBA /UNSAM)

Jurados:  
Esp.Ing. Pedro Martos (FIUBA)  
Ing. Edgardo Comas (CITEDEF/UTN FRBA)  
Ing. Octavio Alpago (FIUBA)

*Este trabajo fue realizado en las Ciudad Autónoma de Buenos Aires, entre Abril  
de 2016 y Noviembre de 2016.*



## *Resumen*

El desarrollo de equipos de posicionamiento satelital es un área de gran importancia económica y estratégica en actividades tales como la agricultura, la navegación segura, la defensa, etc. La proliferación de nuevas constelaciones de satélites y señales de navegación, así como la utilización de los receptores en entornos de baja señal impulsa el desarrollo de nuevas implementaciones. Es así, que el desarrollo de arquitecturas flexibles representa una gran ventaja a la hora de estudiar y ensayar este tipo de sistemas, adaptándolos a diferentes necesidades.

El trabajo planteado abarca el estudio, diseño, simulación y pruebas de un módulo de búsqueda y un módulo de seguimiento sobre FPGA que serán utilizados a futuro para la implementación de un receptor GPS definido por software.



## *Agradecimientos*

A mi familia, que es siempre mi pilar en todo.

A Ariel Lutenberg, por la confianza depositada en mí y por su labor continua en la difusión de la enseñanza y la aplicación de los sistemas embebidos.

A mi director de proyecto final, Nicolás Álvarez, por su ayuda y paciencia ¡Ojalá podamos seguir trabajando juntos en estos temas!



# Índice general

|  |            |
|--|------------|
| <b>Resumen</b>   | <b>III</b> |
| <b>1. Introducción General</b>   | <b>1</b>   |
| 1.1. Introducción  | 1          |
| 1.1.1. Descripción general del sistema GPS   | 1          |
| 1.2. Objetivos   | 3          |
| <b>2. Introducción Específica</b>  | <b>5</b>   |
| 2.1. Generación de señal GPS   | 5          |
| 2.1.1. Esquema general   | 5          |
| 2.1.2. Generación y características del código CA  | 5          |
| 2.1.3. Efecto del medio sobre la señal GPS   | 9          |
| 2.2. Método de búsqueda serial   | 11         |
| 2.2.1. Caso general  | 14         |
| 2.2.2. Caso particular: Código C/A local en fase con código C/A de la señal de entrada                                     | 16         |
| 2.2.3. Caso particular: Oscilador local con igual frecuencia de la señal de frecuencia intermedia                          | 17         |
| 2.2.4. Caso particular: Coincidencia de fase de C/A y desplazamiento de frecuencia de la señal local y la señal de entrada | 18         |
| 2.2.5. Conclusiones  | 19         |
| <b>3. Diseño e Implementación</b>  | <b>21</b>  |
| 3.1. Introducción  | 21         |
| 3.1.1. Consideraciones generales   | 21         |
| 3.1.2. Consideraciones de diseño   | 22         |
| 3.2. Generador de código C/A   | 24         |
| 3.2.1. Interfaz  | 24         |
| 3.2.2. Operación   | 24         |
| 3.2.3. Pruebas   | 25         |
| 3.3. Oscilador controlado numéricamente (NCO)  | 26         |
| 3.3.1. Interfaz  | 26         |
| 3.3.2. Operación   | 26         |
| 3.3.3. Pruebas   | 27         |
| 3.4. Módulo de entrada   | 28         |
| 3.4.1. Interfaz  | 28         |
| 3.4.2. Operación   | 28         |
| 3.4.3. Pruebas   | 29         |
| 3.5. Módulo de memoria   | 30         |
| 3.5.1. Interfaz  | 30         |
| 3.5.2. Operación   | 30         |
| 3.5.3. Pruebas   | 33         |
| 3.6. Módulo de generación de señal GPS local   | 36         |

|  |           |
|--|-----------|
| 3.6.1. Interfaz . . . . .                                  | 36        |
| 3.6.2. Operación . . . . .                                 | 37        |
| 3.6.3. Pruebas . . . . .                                   | 38        |
| 3.7. Módulo de búsqueda . . . . .                          | 40        |
| 3.7.1. Interfaz . . . . .                                  | 40        |
| 3.7.2. Operación . . . . .                                 | 41        |
| 3.7.3. Pruebas . . . . .                                   | 43        |
| 3.8. Módulo de seguimiento . . . . .                       | 45        |
| 3.8.1. Interfaz . . . . .                                  | 45        |
| 3.8.2. Operación . . . . .                                 | 45        |
| 3.8.3. Pruebas . . . . .                                   | 49        |
| <b>4. Ensayos y Resultados</b>                             | <b>53</b> |
| 4.1. Validación de generación de señales réplica . . . . . | 53        |
| 4.2. Validación de módulo de búsqueda . . . . .            | 55        |
| 4.3. Validación de módulo de seguimiento . . . . .         | 57        |
| 4.4. Recursos utilizados . . . . .                         | 58        |
| <b>5. Conclusiones</b>                                     | <b>59</b> |
| 5.1. Conclusiones generales . . . . .                      | 59        |
| 5.2. Próximos pasos . . . . .                              | 59        |
| <b>Bibliografía</b>  | <b>61</b> |



# Índice de figuras

|  |    |
|--|----|
| 1.1. Segmento de control del sistema GPS [7] . . . . .   | 2  |
| 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 . . . . .  | 7  |
| 2.3. Autocorrelación del código C/A . . . . .  | 7  |
| 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 . . . . .   | 9  |
| 2.6. Desviación de la frecuencia central por efecto Doppler . . . . .  | 10 |
| 2.7. Desviación de la frecuencia central por efecto Doppler . . . . .  | 11 |
| 2.8. Esquema general diagrama de búsqueda . . . . .  | 12 |
| 2.9. Gráfico de $C[n_{ol}, \Delta f_{dl}]$ . . . . .   | 16 |
| 2.10. Gráfico de $C[n_{ol} = n_0, \Delta f_{dl}]$ . . . . .  | 17 |
| 2.11. Gráfico de $C[n_{ol}, \Delta f_{dl} = \Delta f_d]$ . . . . .   | 18 |
| 3.1. Diagrama en bloques del <i>frontend</i> . . . . .   | 21 |
| 3.2. Diagrama en bloques general de una FSMD . . . . .   | 23 |
| 3.3. Interfaz del módulo de generación de código C/A . . . . .   | 24 |
| 3.4. Diagrama en bloques del generador de código CA . . . . .  | 25 |
| 3.5. Generación de código C/A (salida <i>caCode</i> para el satélite 1 . . . . .   | 25 |
| 3.6. Interfaz del NCO . . . . .  | 26 |
| 3.7. Diagrama en bloques del NCO . . . . .   | 27 |
| 3.8. Oscilograma de las señales de salida del NCO: seno (superior) y coseno (inferior) . . . . .                                 | 28 |
| 3.9. Valores medidos de frecuencia de salida en función del incremento de fase . . . . .   | 28 |
| 3.10. Interfaz del módulo de entrada . . . . .   | 29 |
| 3.11. Diagrama en bloques del módulo de entrada . . . . .  | 30 |
| 3.12. Diagrama de señales del módulo de entrada . . . . .  | 30 |
| 3.13. Interfaz del módulo de memoria . . . . .   | 31 |
| 3.14. Diagrama en bloques del módulo de memoria . . . . .  | 32 |
| 3.15. Comienzo de la prueba de escritura continua . . . . .  | 33 |
| 3.16. Comienzo del ciclo de grabación . . . . .  | 33 |
| 3.17. Fin del ciclo de grabación de la memoria 0, comienzo de grabación de la memoria 1 . . . . .                                | 34 |
| 3.18. Escritura alternada de las memorias 0 y 1 . . . . .  | 34 |
| 3.19. Fin de escritura de la memoria 1 y comienzo de lectura de memoria 0 . . . . .  | 34 |
| 3.20. Lectura de memoria 0 . . . . .   | 35 |
| 3.21. Escritura alternada de las memorias 0 y 1 en ausencia de nuevo comando de lectura . . . . .                                | 35 |

|  |    |
|--|----|
| 3.22. Lectura continua de la memoria 0 mientras la memoria 1 se está grabando . . . . .                              | 35 |
| 3.23. Detalle del cambio de escritura/lectura de la memoria 1 a la memoria 0 . . . . .                               | 36 |
| 3.24. Interfaz del módulo de generación de señal GPS local . . . . .   | 36 |
| 3.25. Diagrama en bloques del módulo de generación de señales local . . . . .  | 37 |
| 3.26. Inicio de la grabación de señales réplica . . . . .  | 38 |
| 3.27. Detalle de la inversión de fase de las señales réplica . . . . .   | 39 |
| 3.28. Fin de la grabación de las señales réplica . . . . .   | 39 |
| 3.29. Lectura de las señales réplica ( <i>initPhase=0</i> ) . . . . .  | 40 |
| 3.30. Lectura de las señales réplica ( <i>initPhase=1024</i> ) . . . . .   | 40 |
| 3.31. Detalle del cambio de dirección de lectura . . . . .   | 40 |
| 3.32. Interfaz del módulo de búsqueda . . . . .  | 41 |
| 3.33. Diagrama en bloques del módulo de búsqueda . . . . .   | 42 |
| 3.34. Comienzo de la inicialización del módulo de búsqueda . . . . .   | 43 |
| 3.35. Fin de la inicialización del módulo de búsqueda . . . . .  | 43 |
| 3.36. Inicio de generación de señal GPS local . . . . .  | 44 |
| 3.37. Inicio de la correlación de señal GPS local . . . . .  | 44 |
| 3.38. Fin de la correlación de señal GPS local . . . . .   | 45 |
| 3.39. Interfaz del módulo de seguimiento . . . . .   | 45 |
| 3.40. Interfaz del módulo de seguimiento . . . . .   | 46 |
| 3.41. Valores de correlación normalizados para los códigos Early, Prompt y Late (señal en fase) . . . . .            | 48 |
| 3.42. Valores de correlación normalizados para los códigos Early, Prompt y Late (señal atrasada) . . . . .           | 48 |
| 3.43. Diagrama en bloques (énfasis en los bloques utilizados para detectar la sincronía del código C/A) . . . . .    | 49 |
| 3.44. Diagrama en bloques (énfasis en los bloques utilizados para detectar la sincronía de la señal de FI) . . . . . | 50 |
| 3.45. Inicialización módulo de seguimiento . . . . .   | 50 |
| 3.46. Sincronización del código C/A . . . . .  | 51 |
| 3.47. Inicio de la acumulación . . . . .   | 51 |
| 3.48. Fin de la acumulación . . . . .  | 51 |
| 4.1. Diagrama esquemático del procedimiento de validación de las señales réplica . . . . .                           | 54 |
| 4.2. Gráfico de correlación en función de la fase del código C/A para $\Delta f_{dl} = 0$ . . . . .                  | 55 |
| 4.3. Gráfico de correlación en función de la frecuencia para $n_{0l} = 1021, 5$ . . . . .                            | 55 |
| 4.4. Diagrama esquemático del procedimiento de validación del módulo de búsqueda . . . . .                           | 56 |
| 4.5. Gráfico de correlación en función de la frecuencia para $n_{0l} = 5, 5$ . . . . .                               | 56 |
| 4.6. Gráfico de correlación en función de la fase de código C/A para $\Delta f_{dl} = -1000$ . . . . .               | 57 |
| 4.7. Gráfico de correlación en función de la fase de código C/A [chips] . . . . .                                    | 58 |

# Índice de Tablas

|   |    |
|---|----|
| 1.1. Planos orbitales . . . . .                                   | 3  |
| 2.1. Salidas . . . . .  | 8  |
| 3.1. Parámetros del proyecto . . . . .                            | 22 |
| 3.2. Interfaz - Módulo Generador de código C/A . . . . .          | 24 |
| 3.3. Interfaz - NCO . . . . .                                     | 26 |
| 3.4. Interfaz - Módulo de entrada . . . . .                       | 29 |
| 3.5. Interfaz - Módulo de memoria . . . . .                       | 31 |
| 3.6. Interfaz - Módulo de generación de señal GPS local . . . . . | 36 |
| 3.7. Interfaz - Módulo de búsqueda . . . . .                      | 41 |
| 3.8. Interfaz - Módulo de seguimiento . . . . .                   | 46 |
| 4.1. Recursos utilizados - Módulo de búsqueda . . . . .           | 58 |
| 4.2. Recursos utilizados - Módulo de seguimiento . . . . .        | 58 |



*A mi maestro, Cristian Arrieta*



# 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*).

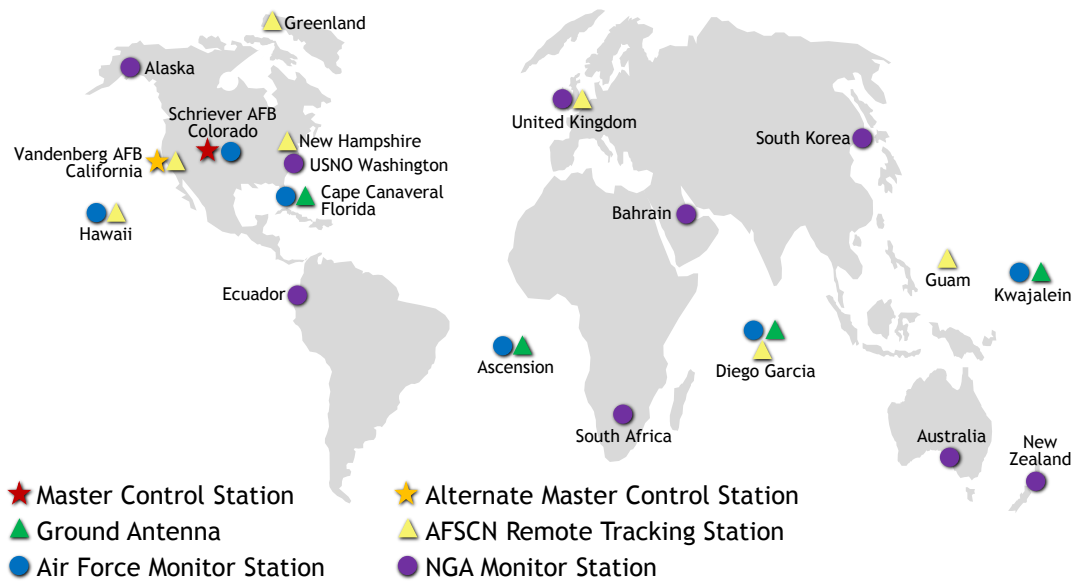
Los servicios de uso civil estaban originalmente sometidos a una degradación intencional de la precisión de sus señales conocida como 'disponibilidad selectiva' (SA, del inglés *selective availability*). La disponibilidad selectiva fue eliminada definitivamente del sistema a través de un decreto de la administración Clinton el 1 de mayo del año 2000 [5]. Los nuevos satélites del Bloque III a ser 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.

El 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.

El 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  $55^\circ$ ) respecto del ecuador terrestre. Los planos se encuentran separados unos  $60^\circ$  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

## GPS Control Segment



Updated April 2016

FIGURA 1.1: Segmento de control del sistema GPS [7]

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 MATLAB en base a datos extraídos de la página oficial de la Federación Rusa (<https://www.glonass-iac.ru/en/GLONASS/>).

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  $180^\circ$  se observan superpuestos debido a la perspectiva. En la figura 1.3 se observa desde una perspectiva lateral, permitiendo apreciar los planos de  $80^\circ$  y  $260^\circ$  (separados  $180^\circ$ ).

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



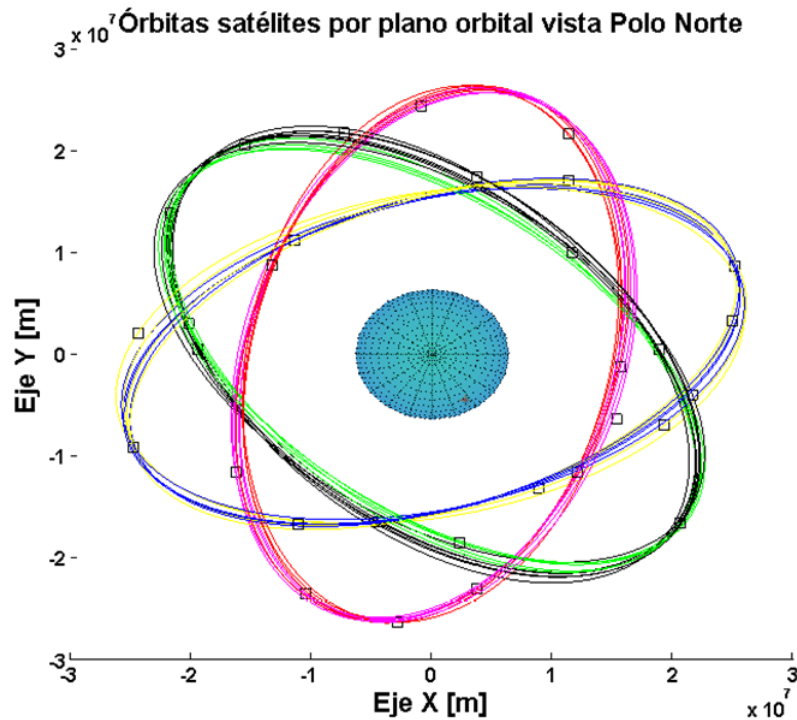


FIGURA 1.2: Órbitas de los satélites del sistema GPS (vista superior) en coordenadas ECEF

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    |

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.

El segmento de usuario está compuesto 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 [4].

## 1.2. Objetivos

Se planteó como objetivos de este trabajo el diseño y simulación de módulos descritos en VHDL destinados a procesar una señal GPS en banda L1 (simulada).

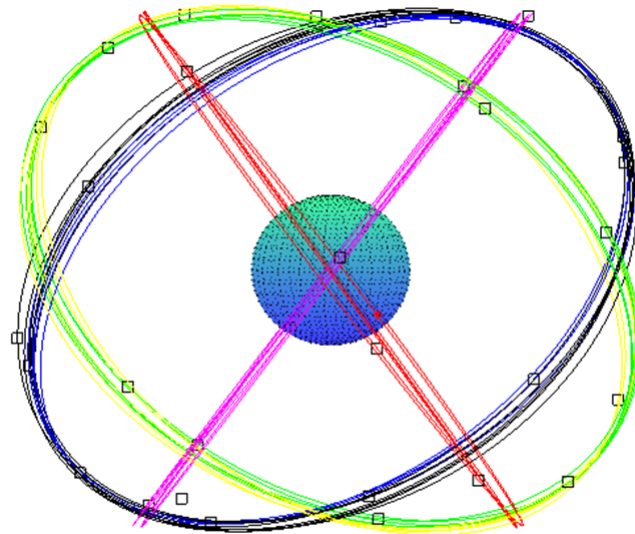


FIGURA 1.3: Órbitas de los satélites del sistema GPS (lateral)

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.

Cómo objetivos secundarios se planteó 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.

## Capítulo 2

# Introducción Específica

El objetivo de esta sección es tratar 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 (en general, se reserva 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.

En la figura 2.1 se observa en primer lugar la operación de mezcla entre el código C/A y el mensaje (estrictamente, siendo una señal digital la operación es una XOR) y luego de la conformación del mensaje ampliado 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 CA

Los códigos C/A pertenecen a la familia de códigos pseudoaleatorios de Gold [6]. El esquema de generación de los códigos de Gold se muestra en la figura 2.2.

En la figura 2.2 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.

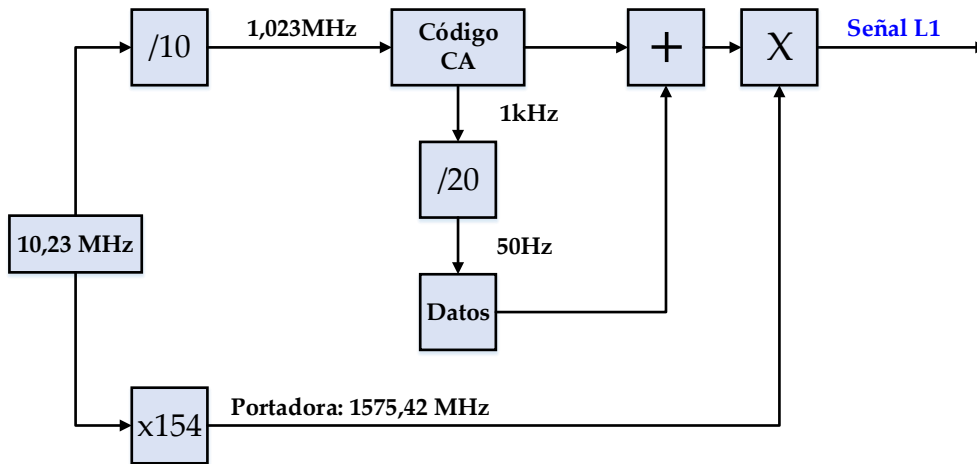


FIGURA 2.1: Esquema general de generación de señal GPS donde se indican las frecuencias específicas de cada señal involucrada

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:

- $\otimes$  :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  $x[n] = x[n + N]$ , lo que usualmente conoce como 'aritmética modular de módulo  $N$ '.

En el caso del código CA de un satélite  $i$ -ésimo, su autocorrelación normalizada 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:

- $M$  :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

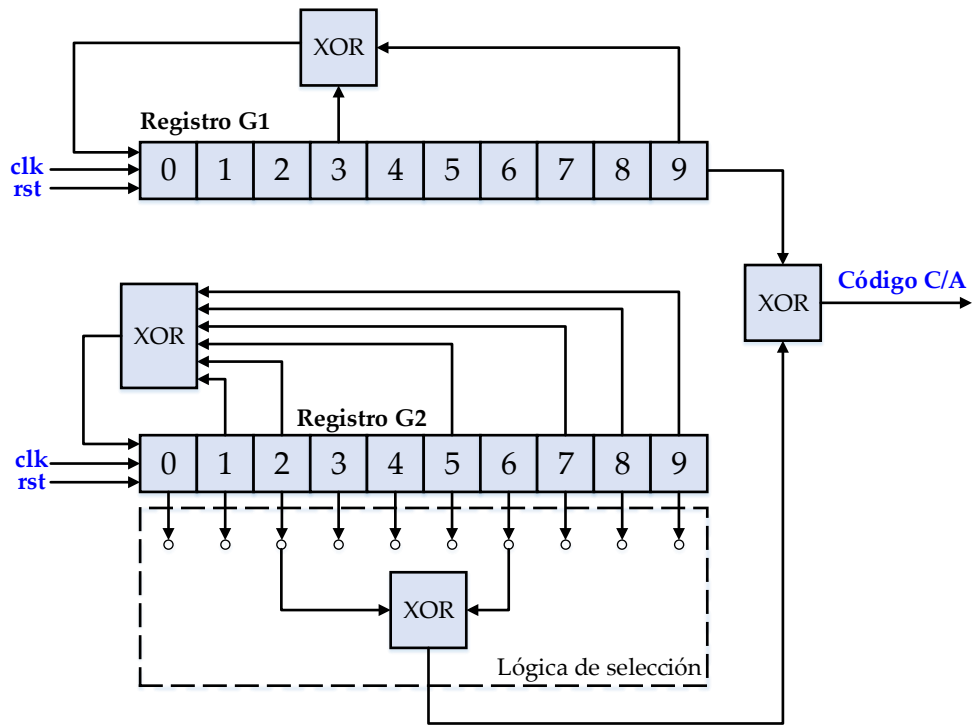


FIGURA 2.2: Esquema de generación de los códigos de Gold

de autocorrelación para el código C/A del satélite 1. El tiempo discreto  $n$  está escalado para indicar desplazamiento en chips.

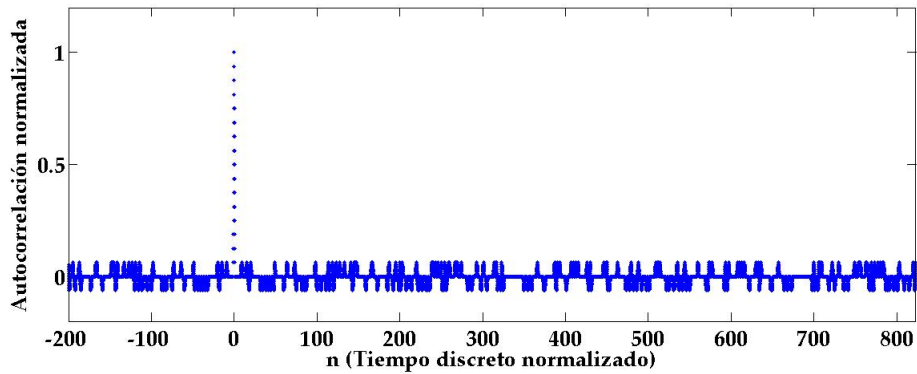


FIGURA 2.3: Autocorrelación del código C/A

En la figura 2.4 se muestra un detalle de la función  $c_{CAi}[n]$  en torno a  $n = 0$ , donde se observa que se desvanece linealmente al desplazarse un chip a derecha o izquierda.

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

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

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             |

Donde se ha conservado la normalización para facilitar la comparación. 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 provenientes de diferentes satélites y para sincronizar la señal de entrada para un satélite determinado de forma local.

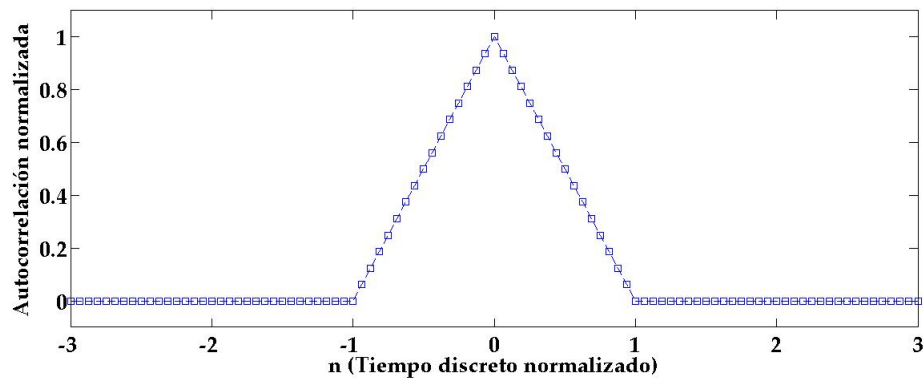


FIGURA 2.4: Autocorrelación del código C/A (detalle en torno a  $n=0$ )

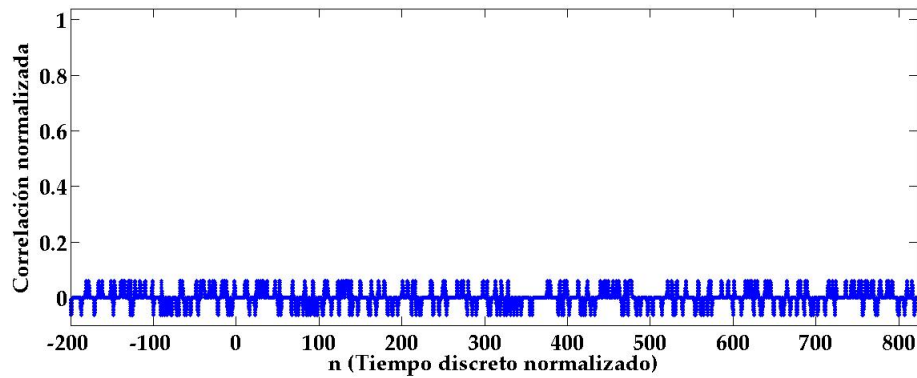


FIGURA 2.5: Correlación cruzada del código C/A del satélite 1 y 2

### 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:

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

donde:

- $\lambda$ : Longitud de onda observada por el receptor [m]
- $\lambda_0$ : Longitud de onda emitida por el satélite [m]
- $c$ : Velocidad de la luz [m/s]

- $\vec{k}$ : Versor en la dirección de la línea de vista receptor-satélite
- $\vec{v}$ : Velocidad del satélite relativa al receptor

Se realizaron simulaciones de la cinemática de los satélites respecto de un receptor estacionario (en reposo relativo a la Tierra) utilizando MATLAB. El receptor se ubicó en las coordenadas  $-34,603333^\circ S$ ,  $-58,381667^\circ O$ . En base a las ecuaciones de posicionamiento indicadas en [12] se discretizó la posición para cada satélite. 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*)<sup>1</sup> 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  $\Delta f = \pm 4k Hz$  respecto de la frecuencia central.

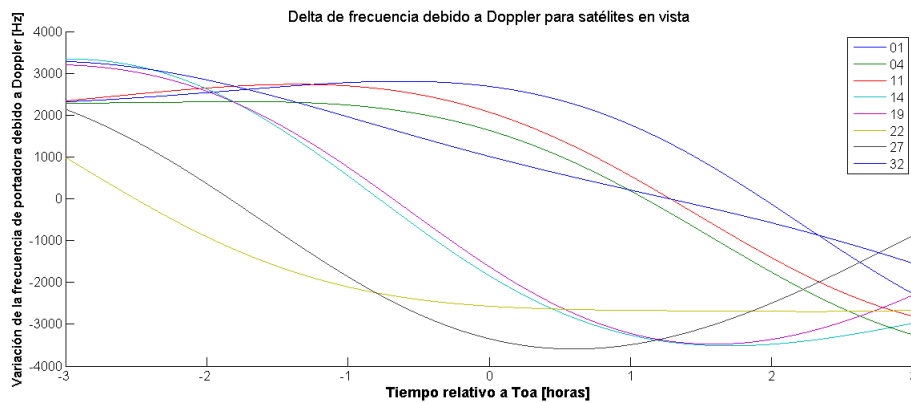


FIGURA 2.6: Desviación de la frecuencia central por efecto Doppler

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  $1 \frac{Hz}{s}$ .

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



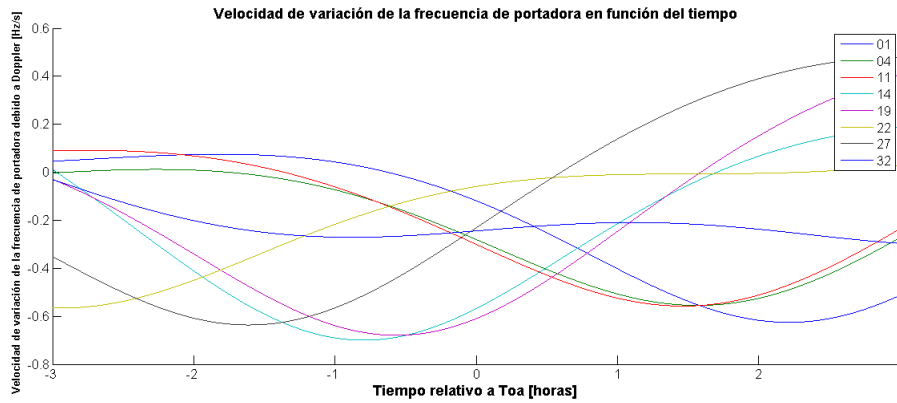


FIGURA 2.7: Desviación de la frecuencia central por efecto Doppler

## 2.2. Método de búsqueda serial

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

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

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]
- $\Delta f_d$ : Desplazamiento de la frecuencia central debida al efecto Doppler [Hz]
- $T_s$ : Tiempo entre muestras, inversa de la frecuencia de muestreo [s]
- $\phi_0$ : Fase inicial de la portadora [adimensional]

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 ( $\phi_0$ )

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  $\phi_0$ .

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 ( $\Delta f_d$ ). 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.

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  $\Omega$  en torno a la frecuencia intermedia central desplazadas una cantidad  $\Delta f_{dl}$  que se asimila al corrimiento por efecto Doppler.

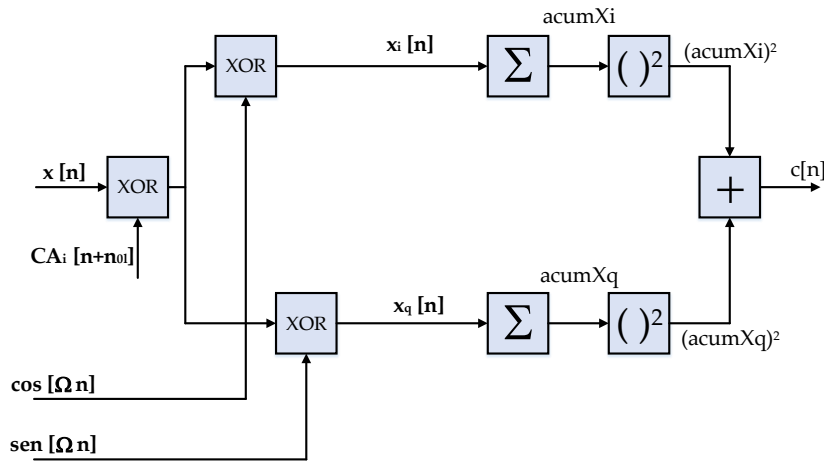


FIGURA 2.8: Esquema general diagrama de búsqueda

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  $CA$  del satélite que se pretende buscar ( $CA_j[n]$ ) y las componentes en cuadratura del oscilador local obteniendo:

$$x_{iCA}[n] = CA_i[n + n_0] \cdot CA_j[n + n_{0j}] \cdot \cos[2\pi(f_i + f_d)T_s n + \phi_0] \cos[2\pi(f_i + \Delta f_{dl})T_s n] \quad (2.6)$$

$$x_{qCA}[n] = CA_i[n + n_0] \cdot CA_j[n + n_{0j}] \cdot \cos[2\pi(f_i + f_d)T_s n + \phi_0] \sin[2\pi(f_i + \Delta f_{dl})T_s n] \quad (2.7)$$

donde:

- $n_{0l}$ : Desplazamiento de la copia del código C/A local
- $\Delta f_{dl}$ : Desplazamiento en frecuencia del oscilador local

Utilizando las identidades trigonométricas para el producto de funciones armónicas [13] se obtiene:

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

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

Las señales descritas 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:

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

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

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

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

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 (debido a que los acumuladores tienen una respuesta de tipo pasabajos), de tal forma que:

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

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

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  $\Delta f_{dl}$  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 ( $\Delta f_{dl}$ ). 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.

### 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:

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

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

Las ecuaciones 2.13 y 2.14 pueden reescribirse de forma más compacta utilizando el enfoque de señales complejas [11].

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:

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

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

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:

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

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

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

$$|c[n_{ol}, \Delta f_{dl}]|^2 = AcumX_i^2[n_{ol}, \Delta f_{dl}] + AcumX_q^2[n_{ol}, \Delta f_{dl}] = C[n_{ol}, \Delta f_{dl}] \quad (2.25)$$

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 [9] 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  $n_{0l} \in [0, 16367]$ . La función de salida se grafica en la figura 2.9.

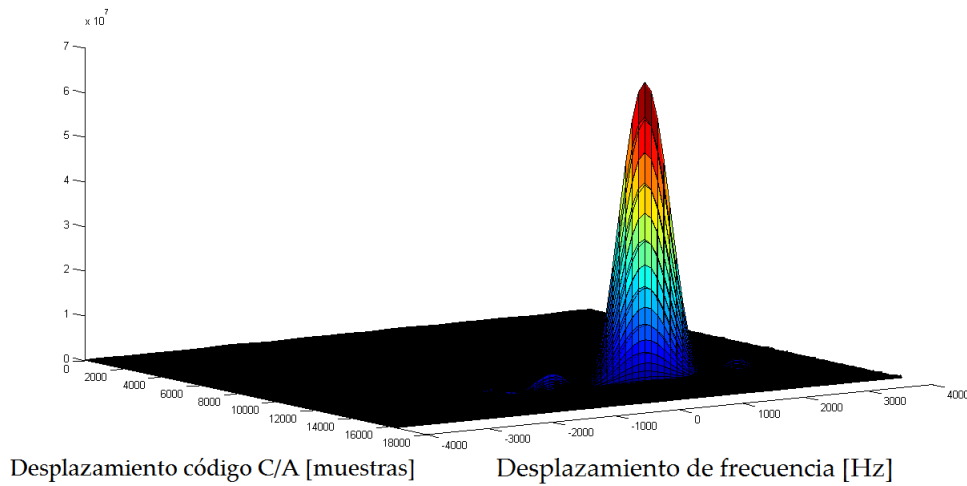


FIGURA 2.9: Gráfico de  $C[n_{0l}, \Delta f_{dl}]$

### 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  $n_0 = n_{0l}$ , de forma tal que la expresión de  $C[n_{0l}, \Delta f_{dl}]$  se reduce a:

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

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

Es decir, en este caso se está integrando en un intervalo de  $N$  muestras una señal compleja cuya única componente de frecuencia vale  $2\pi(\Delta f_d - \Delta f_{dl})$ . 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  $\Delta f_d = 1000Hz$ .

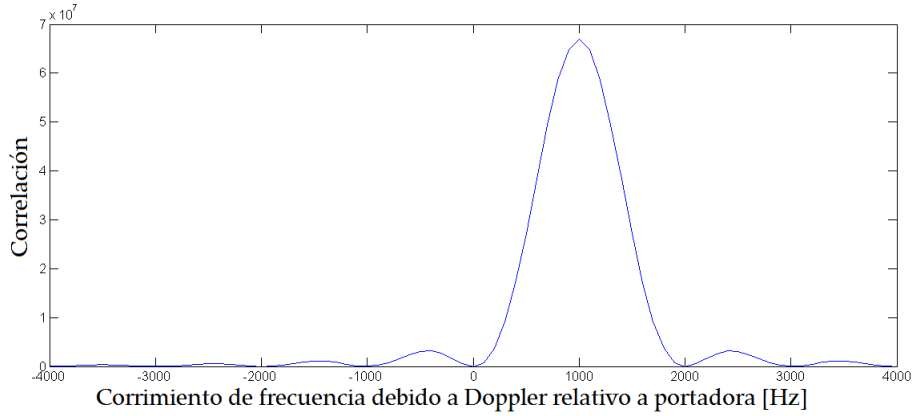


FIGURA 2.10: Gráfico de  $C[n_{ol} = n_0, \Delta f_{dl}]$

### 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  $C[n_{ol}, \Delta f_{dl}]$  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  $\Delta n$  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:

$$C[n_{0l}, \Delta f_{dl} = \Delta f_d] = 1/4 c_{CAi}^2[\Delta n] \quad (2.32)$$

donde:

$c_{CAi}[\Delta n]$ : 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. Para facilitar la interpretación se ha normalizado el eje para indicar la variable directamente en chips.

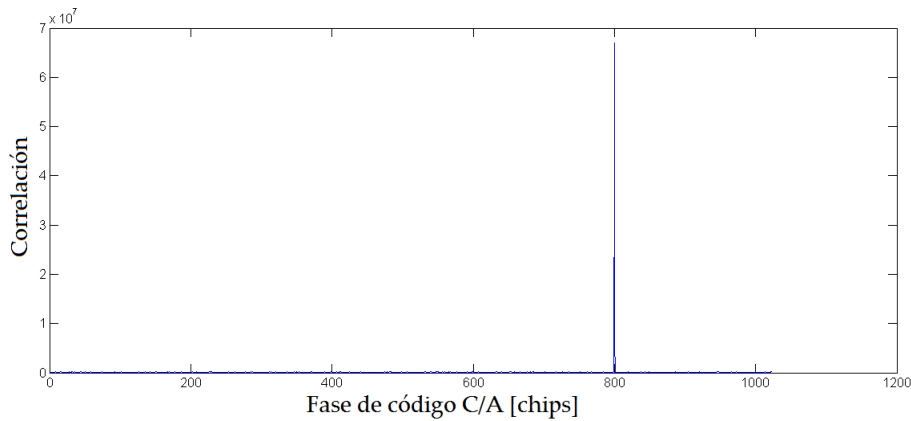


FIGURA 2.11: Gráfico de  $C[n_{0l}, \Delta f_{dl} = \Delta f_d]$

#### 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  $n_0 = n_{0l}$  y  $\Delta f_d = \Delta f_{dl}$  simultáneamente, de forma tal que la expresión de  $C[n_{0l}, \Delta f_{dl}]$  se reduce a:

$$C[n_{0l} = n_0, \Delta f_{dl} = \Delta f_d] = \left(\frac{N}{2}\right)^2 \quad (2.33)$$

En este caso, estamos en presencia del máximo de la función  $C[n_{0l}, \Delta f_{dl}]$  cuyo valor será una constante dependiendo solamente de la cantidad de muestras en el intervalo de acumulación considerado ( $N$ ).



### 2.2.5. Conclusiones

En este capítulo, se presentaron las ecuaciones y métodos básicos para comprender la generación de las señales GPS, así como su detección por medio del método de búsqueda serial. En el capítulo siguiente, se estudiarán diferentes módulos digitales que tienen por objeto realizar las operaciones planteadas en este capítulo.



## Capítulo 3

# Diseño e Implementación

En este capítulo se describen las interfaces, estructura, operación y señales generadas por los diferentes módulos que conforman el proyecto.

### 3.1. Introducción

#### 3.1.1. Consideraciones generales

En este capítulo, se explica el diseño e implementación de los módulos destinados a procesar la señal de salida de un *frontend* de GPS. 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.

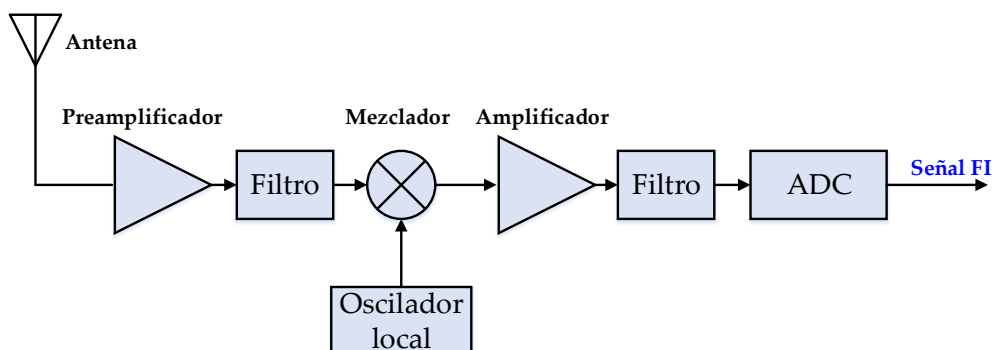


FIGURA 3.1: Diagrama en bloques del *frontend*

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 de Maxim Integrated Inc. o el SE4150 de Skyworks Inc.

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 [8]<sup>1</sup> 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 (consistente en una componente de código C/A y una portadora en frecuencia intermedia) 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. 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*) [2]. Una máquina de estados finita con camino de datos combina una máquina de estados finita y circuitos secuenciales convencionales. En la figura 3.2 se indica un diagrama en bloques de una FSMD genérica.

En la figura 3.2 se observan las dos partes principales de una FSMD. Por un lado, la máquina de estados (FSMD), 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 especifican las operaciones a realizar por los circuitos 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.

El ciclo de trabajo utilizado en este trabajo para diseñar, implementar y simular las FSMD se describe a continuación:

1. Definir el algoritmo a implementar

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

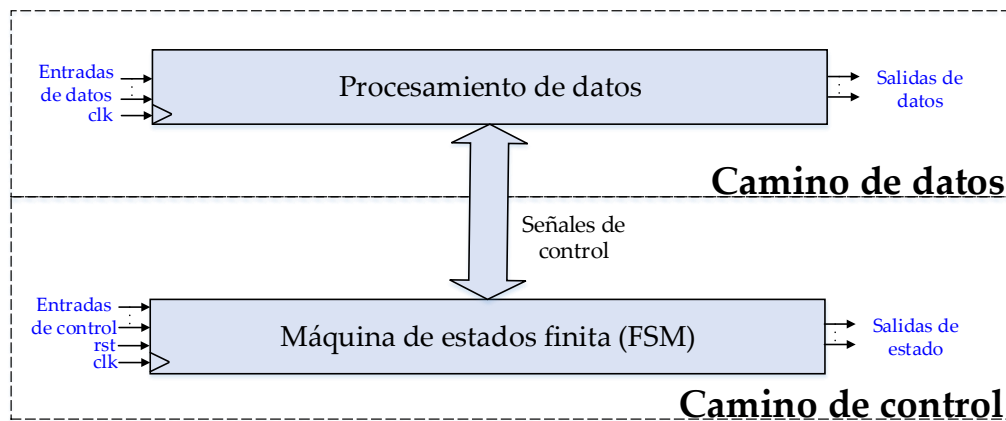


FIGURA 3.2: Diagrama en bloques general de una FSM

2. Definir la interfaz (entradas y salidas) de la FSM
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
6. Diseñar la FSM
7. Codificar el diseño en VHDL
8. Diseñar y realizar las pruebas (*testbenches*)

La metodología descrita, ha demostrado ser robusta para realizar la implementación de los módulos planteados. 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: en primer lugar, porque puede generarse con un error menor que 0,03% utilizando los módulos de control digitales de reloj (*DCM, Digital Clock Manager*) 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*. En segundo lugar, 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, ya que se simplifica la generación de las señales internas, adaptando las frecuencias de algunos módulos por medio de circuitos *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. Generador de código C/A

### 3.2.1. Interfaz

En la figura 3.3 se indica la interfaz del generador de código C/A, estando su descripción detallada en la tabla 3.2.

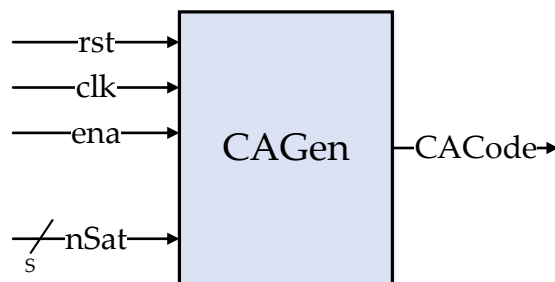


FIGURA 3.3: Interfaz del módulo de generación de código C/A

TABLA 3.2: 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 |

En las figuras donde se ilustran 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 3.3, donde *S* representa la cantidad de bits de la señal, expresada de forma genérica. Esta notación se utilizará consistentemente en todo el trabajo.

### 3.2.2. Operación

Este módulo genera los códigos C/A del satélite que se indica en su entrada *nSat*. La entrada *nSat* se interpreta 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 3.4 se muestra un diagrama en bloques del generador. El esquema 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.

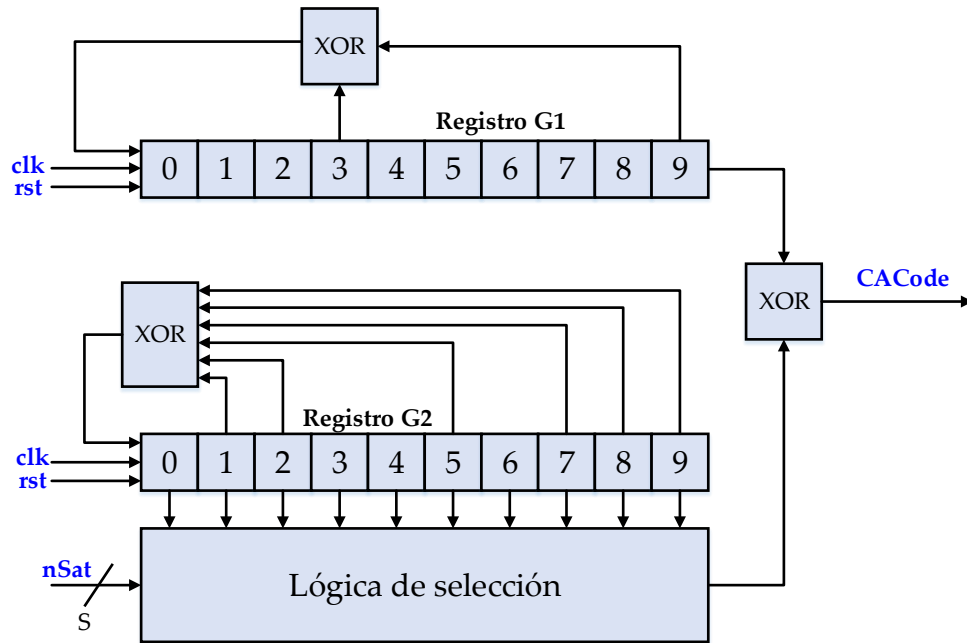


FIGURA 3.4: Diagrama en bloques del generador de código CA

### 3.2.3. 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 C/A generadas separadamente por medio de un *script* de MATLAB. En la figura 3.5 se muestra a modo de ejemplo la generación de una señal de código C/A para el satélite 1 (salida *CA Code*).

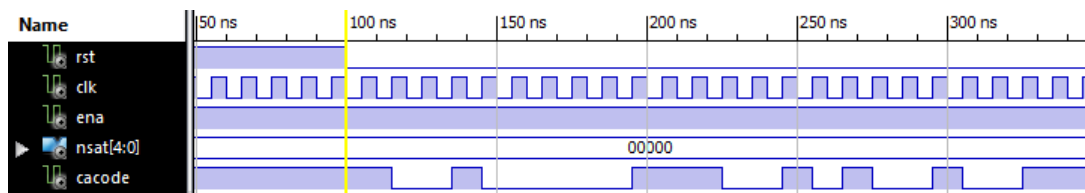


FIGURA 3.5: Generación de código C/A (salida *caCode* para el satélite 1)

### 3.3. Oscilador controlado numéricamente (NCO)

#### 3.3.1. Interfaz

En la figura 3.6 se indica la interfaz del NCO, estando su descripción detallada en la tabla 3.3.

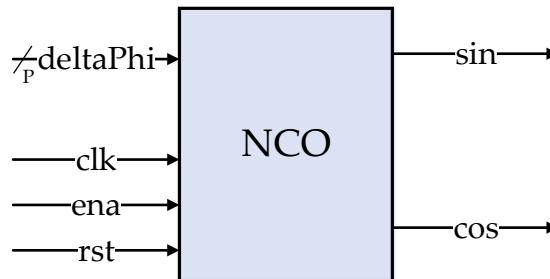


FIGURA 3.6: Interfaz del NCO

TABLA 3.3: 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      |

#### 3.3.2. Operación

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.

En la figura 3.7 se muestra un diagrama en bloques del NCO. La frecuencia del NCO está dada por [10]:

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

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

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).

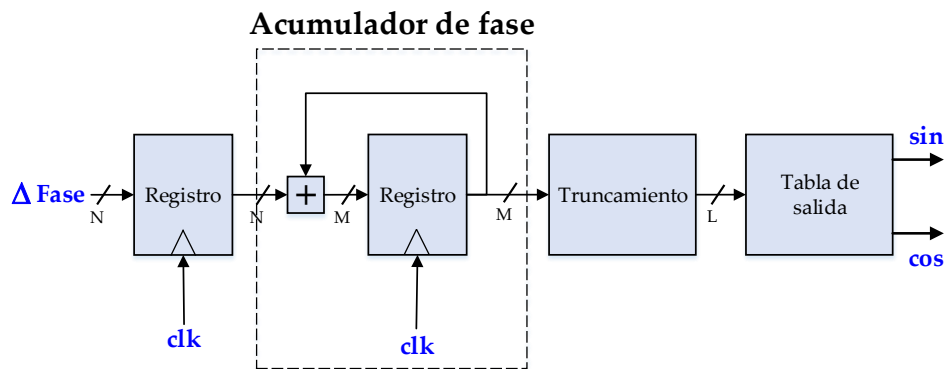


FIGURA 3.7: Diagrama en bloques del NCO

### 3.3.3. Pruebas

Se comprobó el correcto funcionamiento del NCO instanciándolo en la FPGA. La frecuencia de salida se midió por medio de un osciloscopio digital Hantek MSO5202D (ancho de banda de 200 MHz). En la figura 3.8 se observa un oscilograma de las señales seno y coseno medidas.

A fin de comprobar la correcta variación de la frecuencia de salida en función de la señal de control, se instanció el módulo NCO agregando a su entrada un bloque combinacional (que en el kit de desarrollo se controla por medio de pulsadores y teclas) el cual introduce un número signado de 9 bits (-256 a 255). Este número permite variar la frecuencia de salida en torno al valor de referencia (la frecuencia de FI 4,092 MHz) en pasos de 20 Hz aproximadamente. Se relevó la frecuencia según la palabra de control a la entrada y se realizó el gráfico que se muestra en la figura 3.9, comprobándose la linealidad entre el incremento de fase  $\Delta\phi$  y la frecuencia de salida  $f_o$ .

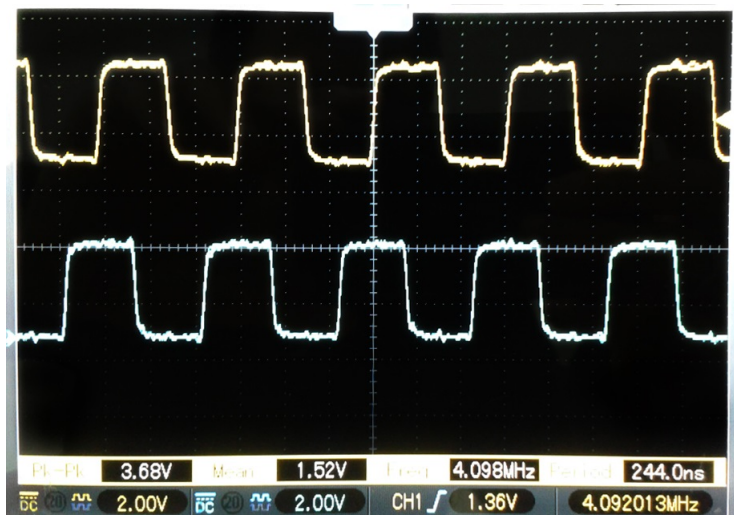


FIGURA 3.8: Oscilograma de las señales de salida del NCO: seno (superior) y coseno (inferior)

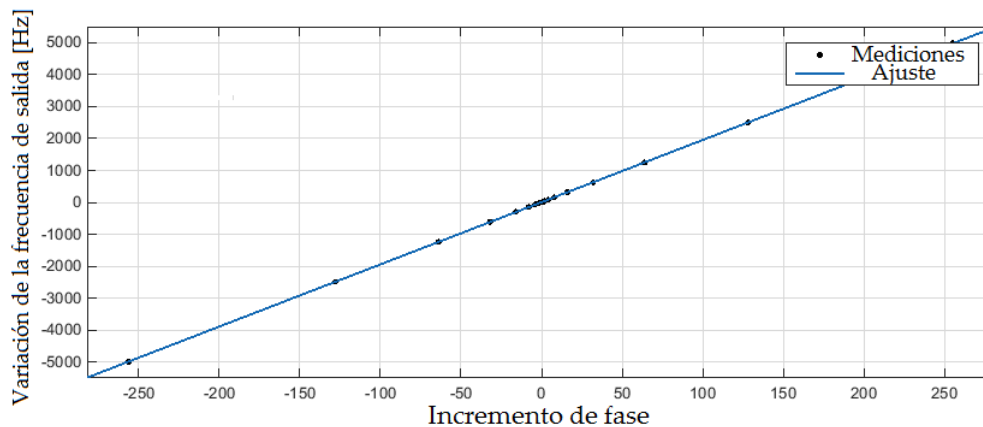


FIGURA 3.9: Valores medidos de frecuencia de salida en función del incremento de fase

## 3.4. Módulo de entrada

### 3.4.1. Interfaz

En la figura 3.10 se indica la interfaz del módulo de entrada, estando su descripción detallada en la tabla 3.4.

### 3.4.2. Operación

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.

En la figura 3.12 se puede observar un diagrama en bloques del módulo de entrada. El primer flip flop se encuentra conectado al reloj del *frontend* (denominado

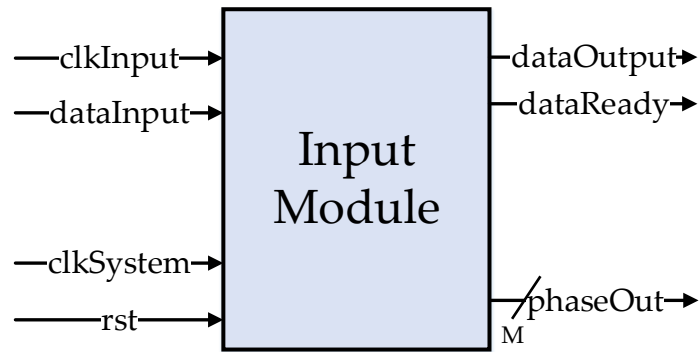


FIGURA 3.10: Interfaz del módulo de entrada

TABLA 3.4: 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 |

*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 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  $f_{clk} = 81,84MHz$  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 (correspondiendo a un período de código C/A) de forma periódica, ya que como se verá, es importante conocer la fase de la señal de entrada en los módulos que se encuentran a continuación.

### 3.4.3. 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.

En la figura 3.12 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*.

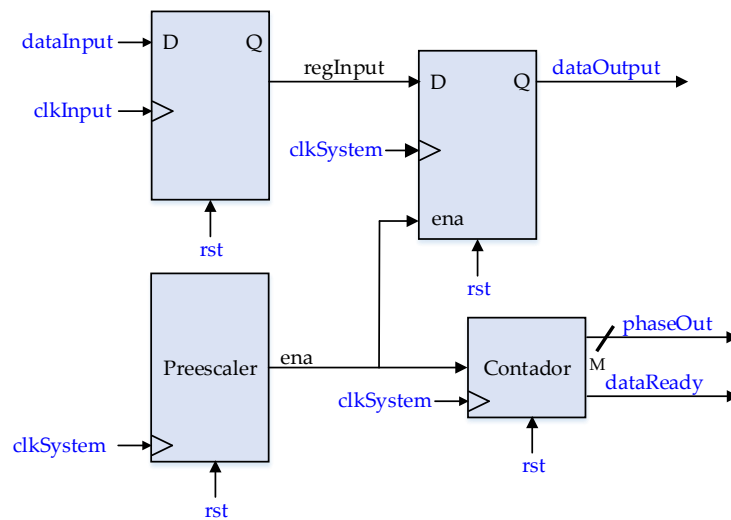


FIGURA 3.11: Diagrama en bloques del módulo de entrada

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*. *dataReady* se pone en alto un ciclo de reloj indicando que hay un nuevo dato válido y *phaseOut* incrementa su valor.

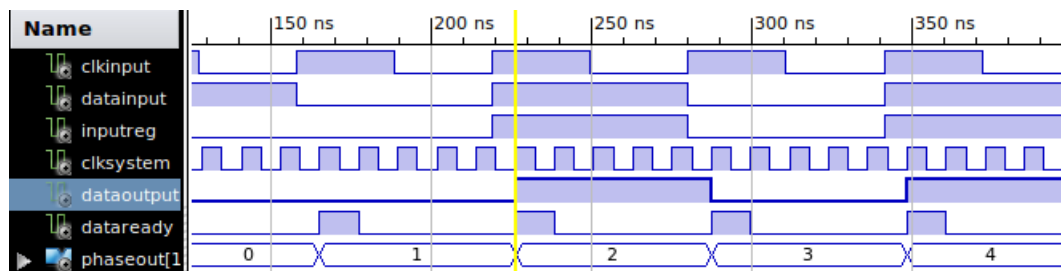


FIGURA 3.12: Diagrama de señales del módulo de entrada

### 3.5. Módulo de memoria

#### 3.5.1. Interfaz

En la figura 3.13 se indica la interfaz del módulo de memoria, estando su descripción detallada en la tabla 3.5.

#### 3.5.2. Operación

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 ( $f_{clk} = 81,84\text{MHz}$ ). Esto permite:

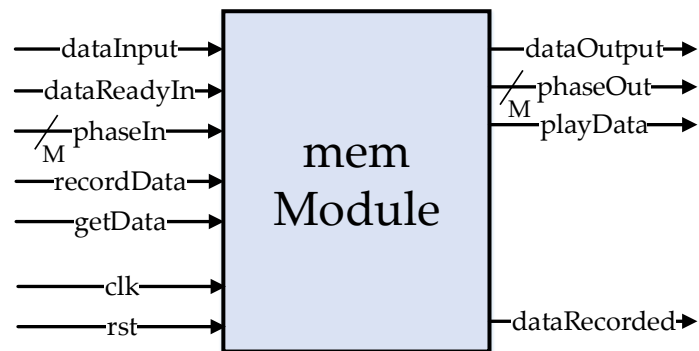


FIGURA 3.13: Interfaz del módulo de memoria

TABLA 3.5: 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 3.14 se muestra un diagrama en bloques del módulo de memoria, donde se puede observar que el módulo está implementado como una FSMMD según lo descrito en la sección 3.1.2.

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') y lógica adicional. Ambas memorias se implementaron utilizando una *blockRAM* de la FPGA. Las memorias 0 y 1 se turnan alternativamente a fin de registrar la señal de entrada

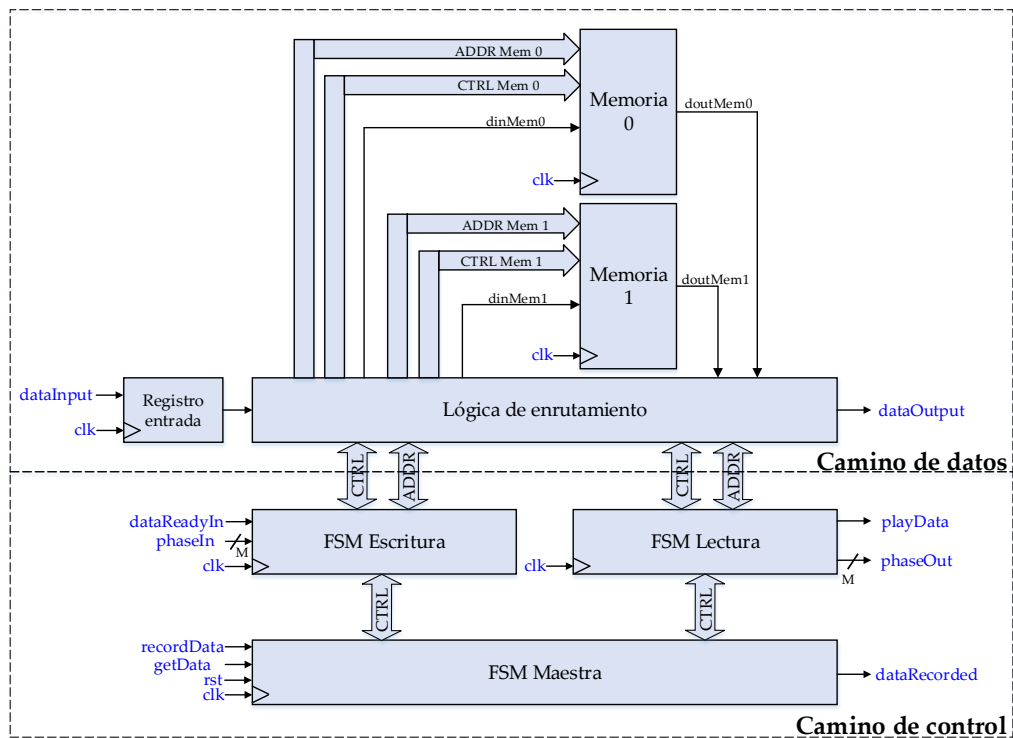


FIGURA 3.14: Diagrama en bloques del módulo de memoria

de forma tal que una vez en régimen siempre haya una de ellas disponible para la lectura. Esto permite 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, como se indicará en breve.

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). Así, se inicia la grabación de los datos de entrada de forma alineada (aguarda la condición *PhaseIn* = 0), dándole la orden a 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 en qué 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.

### 3.5.3. Pruebas

Se realizaron pruebas (*testbenches*) a fin de comprobar el funcionamiento del módulo de memoria en diferentes condiciones. Las mismas se describen a continuación.

#### Escritura continua

En esta prueba se le indicó al módulo que registre la señal de entrada (a los fines de la prueba una señal cuadrada) por medio de la entrada de control (*recordData* = 1). En la figura 3.15, se observa el comienzo de la prueba. Una vez detectada la orden *recordData* = 1, la ‘FSM Escritura’ aguarda la fase cero (*phaseIn* = 0) para alinearse, permaneciendo en el estado *detect\_frame*.

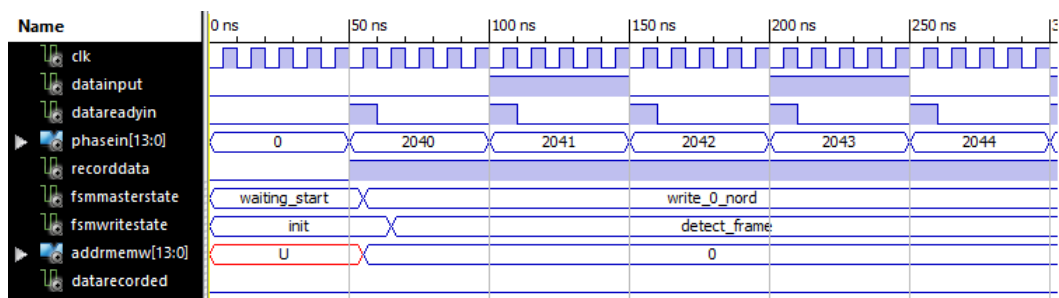


FIGURA 3.15: Comienzo de la prueba de escritura continua

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 escritura *weamemw* y actualizando las direcciones de memoria *addrmemw* (ver figura 3.16).

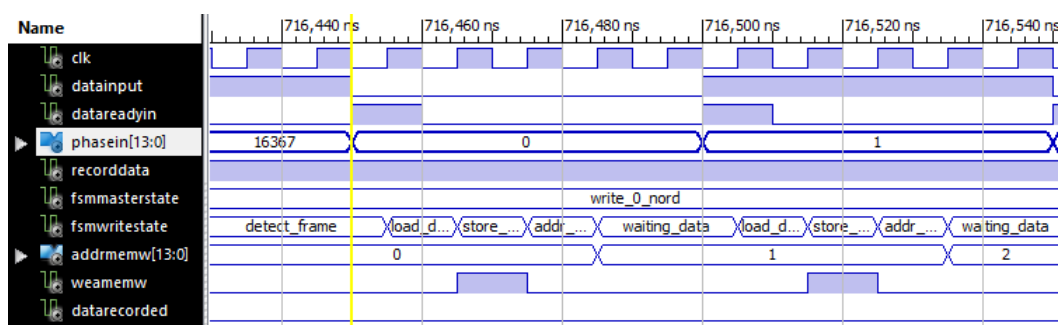


FIGURA 3.16: Comienzo del ciclo de grabación

Al finalizar el ciclo de grabación (se grabaron 16.368 muestras), 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 3.17).

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 3.18.

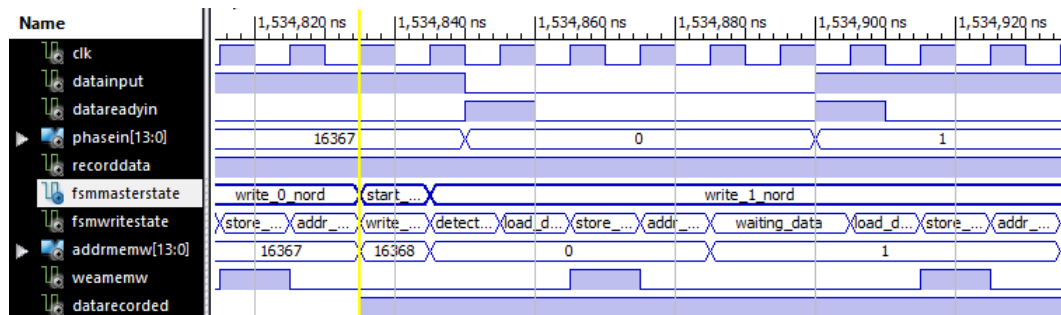


FIGURA 3.17: Fin del ciclo de grabación de la memoria 0, comienzo de grabación de la memoria 1

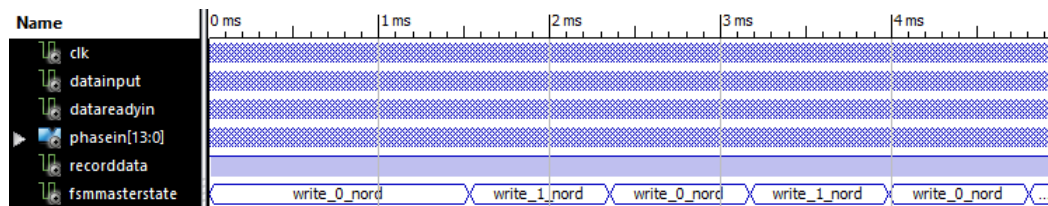


FIGURA 3.18: Escritura alternada de las memorias 0 y 1

## Única lectura

En esta prueba se indicó al módulo que registre la señal de entrada (a los fines de la prueba 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 3.19.

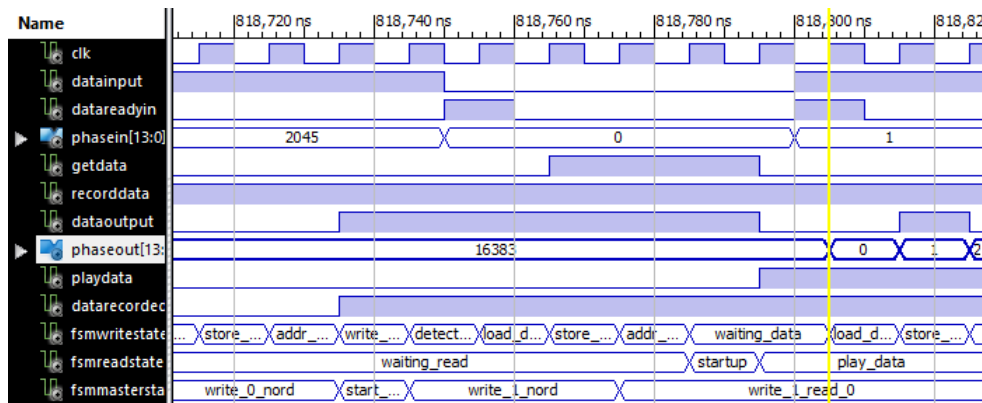


FIGURA 3.19: Fin de escritura de la memoria 1 y comienzo de lectura de memoria 0

En la figura 3.20 se observa un detalle de la reproducción de la señal de salida. Se observa la señal cuadrada reproducida a la frecuencia del clock del sistema indicando a través de `phaseOut` el número de muestra reproducida.

En ausencia de comandos de lectura adicionales, el módulo de memoria permanece escribiendo los bancos de memoria alternadamente (ver figura 3.21).



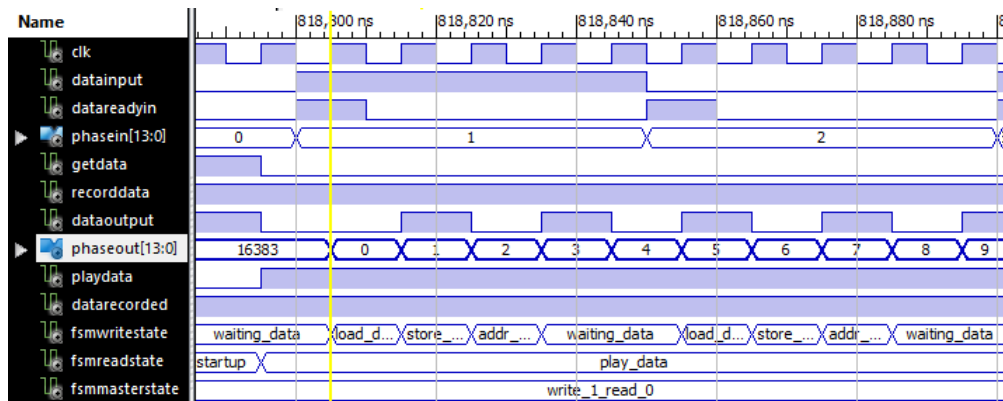


FIGURA 3.20: Lectura de memoria 0

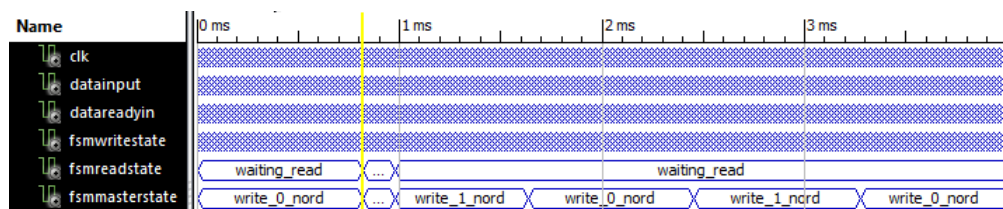


FIGURA 3.21: Escritura alternada de las memorias 0 y 1 en ausencia de nuevo comando de lectura

### Lectura/Escritura continua

En esta prueba se indicó al módulo que registre la señal de entrada (a los fines de la prueba 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.

En la figura 3.22 se puede apreciar como ante sucesivos pedidos de lectura se procede a reproducir la memoria 0 mientras la memoria 1 se está grabando (el estado de la 'FSM Maestra' permanece en *write\_1\_read\_0*).



FIGURA 3.22: Lectura continua de la memoria 0 mientras la memoria 1 se está grabando

En la figura 3.23 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.

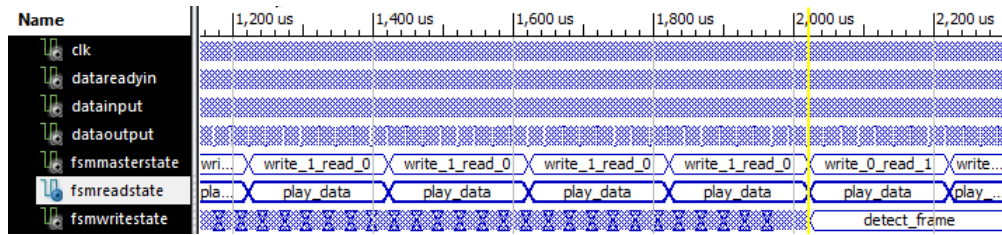


FIGURA 3.23: Detalle del cambio de escritura/lectura de la memoria 1 a la memoria 0

## 3.6. Módulo de generación de señal GPS local

### 3.6.1. Interfaz

En la figura 3.24 se indica la interfaz del módulo de generación de señal GPS local, estando su descripción detallada en la tabla 3.6.

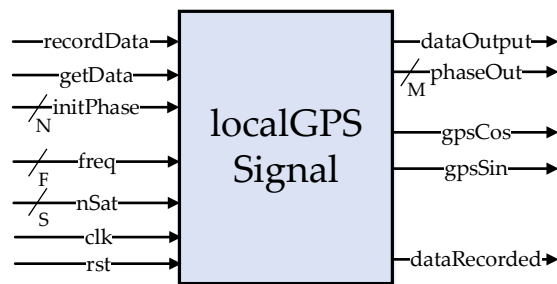


FIGURA 3.24: Interfaz del módulo de generación de señal GPS local

TABLA 3.6: 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                 |

### 3.6.2. Operación

El módulo de entrada tiene por objeto generar, grabar internamente 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 poniendo en alto la entrada *getData*. 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 3.25 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.

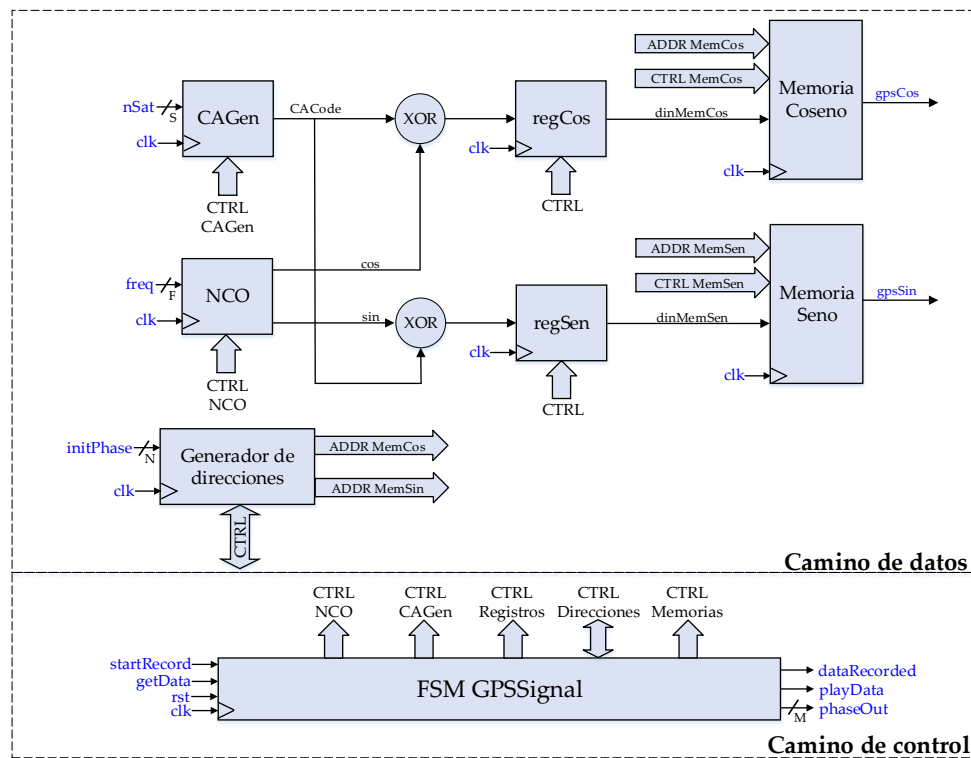


FIGURA 3.25: Diagrama en bloques del módulo de generación de señales local

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, *buffers* 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 signado, permitiendo variar la frecuencia central del NCO respecto de la frecuencia de la FI en pasos de 500Hz a ambos lados. El rango de ajuste puede variarse en el intervalo  $[-8000, 8000)Hz$ .

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*. La señal *playData* se pone en alto dos ciclos de reloj antes de la reproducción para comunicar la situación a los "módulos cliente".

### 3.6.3. 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*. 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. La validación se describe en el Capítulo 4.

En la figura 3.26 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.

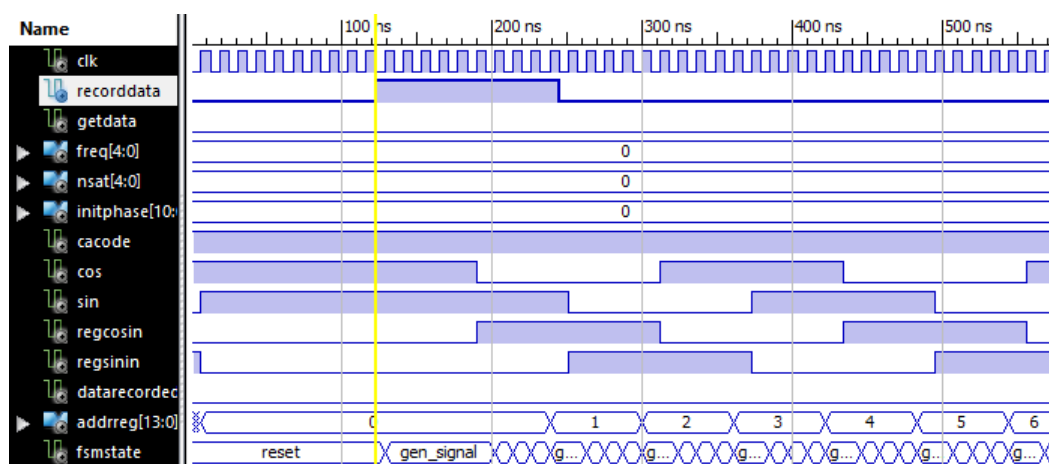


FIGURA 3.26: Inicio de la grabación de señales réplica

En la figura 3.27 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*).

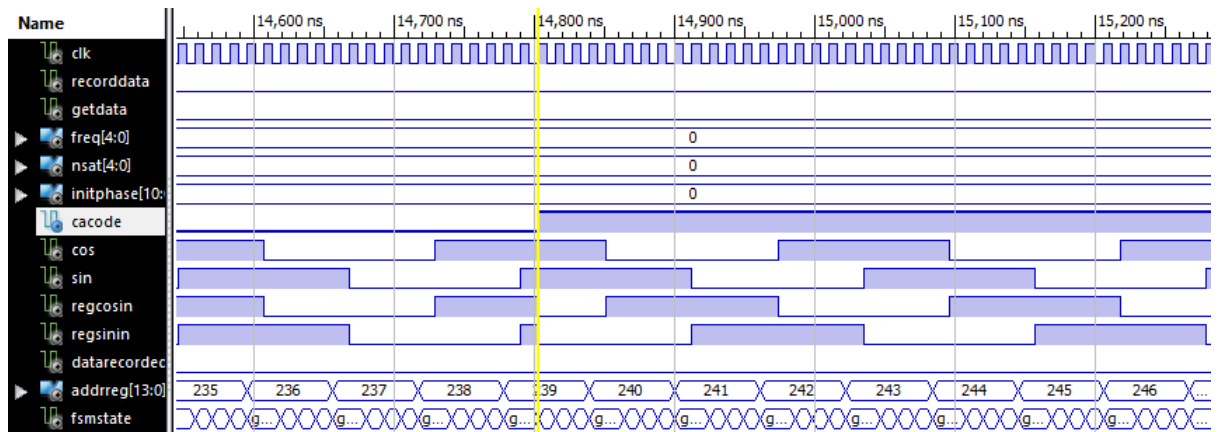


FIGURA 3.27: Detalle de la inversión de fase de las señales réplica

En la figura 3.28 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*.

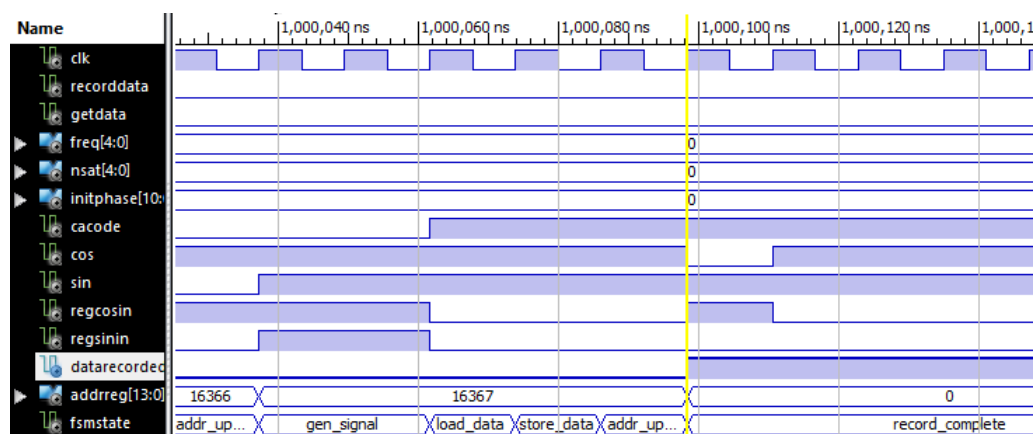
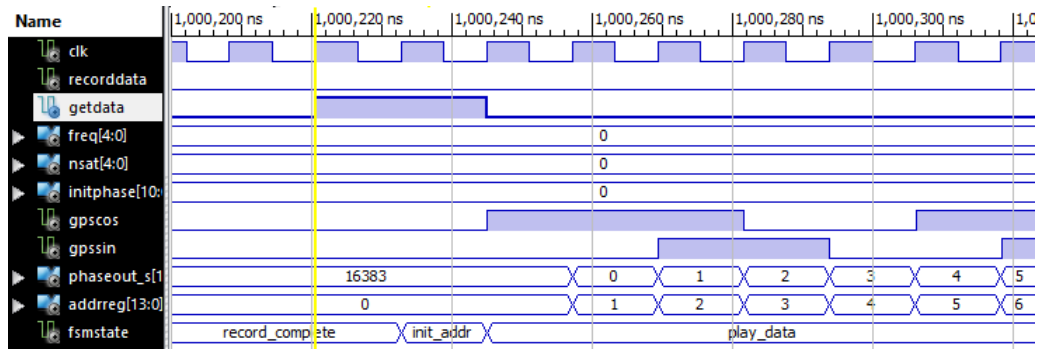
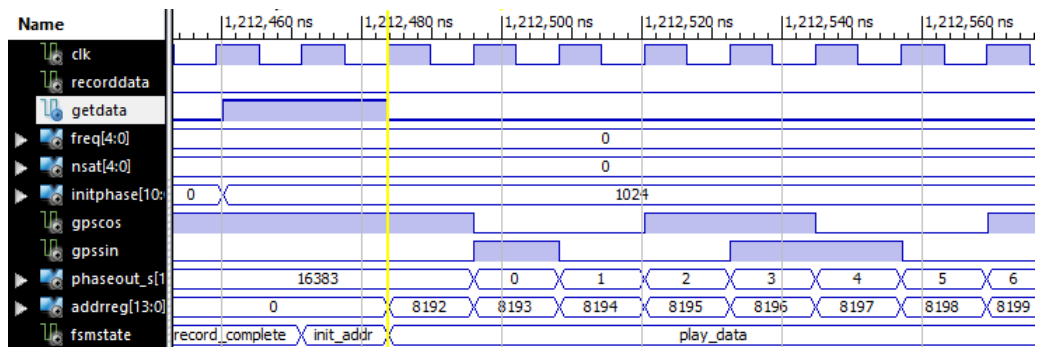


FIGURA 3.28: Fin de la grabación de las señales réplica

Para reproducir las señales réplica se le envía al módulo el comando *getData* = 1. 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 3.29 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 3.30 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) se procede a leer a partir de la dirección inicial nuevamente (*addrreg* = 0)

FIGURA 3.29: Lectura de las señales réplica (*initPhase=0*)FIGURA 3.30: Lectura de las señales réplica (*initPhase=1024*)

como se indica en la figura 3.31, ya que la memoria se utiliza como un *buffer* circular.

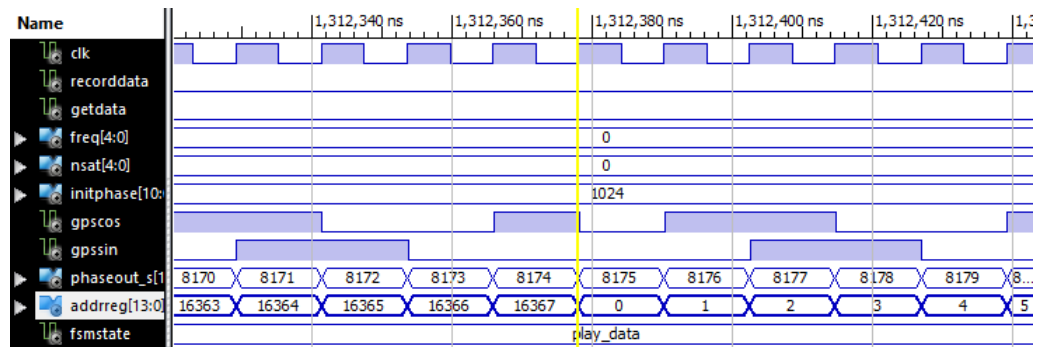


FIGURA 3.31: Detalle del cambio de dirección de lectura

## 3.7. Módulo de búsqueda

### 3.7.1. Interfaz

En la figura 3.32 se indica la interfaz del módulo de búsqueda, estando su descripción detallada en la tabla 3.7.

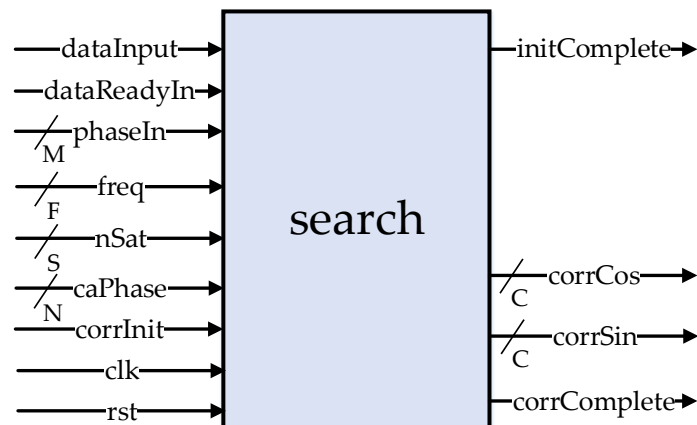


FIGURA 3.32: Interfaz del módulo de búsqueda

TABLA 3.7: Descripción de la interfaz 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                            |
| 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                       |

### 3.7.2. 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 3.25 se presenta un diagrama en bloques del módulo. Éste 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*).

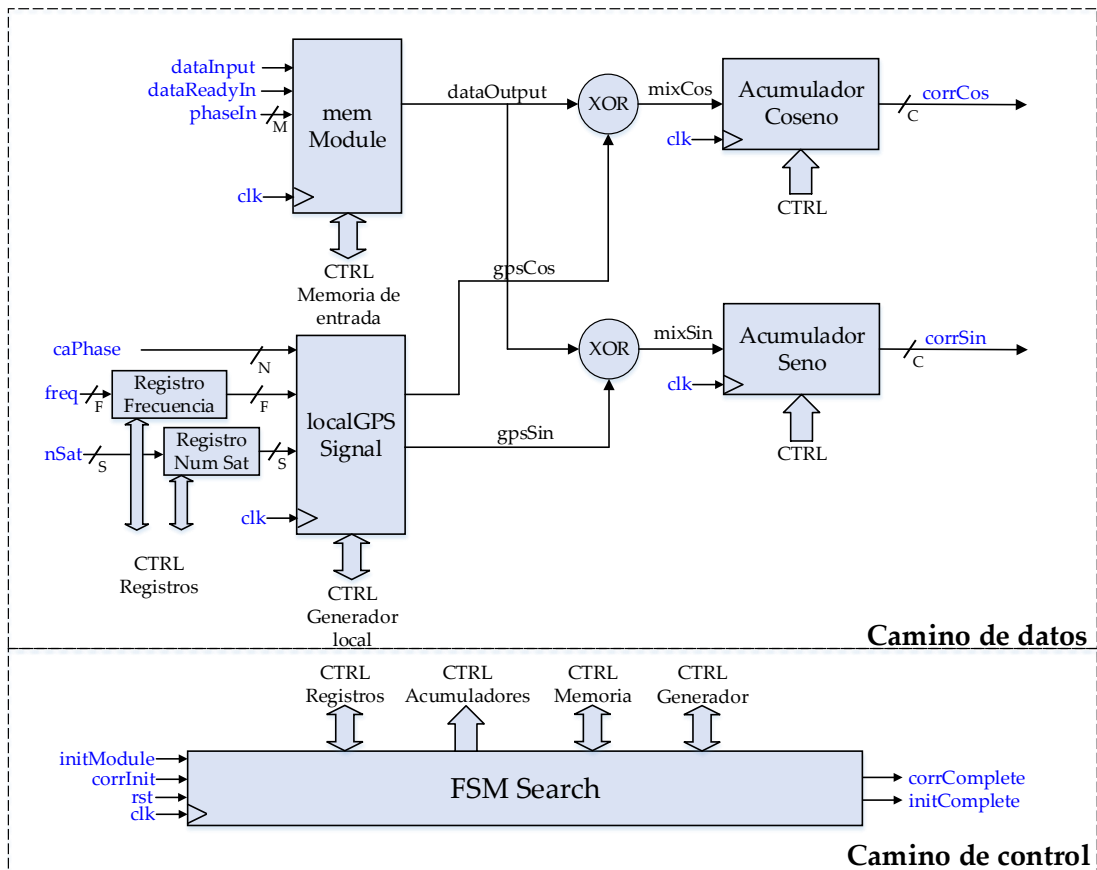


FIGURA 3.33: Diagrama en bloques del módulo de búsqueda

El camino de datos se constituye por:

- **Módulo de memoria:** Su función es la de mantener un flujo de muestras continuo y actualizado de la señal de entrada (ver sección 3.5).
- **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 3.6.
- **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 3.6).



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 3.5). 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.

### 3.7.3. 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 3.34, se indica la transición de la entrada *initModule*. 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.

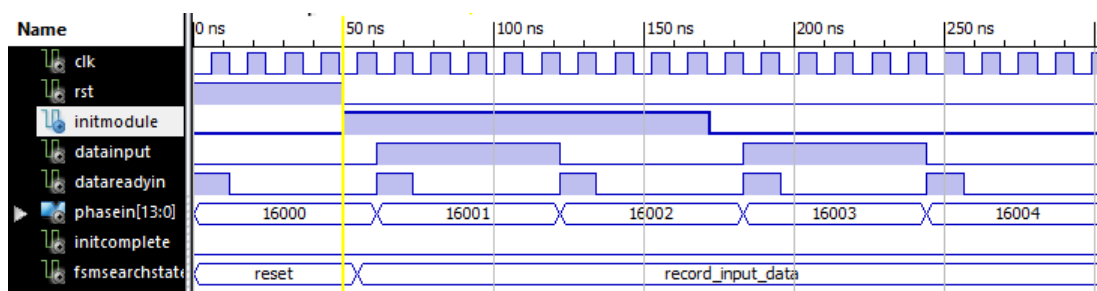


FIGURA 3.34: Comienzo de la inicialización del módulo de búsqueda

En la figura 3.35 se puede observar el fin de la inicialización. Nótese que ya se ha completado la grabación de un ciclo de muestras (la muestra final es la número 16367).

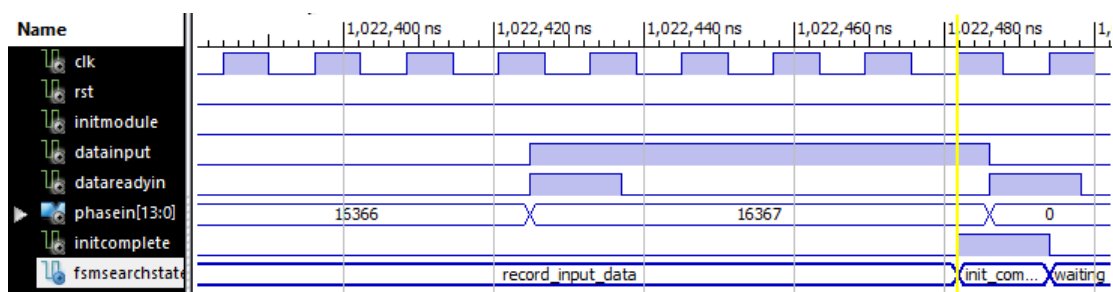


FIGURA 3.35: Fin de la inicialización del módulo de búsqueda

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, por ende, la 'FSM Search' pasa al estado *record\_gps\_freq\_sat* (ver figura 3.36).

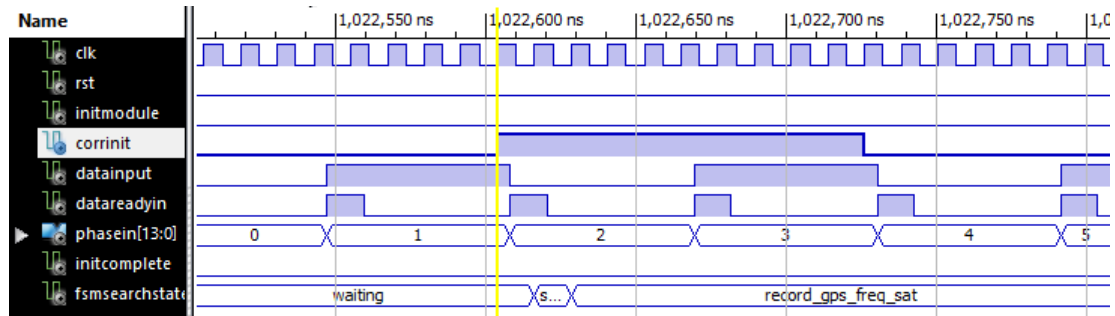


FIGURA 3.36: Inicio de generación de señal GPS local

Una vez generada la señal de réplica se inicia la correlación (ver figura 3.37). El estado de la 'FSM Search' pasa a *corr* y los acumuladores (*corrCos* y *corrSin*) comienzan a operar. Nótese que las señales correlacionadas están en fase, coincidiendo las señales internas *phaseout\_s* (correspondiente al número de muestra del módulo de memoria) y *phaseout* (correspondiente al número de muestra del módulo de generación de réplicas).

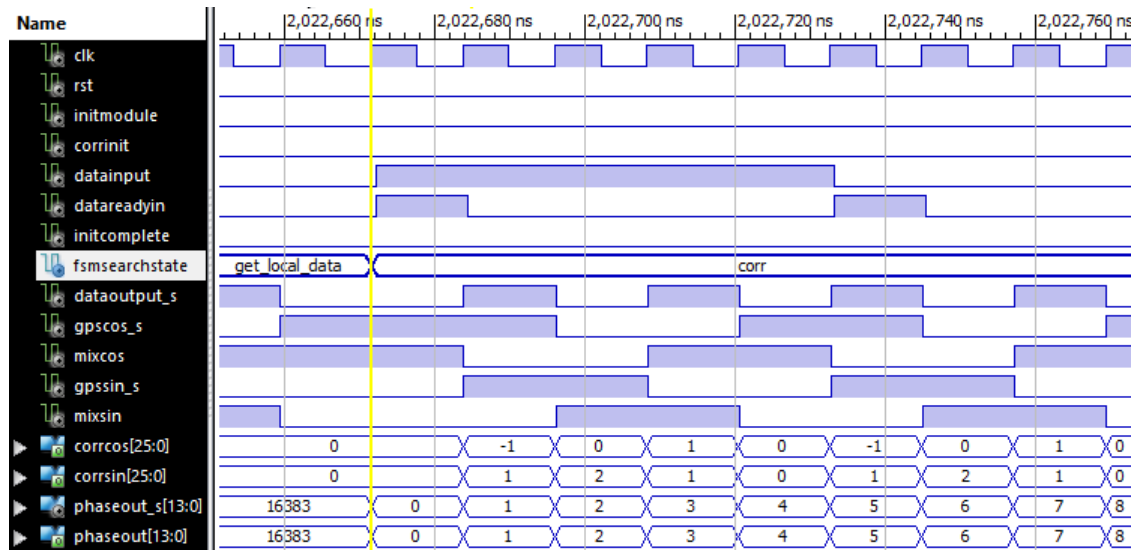


FIGURA 3.37: Inicio de la correlación de señal GPS local

Finalmente, cuando se ha completado la correlación, la salida *corrComplete* pasa a estado alto. Como se ha explicado, en caso de que se inicie una nueva correlación variando sólo la fase de la señal inicial (*caPhase*), ésta comenzará inmediatamente (ver figura 3.38). En caso contrario, se necesitará pasar por el estado *record\_gps\_freq\_sat* para regenerar las señales réplicas locales.

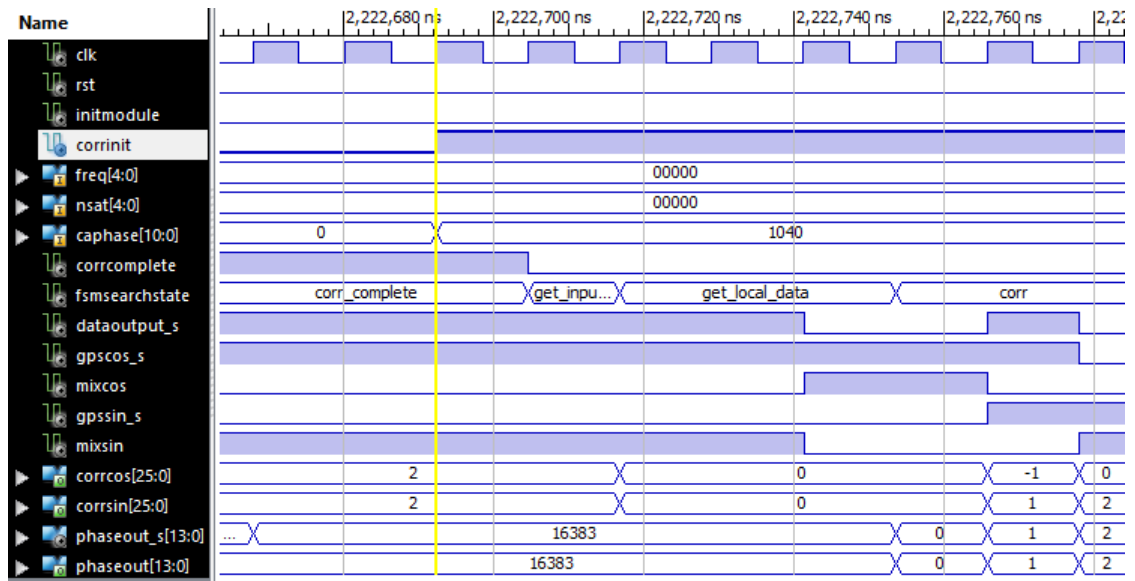


FIGURA 3.38: Fin de la correlación de señal GPS local

### 3.8. Módulo de seguimiento

#### 3.8.1. Interfaz

En la figura 3.39 se indica la interfaz del módulo de seguimiento, estando su descripción detallada en la tabla 3.7.

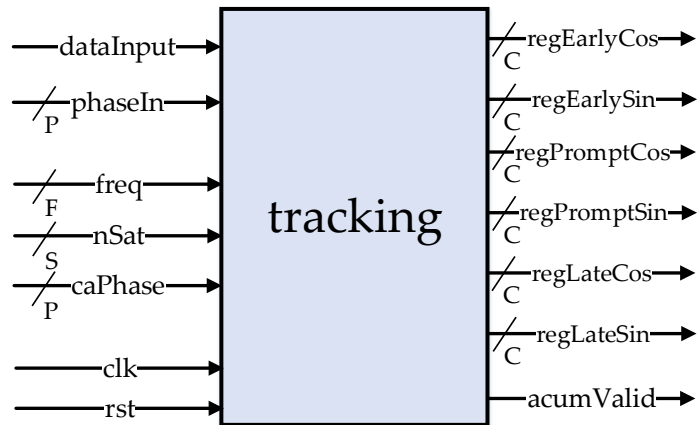


FIGURA 3.39: Interfaz del módulo de seguimiento

#### 3.8.2. 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 3.40 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.

TABLA 3.8: 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                      |

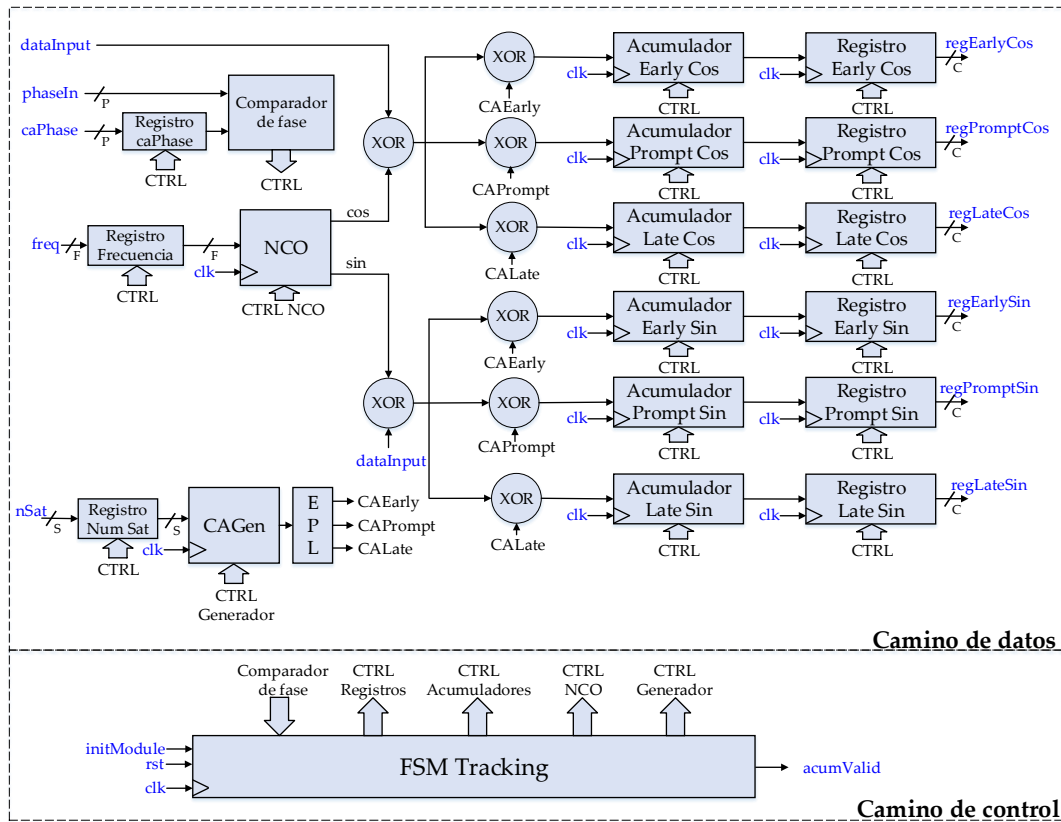


FIGURA 3.40: Interfaz del módulo de seguimiento

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  $0,02Hz$  aproximadamente (ver sección 3.3).
- **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 entre si).
- **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 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.

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 del dato es transferido 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 se supone que la señal de entrada está en fase con el código C/A Prompt, el valor de correlación para las señales C/A Early (adelantado medio chip, equivalente a 8 muestras) y C/A Late (atrasado medio chip, equivalente a 8 muestras) será igual (en ausencia de ruido) y la mitad que para el valor de C/A Prompt. 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 3.41.

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 3.42).

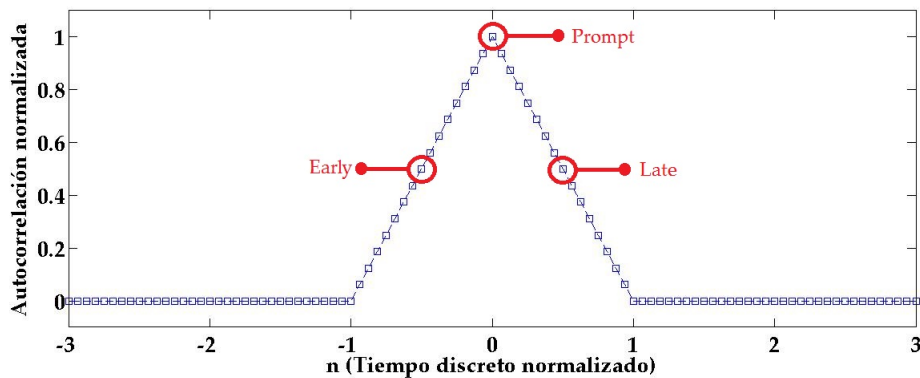


FIGURA 3.41: Valores de correlación normalizados para los códigos Early, Prompt y Late (señal en fase)

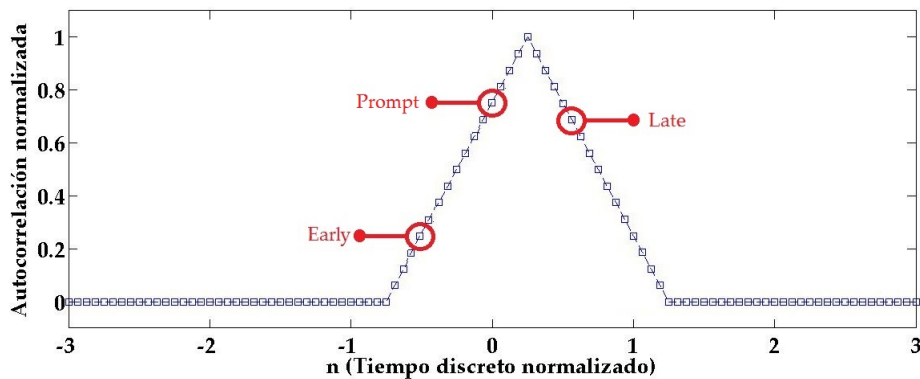


FIGURA 3.42: Valores de correlación normalizados para los códigos Early, Prompt y Late (señal atrasada)

En la figura 3.43 se pone 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 [3].

### 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 la hipótesis de 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 3.44 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 sin la realimentación y es una estructura muy utilizada para la detección de las señales GPS [1], donde los acumuladores suplen los filtros pasabajos utilizados en la estructura tradicional del lazo.

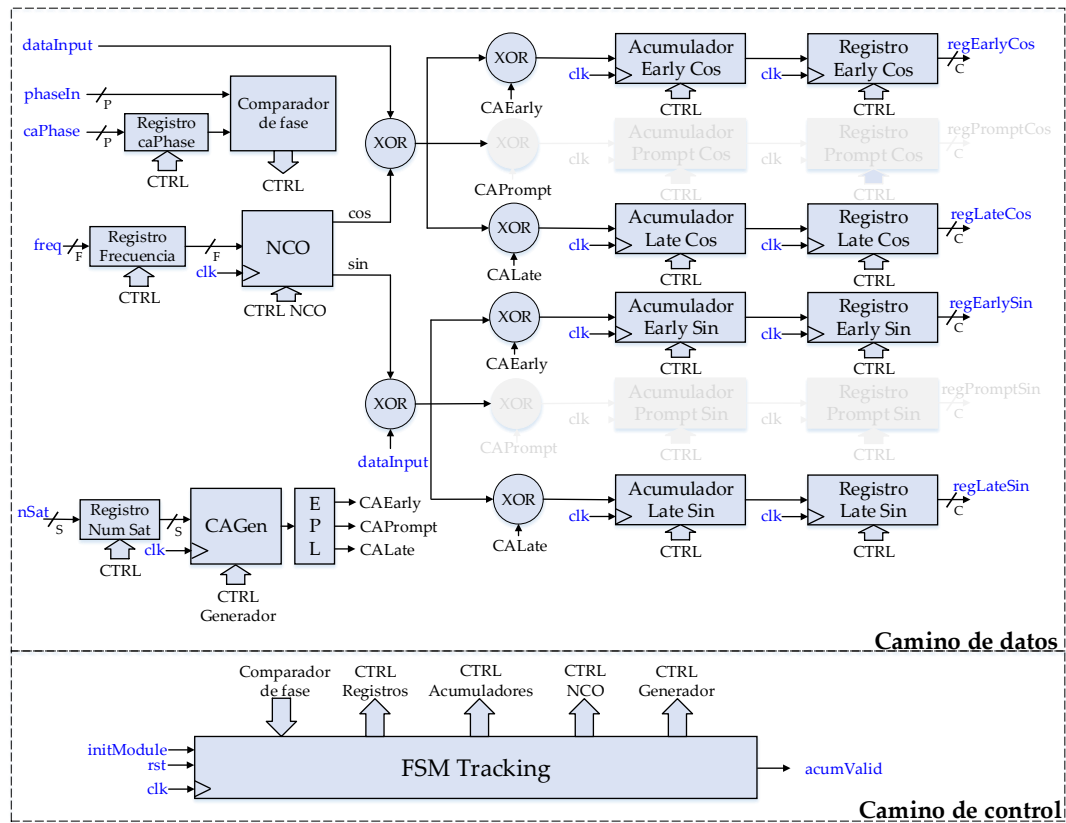


FIGURA 3.43: Diagrama en bloques (énfasis en los bloques utilizados para detectar la sincronía del código C/A)

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

### 3.8.3. 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  $nSat = 1$ .

En la figura 3.45 se puede observar la inicialización del módulo. La entrada *setStatus* se pone en estado alto, 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.

En la figura 3.46 se puede notar como al llegar a la muestra 152, 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 3.46, 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).

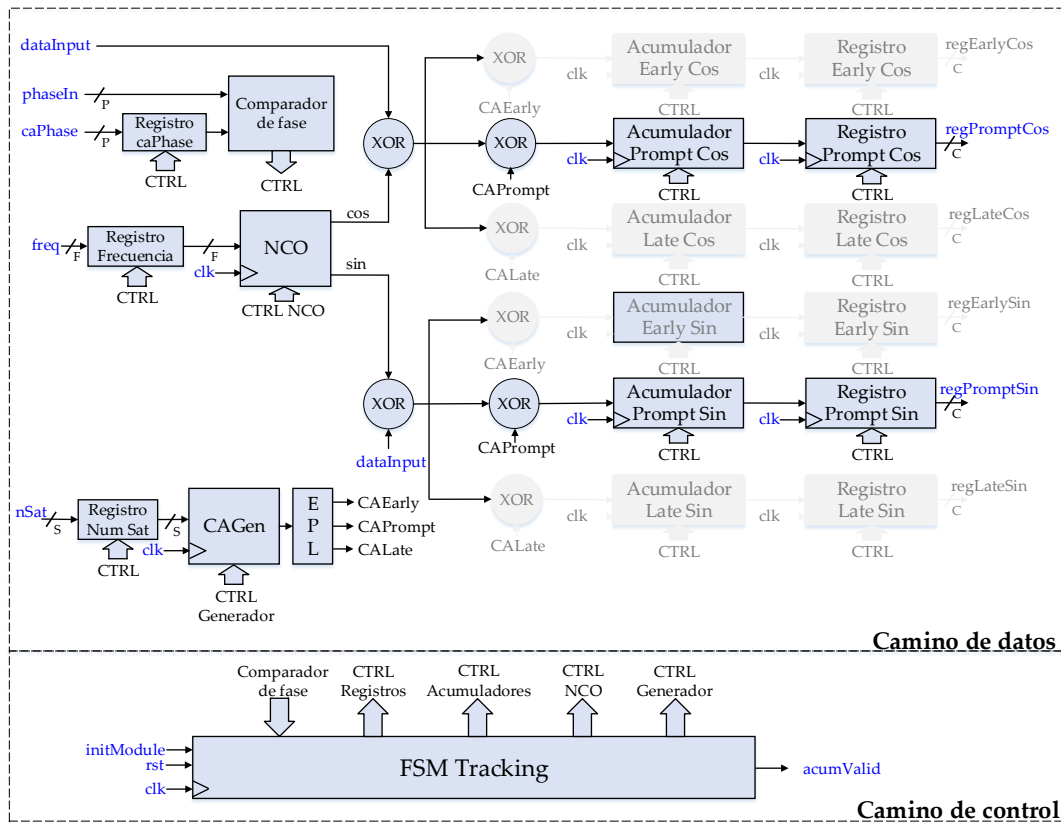


FIGURA 3.44: Diagrama en bloques (énfasis en los bloques utilizados para detectar la sincronía de la señal de FI)

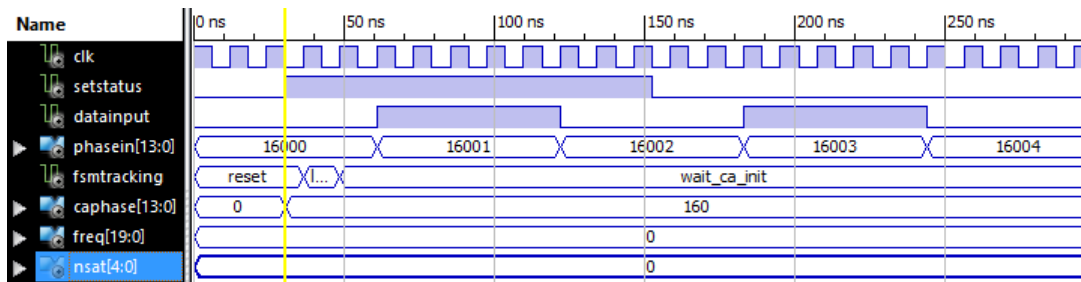


FIGURA 3.45: Inicialización módulo de seguimiento

La 'FSM Tracking' pasa al estado *wait\_rollover* a la espera del inicio del *frame* (muestra 0).

En la figura 3.47 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*.

En la figura 3.48 se puede observar al finalizar el *frame* (muestra 16367) la actualización de los registros de salida. 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.



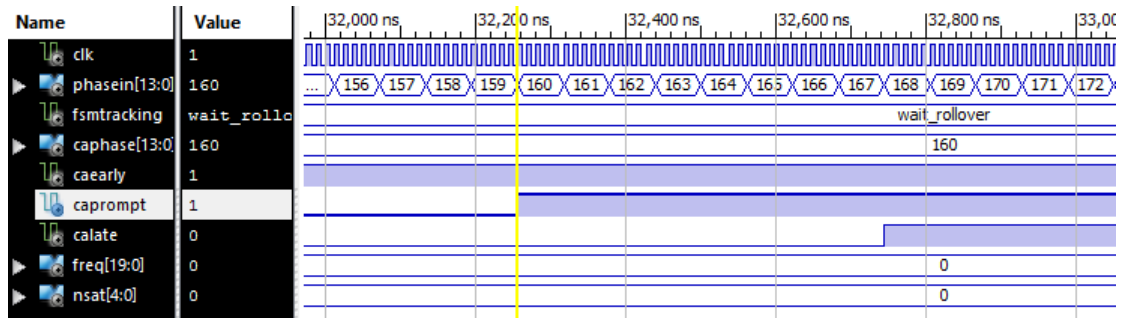


FIGURA 3.46: Sincronización del código C/A

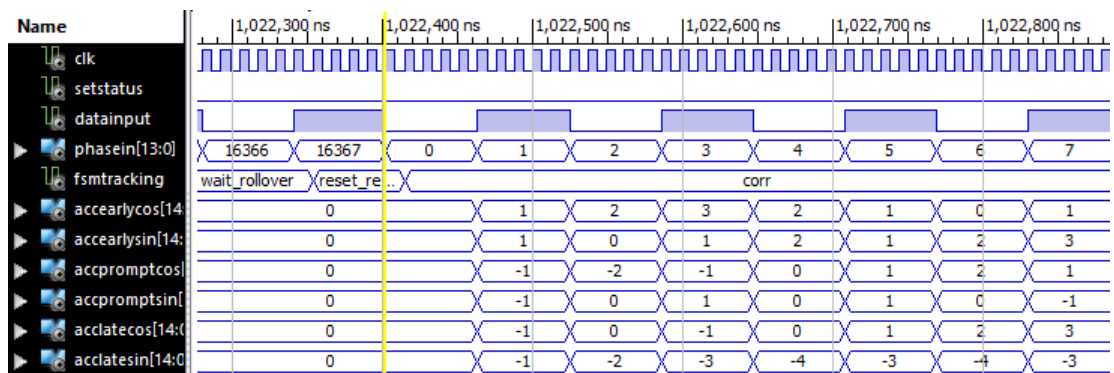


FIGURA 3.47: Inicio de la acumulación



FIGURA 3.48: 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*.



## Capítulo 4

# Ensayos y Resultados

En este capítulo se describen diferentes validaciones que se realizaron para comprobar el correcto funcionamiento de los módulos planteados utilizando como referencia modelos de simulación desarrollados durante la etapa de estudio del proyecto.

### 4.1. Validación de generación de señales réplica

En esta sección se describe la validación de las señales réplica generadas por el módulo de 'Generación de señales GPS local'.

En la etapa de estudio, se desarrolló en MATLAB un conjunto de *scripts* y funciones con el objeto de comprobar las técnicas de generación y detección de señales GPS cuyo análisis matemático se describió en el capítulo 2.

La función *GenGPSSignal* genera de forma sintética una señal GPS (producto entre una señal portadora en la FI y el código C/A de un satélite dado):

```
[SenalGPS]=GenGPSSignal(fi,deltaf,initPhaseCarrier,bits,nSat,
initPhaseCA,length,samplesPerChip)
```

donde:

- *fi*: Frecuencia intermedia [Hz]
- *deltaf*: Variación de la frecuencia intermedia por efecto Doppler [Hz]
- *initPhaseCarrier*: Fase inicial de la portadora [radianes]
- *bits*: Cantidad de bits utilizados para la generación de la señal portadora
- *nSat*: Número de satélite cuyo código C/A se aplicará
- *initPhaseCA*: Fase inicial del código C/A [muestras]
- *length*: Longitud de la señal a obtener [muestras]
- *samplesPerChip*: Muestras por chip
- *SenalGPS*: (Vector de salida) Senal GPS generada

La función *Search* 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).

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

- 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

Las funciones anteriores se utilizaron para generar las señales sintéticas y las matrices de correlación graficadas en el capítulo 2.

Con el objeto de comprobar la correcta generación de las señales réplica generadas en la FPGA (módulo 'Generación de señales GPS local', ver sección 3.6), se registró la señal generada en el *testbench* del módulo por medio del procedimiento *write* del paquete *textio* el cual permite crear un archivo de texto en base a las señales generadas.

La señal obtenida del ensayo, a la que se denominará *SenalGPSReplica* se comparó con una señal generada a través de la función *GenGPSSignal* a la que se denominará *SenalGPSPatron*. Ambas señales se generaron utilizando los mismos parámetros de entrada y fueron sometidas al mismo procesamiento a través de la función *Search* a fin de obtener su matriz de correlación. El procedimiento de ensayo se ilustra de forma esquemática en la figura 4.1.

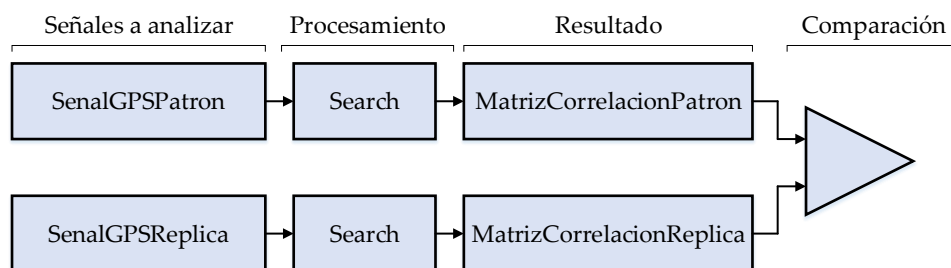


FIGURA 4.1: Diagrama esquemático del procedimiento de validación de las señales réplica

A modo de ejemplo, se generaron señales con los mismos parámetros de entrada (número de satélite: 1, desplazamiento de código C/A  $n_{0l}$ : 1021,5 chips, desplazamiento de frecuencia central  $\Delta f_{dl}$ : 0 Hz). En la figura 4.2 se muestra el gráfico de la columna de la matriz de correlación ( $\Delta f_{dl} = 0$ ) en función de la fase del código C/A para la señal patrón y la señal réplica.

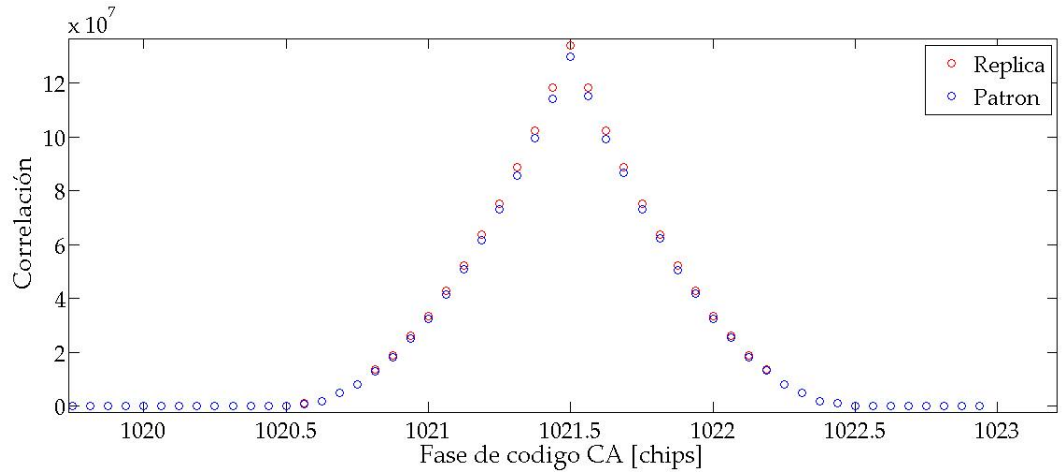


FIGURA 4.2: Gráfico de correlación en función de la fase del código C/A para  $\Delta f_{dop} = 0$

En la figura 4.3 se muestra el gráfico de la fila de la matriz de correlación ( $n_{0l} = 1021,5$ ) en función del desplazamiento de frecuencia para la señal patrón y la señal réplica.

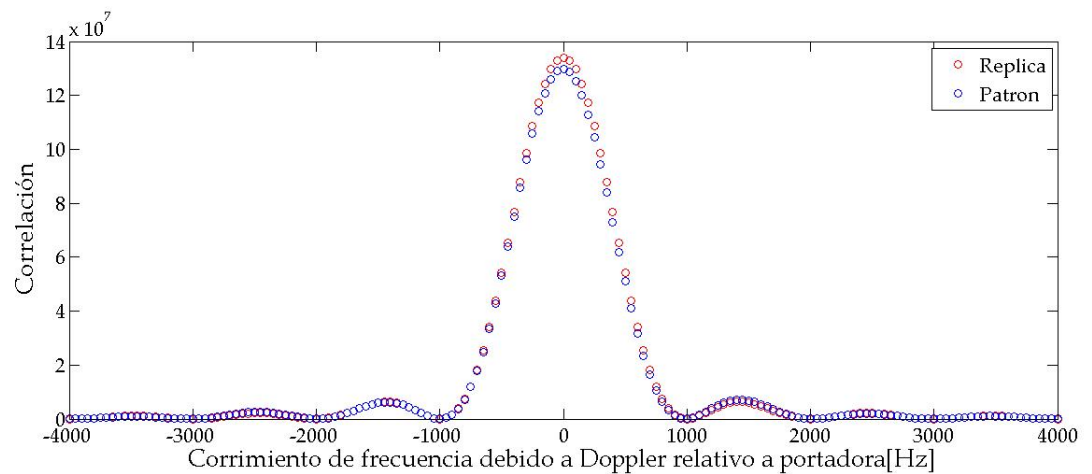


FIGURA 4.3: Gráfico de correlación en función de la frecuencia para  $n_{0l} = 1021,5$

El procedimiento se repitió para diferentes parámetros de entrada comprobándose la correcta generación de las señales réplica.

## 4.2. Validación de módulo de búsqueda

El módulo de búsqueda realiza las operaciones matemáticas explicadas en el capítulo 2. Estas operaciones son equivalentes a las realizadas por la función *Search*. A fin de comprobar el correcto funcionamiento del módulo de búsqueda (ver sección 3.7) se planteó un esquema de prueba como el que se indica en la figura 4.4.

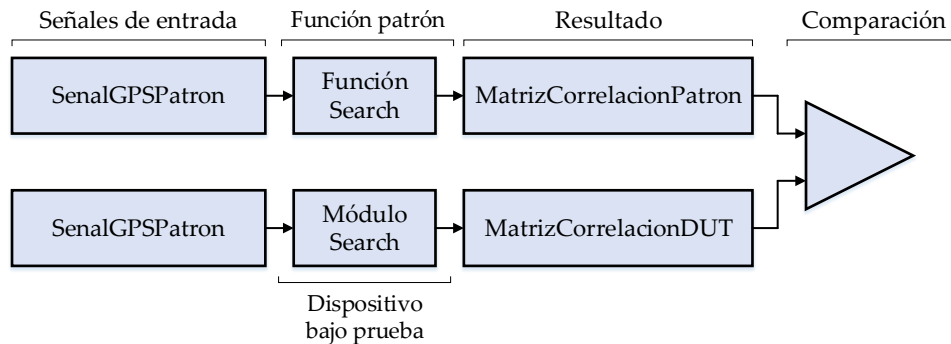


FIGURA 4.4: Diagrama esquemático del procedimiento de validación del módulo de búsqueda

En primer lugar, una señal patrón se procesa a través de la función *Search* dando por resultado una matriz de correlación que se denominará *MatrizCorrelacionPatron*, en segundo lugar, la misma señal se cargará al *testbench* del módulo de búsqueda a través de un archivo de texto que se reproducirá como señal de entrada. Las salidas de correlación producidas para el módulo de búsqueda para diferentes frecuencias de desplazamiento y fases de código C/A se grabarán en un archivo de texto. En base a estas salidas se construirá una matriz *MatrizCorrelacionDUT*. Finalmente, se compararán ambas matrices para determinar si el resultado logrado es el esperado.

A modo de ejemplo se realizó la operación de búsqueda sobre una señal con las siguientes características: desplazamiento de la frecuencia central  $\Delta f_c = -1000\text{Hz}$  y desplazamiento del código C/A  $n_0 = 5,5\text{ chips}$ . La señal se sometió al análisis de la función *Search* obteniéndose la matriz de correlación *MatrizCorrelacionPatron*. Por otro lado, se alimentó al módulo de búsqueda con la señal de entrada a través del *testbench* y se obtuvieron las salidas, a partir de las cuales se construyó la matriz *MatrizCorrelacionDUT*.

En ambos casos se pudo detectar el máximo de correlación, indicando la frecuencia y la fase de C/A. En la figura 4.5 se puede observar el gráfico en función de la frecuencia para la fila  $n_{0l} = 5,5$  (donde se encuentra el máximo de correlación) para los valores de correlación producidos por los diferentes métodos.

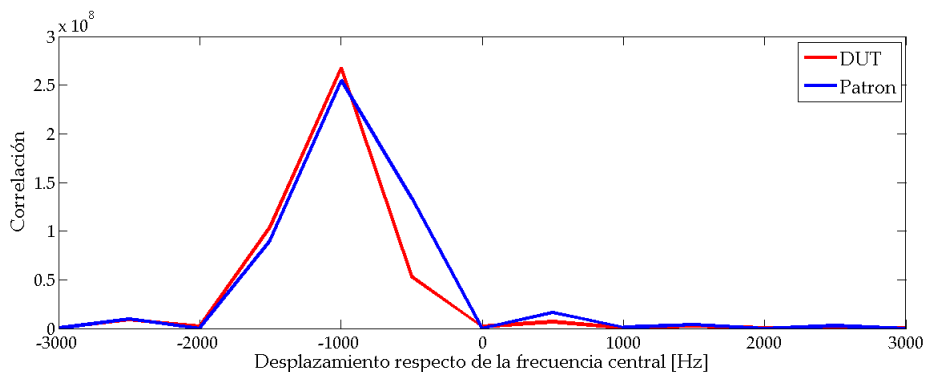


FIGURA 4.5: Gráfico de correlación en función de la frecuencia para  $n_{0l} = 5,5$

En la figura 4.6 se puede observar el gráfico en función de la fase del código C/A para la fila  $\Delta f_{dl} = -1000$  (dónde se encuentra el máximo de correlación) para los valores de correlación producidos por los diferentes métodos.

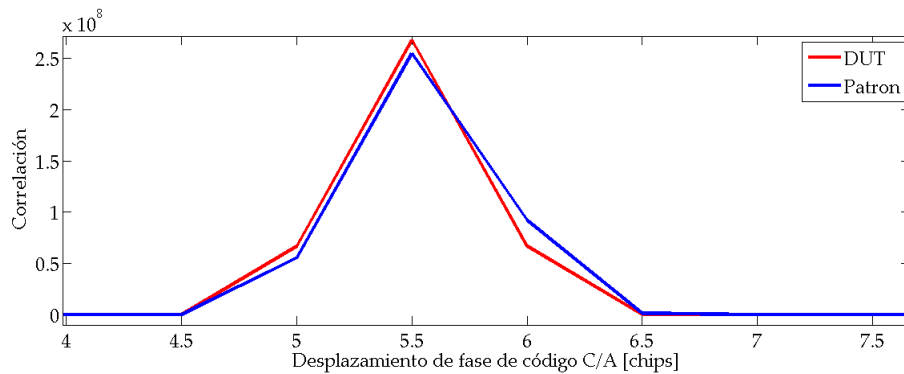


FIGURA 4.6: Gráfico de correlación en función de la fase de código C/A para  $\Delta f_{dl} = -1000$

### 4.3. Validación de módulo de seguimiento

El módulo de seguimiento produce a su salida valores de comparación de la señal de entrada respecto de un oscilador local en cuadratura (seno y coseno) y réplicas del código C/A desplazadas entre sí (Early, Prompt y Late). Estos valores pueden utilizarse para determinar si la señal de entrada está en fase respecto de la réplica del código C/A local y la fase/frecuencia del oscilador local es idéntica a la de entrada.

A fin de comprobar el correcto funcionamiento del módulo de seguimiento se alimentó a su entrada una señal con las siguientes características: desplazamiento de la frecuencia central  $\Delta f_c = 0Hz$  y desplazamiento del código C/A  $n_0 = 1018 \text{ chips}$  (equivalente a un desplazamiento de 16288 muestras).

Se realizó un *testbench* en el cual se cargó la señal de entrada por medio de una señal generada sintéticamente guardada en un archivo de texto. Se solicitó al módulo que realice la comparación para diferentes valores de fase de código C/A en torno al valor conocido de inicio (1018 chips = 16288 muestras).

En la figura 4.7 se puede observar los diferentes valores de correlación para los acumuladores Early, Prompt y Late. La correlación se calcula obteniendo la raíz cuadrada de la suma de los acumuladores seno y coseno (para cada código) al cuadrado.

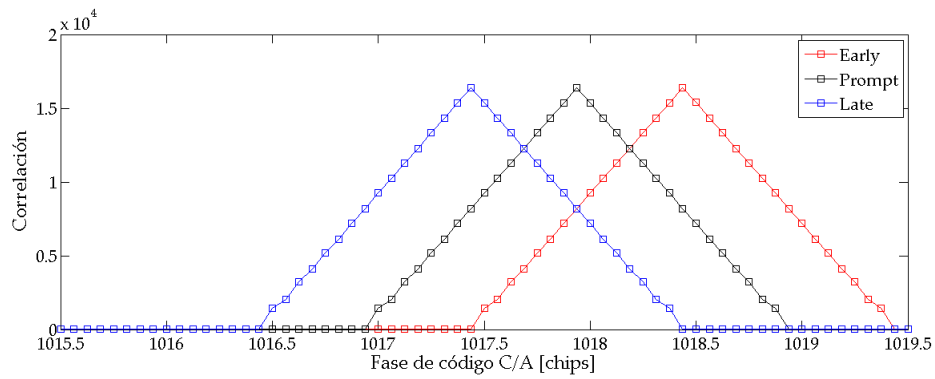


FIGURA 4.7: Gráfico de correlación en función de la fase de código C/A [chips]

#### 4.4. Recursos utilizados

En esta sección se indican los recursos utilizados de la FPGA de referencia (Xilinx Spartan 3E 500) para el módulo de búsqueda y seguimiento.

TABLA 4.1: Recursos utilizados para el módulo de búsqueda

| Recurso          | Utilizados | Disponible | Porcentaje |
|------------------|------------|------------|------------|
| Slice Flip Flops | 214        | 9.312      | 2 %        |
| 4 Input LUTs     | 326        | 9.312      | 3 %        |
| Occupied slices  | 278        | 4.656      | 5 %        |
| RAMB16s          | 4          | 20         | 20 %       |

TABLA 4.2: Recursos utilizados para el módulo de seguimiento

| Recurso          | Utilizados | Disponible | Porcentaje |
|------------------|------------|------------|------------|
| Slice Flip Flops | 342        | 9.312      | 3 %        |
| 4 Input LUTs     | 261        | 9.312      | 2 %        |
| Occupied slices  | 282        | 4.656      | 6 %        |
| RAMB16s          | 0          | 20         | 0 %        |

A ambos módulos se les agregó el *timing constraint* de reloj a 81,84Mhz resultando que ambos cumplieron con el *timing* exigido.



## Capítulo 5

# Conclusiones

### 5.1. Conclusiones generales

A modo de conclusión se listan los diferentes logros alcanzados y aportes realizados:

- Se logró estudiar analíticamente y desarrollar librerías para simular y analizar señales GPS sintéticas. Este trabajo permitirá validar el correcto funcionamiento de un *frontend*, estudiando la presencia de señales provenientes de los satélites y sus características.
- Partiendo de un nivel inicial básico de conocimiento de lenguajes de descripción de hardware se pudo ir avanzando en el uso de estructuras más complejas (como las FSMs) a fin de implementar los módulos presentados.
- Se profundizó el uso de lenguajes de descripción de hardware (VHDL), en especial, en lo referente a la síntesis de circuitos.
- Se desarrollaron bancos de prueba para los diferentes módulos permitiendo corregir errores de funcionamiento y validar la lógica implementada.
- Se utilizaron herramientas específicas (GIT) a fin de versionar las diferentes etapas de diseño y validación de los módulos planteados.
- Se logró un mejor conocimiento de la herramienta de síntesis y simulación (Xilinx ISE) así como de la estructura interna de la FPGA (memorias blockRAM, DCM, etc.)

### 5.2. Próximos pasos

Una vez se pueda contar con un *frontend* GPS, se podrían realizar pruebas sobre los módulos con señales reales. Una primera aproximación, seguramente incluiría registrar las señales provenientes del *frontend* y procesarlas con los *scripts* desarrollados a fin de realizar una validación con señales reales.

Con el objeto de probar los módulos descritos, será necesario implementar hardware que permita la comunicación, por ejemplo, con un microcontrolador para realizar comprobaciones y más adelante, implementar algoritmos de control.



# Bibliografía

- [1] Kaplan E. , Hegarty C. *Understanding GPS: principles and applications*. 2.<sup>a</sup> ed. Artech House, Inc., 2006. ISBN: 1-58053-894-0.
- [2] P. Chu. *FPGA Prototyping by VHDL Examples*. John Wiley & Sons, Inc., 2008. ISBN: 978-0-470-18531-5.
- [3] L. Couch. *Digital and Analog Communication Systems*. 8.<sup>a</sup> ed. Pearson Education Ltd., 2013. ISBN: 0-273-77421-2.
- [4] Department of Defense. *GPS Standard Positioning System Performance Standard*. 4.<sup>a</sup> ed. 2008. URL: <http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf>.
- [5] *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/](https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/policy/availability/) (visitado 10-11-2016).
- [6] 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.
- [7] *GPS - Control Segment*. URL: <http://www.gps.gov/systems/gps/control/> (visitado 10-11-2016).
- [8] C. Hegarty. *Analytical Model for GNSS Receiver Implementation Losses*. The Mitre Corporation. URL: [https://www.mitre.org/sites/default/files/pdf/09\\_3995.pdf](https://www.mitre.org/sites/default/files/pdf/09_3995.pdf).
- [9] 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).
- [10] *LogiCORE IP DDS Compiler v4.0*. Xilinx Inc. 1967. URL: [https://www.xilinx.com/support/documentation/ip\\_documentation/dds\\_ds558.pdf](https://www.xilinx.com/support/documentation/ip_documentation/dds_ds558.pdf).
- [11] 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).
- [12] J. Bao-yen Tsui. *Fundamentals of global positioning system receivers : a software approach*. 2.<sup>a</sup> ed. John Wiley & Sons, Inc., 2005. ISBN: 0-471-70647-7.
- [13] E. Weisstein. *Werner Formulas*. URL: <http://mathworld.wolfram.com/WernerFormulas.html> (visitado 10-11-2016).