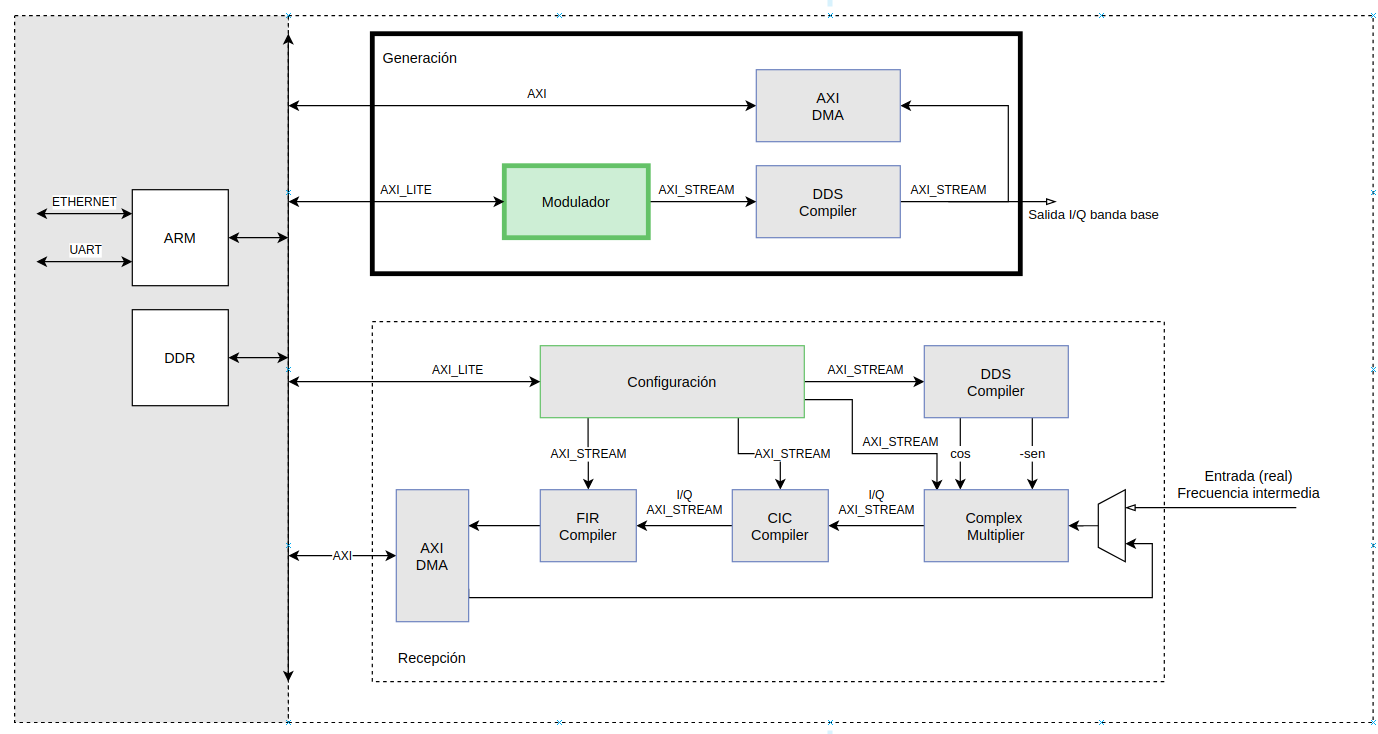
# Trabajo Práctido Final CESE 10ma cohorte

### Resumen

El bloque de hardware implementado comprende una parte del desarrollo del trabajo final de la CESE «Desarrollo de etapa digital para Radar Pulsado Multipropósito». El sistema en general se muestra en la imagen a continuación, y el presente trabajo es parte del bloque «Modulador» marcado en color verde.

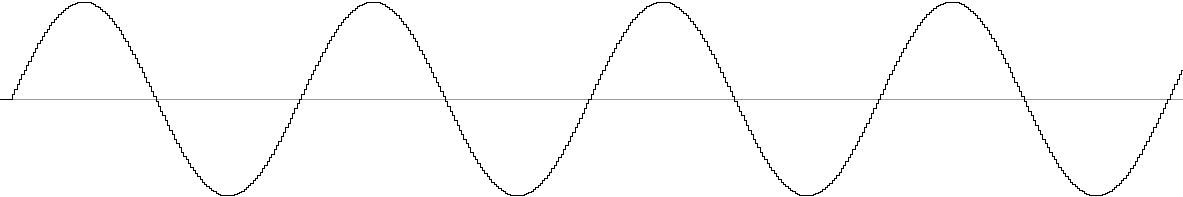


La funcionalidad del bloque de Generación es generar formas de onda típicamente usadas en radares, con combinaciones de formas de onda continuas, pulsadas, moduladas en frecuencia o en fase.

Para la síntesis de las señales se eligió en el trabajo final de la CESE utilizar el IP Core de Xilinx DDS Compiler, pero para la generación de las formas de onda mencionadas, era necesario configurar el DDS Compiler de forma dinámica para generar pulsos y variaciones en la frecuencia y fase instantánea de la señal de salida.

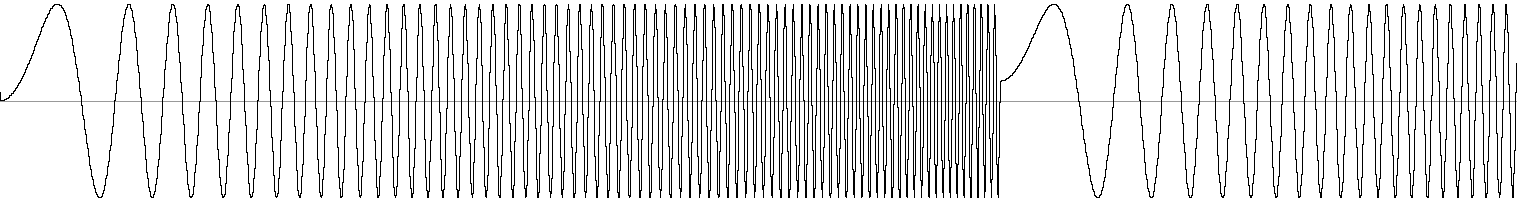
### Formas de onda - Continua

1) Continua



El proyecto estipula un radar con un ancho de banda de 20 MHz. Por lo tanto las ondas continuas a generar se deben poder configurar con un límite superior en 20MHz.

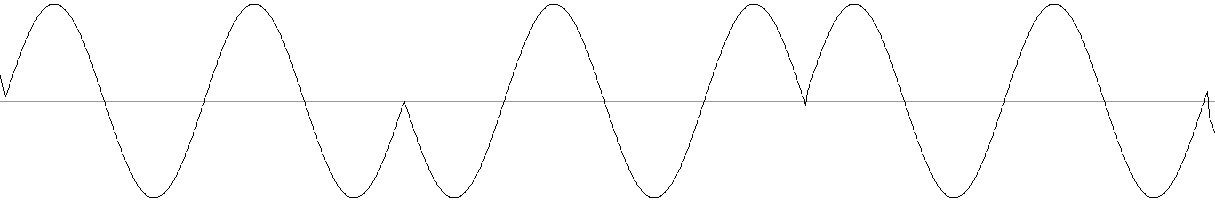
2) Continua modulada en frecuencia



Una de las modulaciones más empleadas en RADAR es la conocida como «chirp». Corresponde a una modulación lineal de la frecuencia instantánea de la señal.

La modulación implementada corresponde a una del tipo «up-chirp»

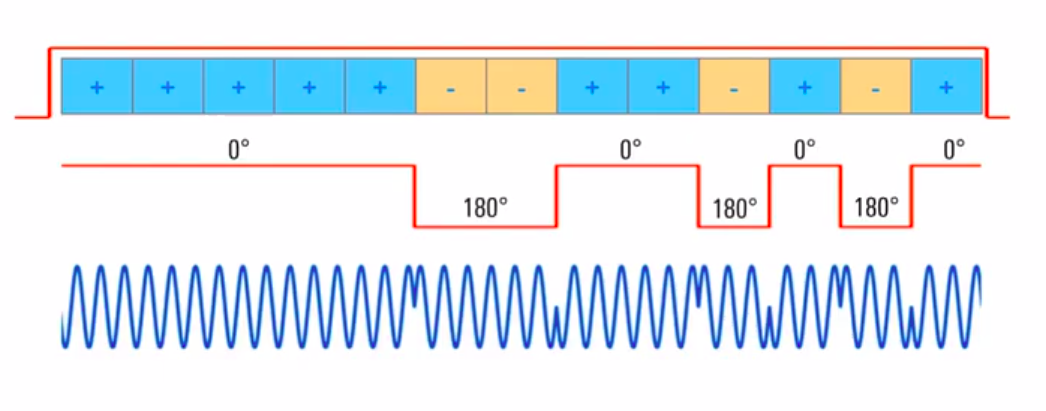
3) Continua modulada en fase



Los requerimientos del trabajo final establecen la implementación de una módulación en fasa conocida como «códigos Barker». Estos códigos corresponden con una secuencia de cambios de fase de una señal, que presentan un pico máximo de autocorrelación bien definido.

Los códigos son los siguientes, teniendo para un «1» y para un «-1» un cambio de fase de 180° de la señal senoidal.

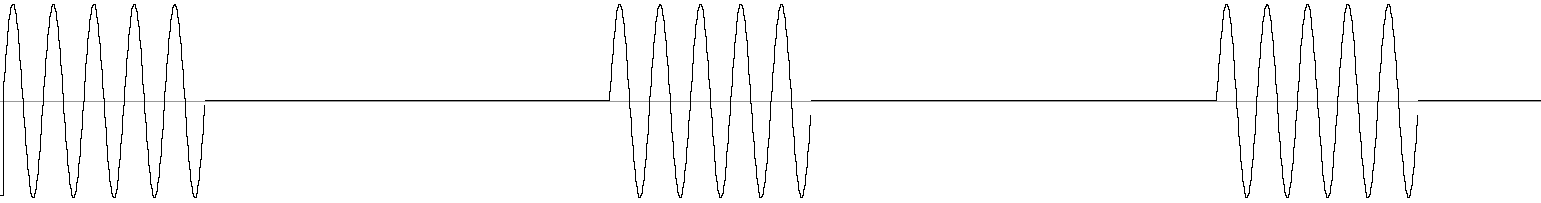
| Longitud | Código |
| --- | --- |
| 2 | + - |
| 3 | + + - |
| 4 | + - + + |
| 5 | + + + - + |
| 7 | + + + - - + - |
| 11 | + + + - - - + - - + - |
| 13 | + + + + + - - + + - + - + |



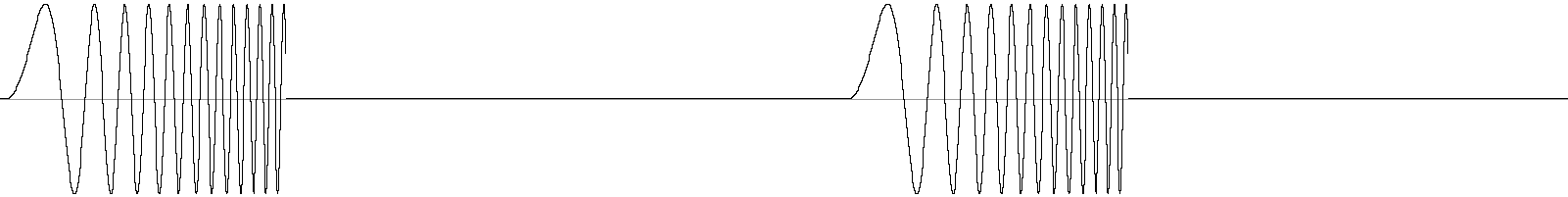
### Formas de onda - Pulsada

En este modo de operación se generan pulsos de las distintas formas de onda mostradas anteriormentes. Es necesario configurar el ancho de cada pulso, y el intervalo de repetición entre pulsos (período).

4) Pulsada sin modulación



5) Pulsada modulada en frecuencia

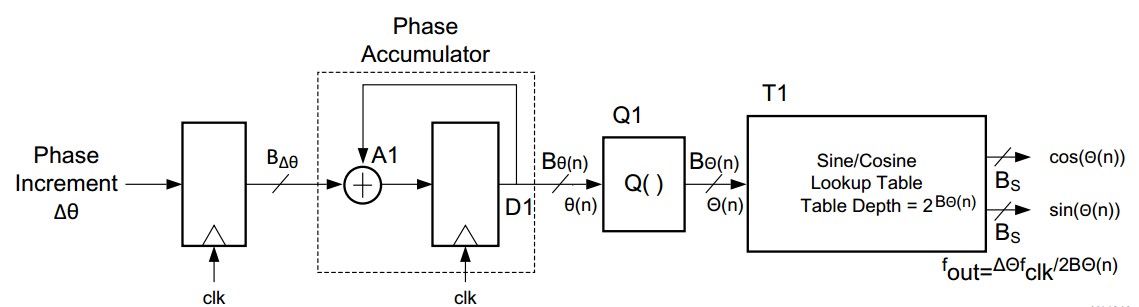


6) Pulsada modulada en fase

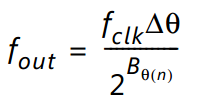


### DDS Compiler

El IP Core «DDS Compiler» es un sintetizador digital de señales senoidales (Direct Digital Synthesizer), que, básicamente, cuenta con un acumulador de fase y una memoria de look up.



A la salida del DDS se generan señales senoidales con una frecuencia dada por:



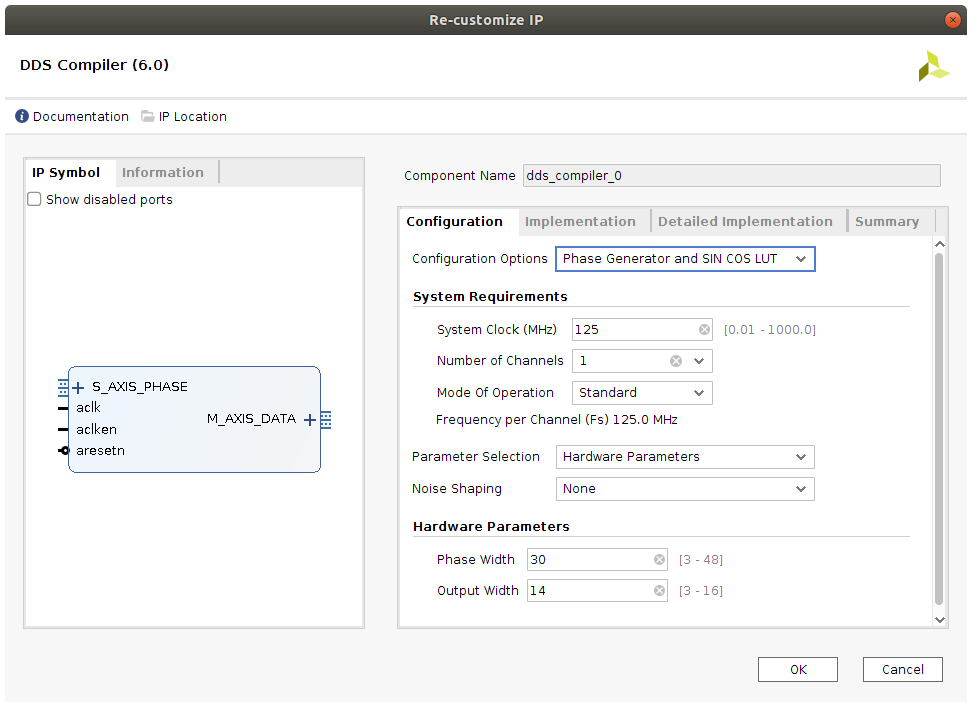
Con:

***fclk***: Frecuencia de reloj.

***Δθ***: Incremento de fase. Este es el parámetro de entrada principal, y nos va a dar la posibilidad de configurar dinámicamente la frecuencia de las señales generadas.

***Bθ(n)***: Cantidad de bits del acumulador de fase.

El bloque IP, se configuró según se muestra en la siguiente imagen:

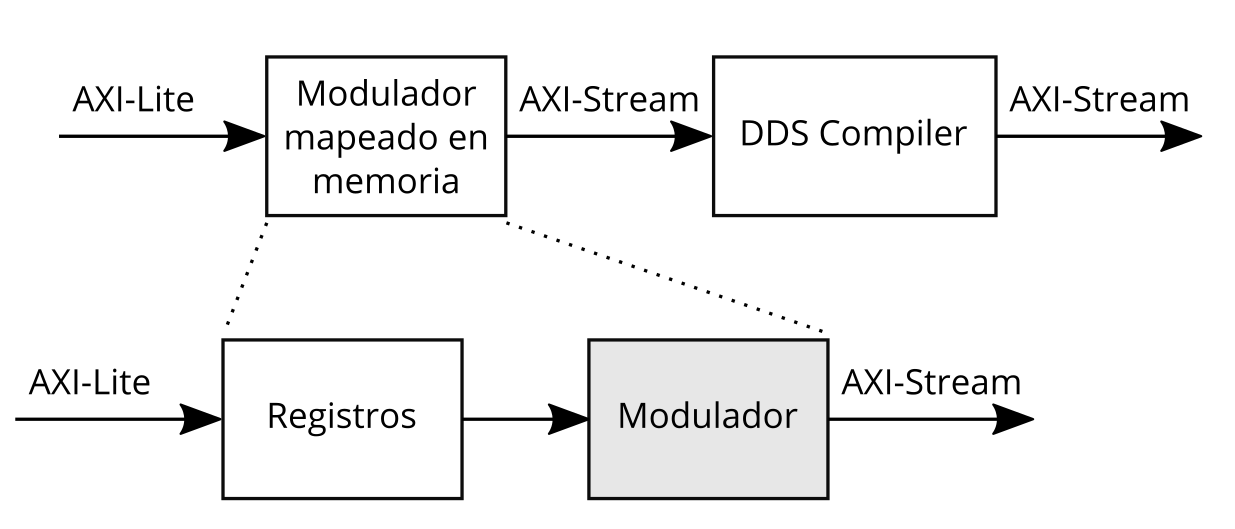


La interfaz de entrada y salida del Core es compatible con el bus AXI4-Stream, y es a través de este bus que el bloque de lógica desarrollado en el trabajo, configura la modulación de las señales.

### Bloque Modulador

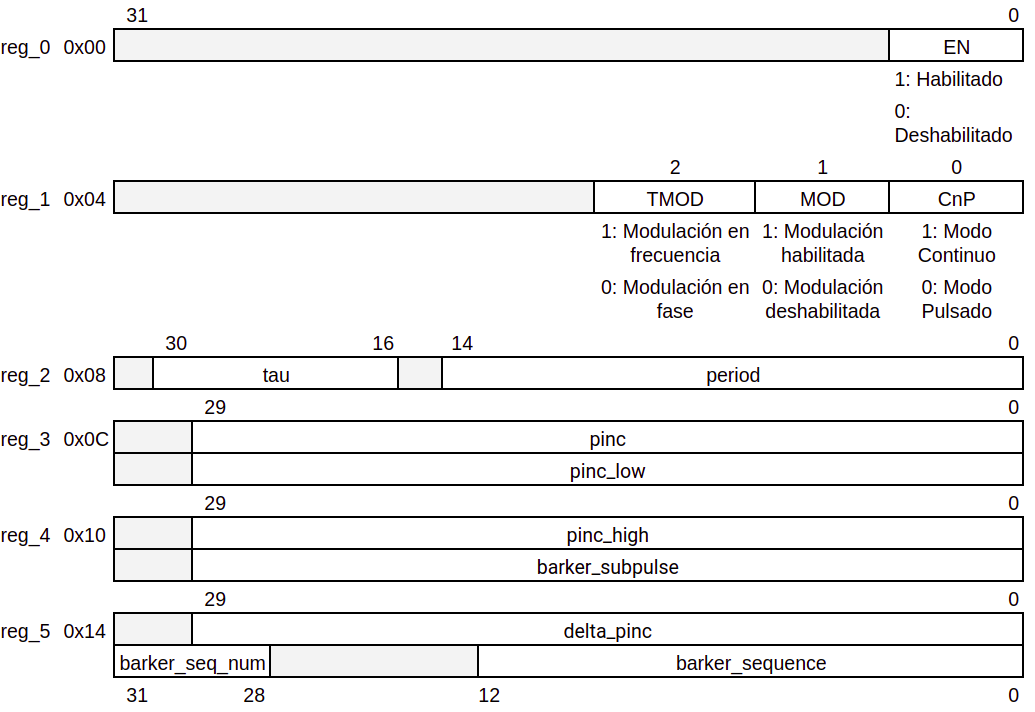
El bloque modulador que se muestra en este trabajo, forma parte de la sección de Generación del trabajo final, pero a su vez forma parte de un bloque que contiene una serie de registros de configuración mapeados a memoria y accesibles a través de un bus AXI-Lite.

El bloque desarrollado es el que se muestra en gris en la imagen a continuación.



### Mapa de memoria

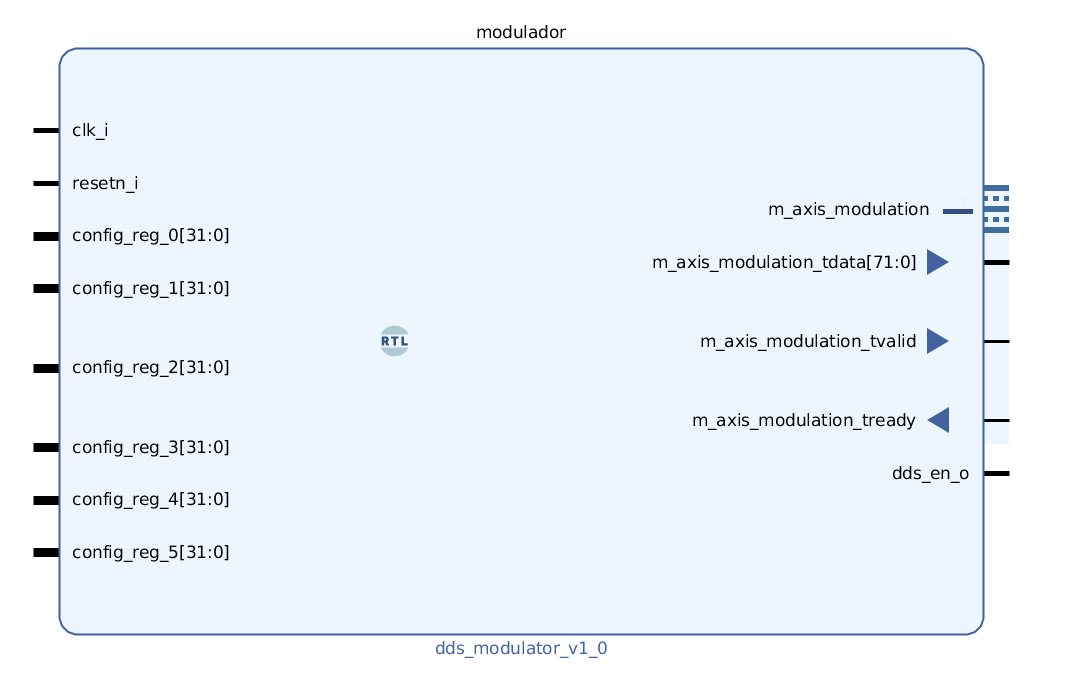
A través de los registros se establece la configuración del modulador, con los campos de cada registro organizados de la siguiente manera:



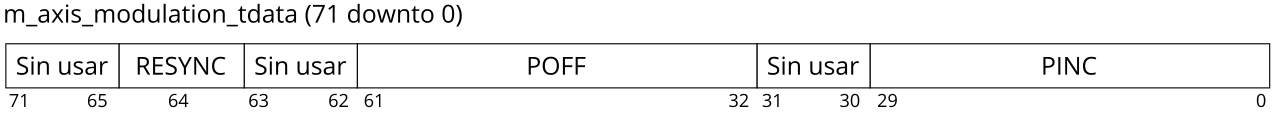
| reg\_0 | Habilitación/Deshabilitación del modulador |
| --- | --- |
|  |  |
| reg\_1 | Bits de configuración de modo continuo o pulsado, y tipo de modulación. |
|  |  |
| reg\_2 | **period:** En modo pulsado, Período del pulso |
|  | **tau:** En modo pulsado, Ancho del pulso |
|  |  |
| reg\_3 | **pinc:** En modo continuo o modulado en fase, Incremento de fase (constante) para el DDS |
|  | **pinc\_low:** En modo modulado en frecuencia, equivale al incremento de fase inicial del DDS |
|  |  |
| reg\_4 | **pinc\_high:** En modo modulado en frecuencia, equivale al incremento de fase final del DDS |
|  | **barker\_subpulse:** En modo modulado en fase, Ancho de cada subpulso |
|  |  |
| reg\_5 | **delta\_pinc:** En modo modulado en frecuencia, equivale a la pendiente de cambio de la fase del DDS. |
|  | **barker\_sequence:** En modo modulado en fase, Código binario correspondiente al código Barker**.** |
|  | **barker\_seq\_num:** En modo modulado en fase, Número de código Barker**.** |

### Señales de entrada y salida

- Los registros de configuración se conectan directamente al módulo mediante las señales *config\_reg\_N*.



- Según la configuración elegida, las señales de entrada al DDS a comandar con el modulador, son las siguientes, y forman parte del bus AXI-Stream *m\_axis\_modulation*.



Los campos de la señal *m\_axis\_modulation\_tdata* representan lo siguiente:

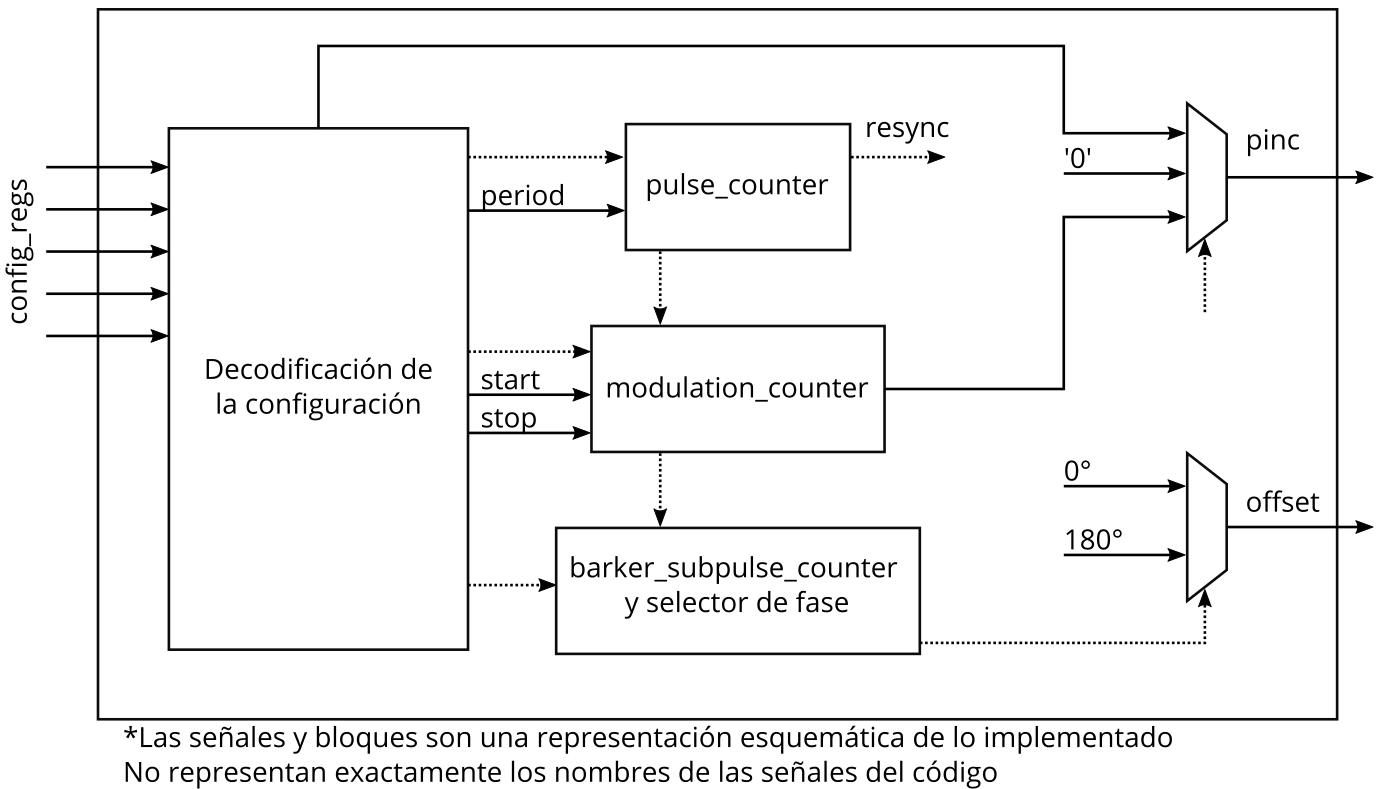
- **PINC**: Configuración externa del incremento de fase del DDS. **Este parámetro setea la frecuencia instantánea de salida.**

- **POFF**: Configuración externa del cambio de fase del DDS. Este parámetro suma una fase constante al acumlador de fase del DDS. **Este parámetro se utiliza para generar cambios de fase.**

- **RESYNC**: Bandera que inhibe el funcionamiento del acumulador del DDS, y establece su fase interna según el valor de PINC. **Esta señal de control es utilizada para generar los pulsos en modo pulsado, inhibiendo la generación de señales y "resetando" a un valor inicial para cada pulso.**

- La señal *dds\_en\_o* actúa como "clock enable" del core DDS.

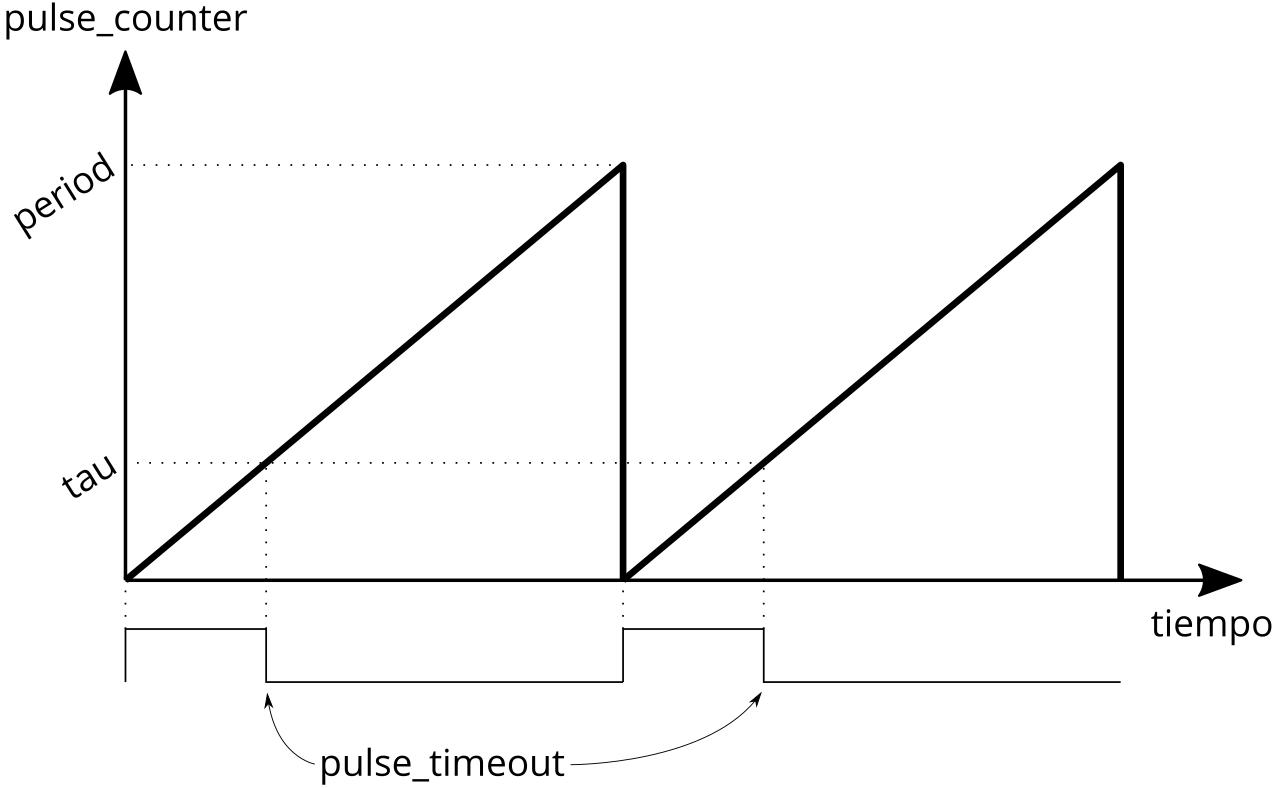
## Arquitectura interna del bloque modulador



### Modo Pulsado vs Modo Continuo

En modo pulsado se generan señales por un período de tiempo y luego se inhibe la generación hasta iniciar el próximo período. Esto se realiza mediante el contador *pulse\_counter.*

El período y ancho de pulso se configuran externamente, y una vez habilitado el contador cuenta periódicamente, reseteandose en la cuenta máxima, y generando una señal *pulse\_timeout\_n* que indica si se llegó o no al ancho de pulso.



***Pulse Counter: Representación de la cuenta para la generación de pulsos***

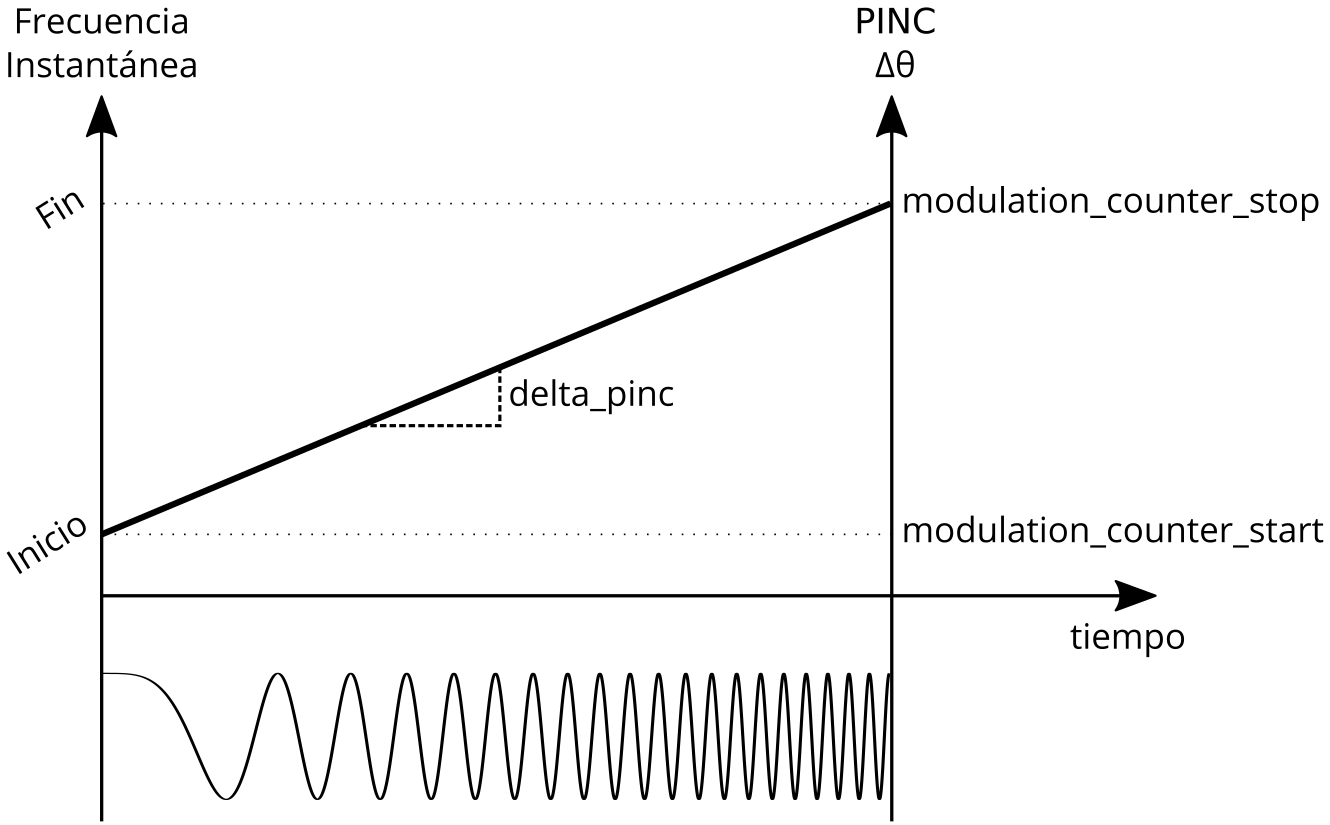
La señal de timeout del contador es utilizada para inhibir la generación del DDS mediante la señal RESYNC del bus de configuración.

En modo contínuo, el contador de pulsos está deshabilitado.

### Modulación en frecuencia

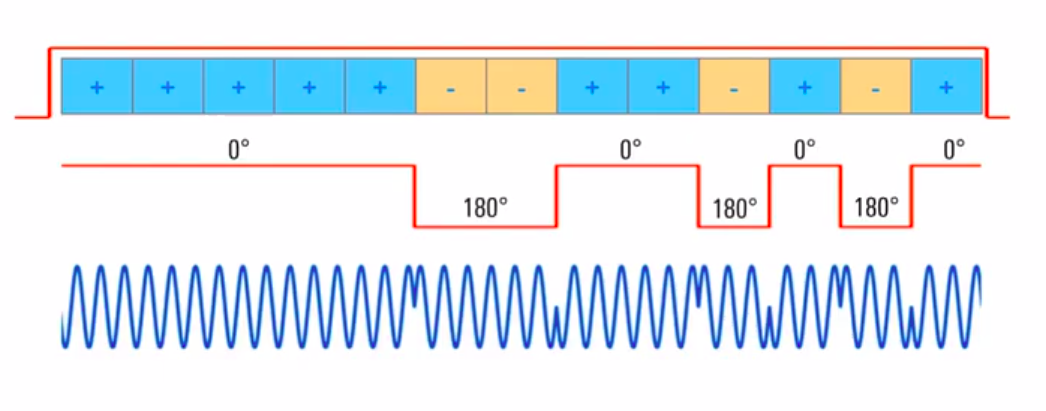
Para variar linealmente la frecuencia de salida del DDS se debe generar una rampa que se incremente desde el valor inicial al final de incremento de fase (*Δθ*, ó PINC). Estos valores estarán dados por el ancho de banda de la modulación en frecuencia que se desee.

La rampa se implementó mediante otro contador (modulation\_counter). El bloque implementado recibe a través de las señales de configuración el valor de *Δθ* de inicio y fin, y el valor de incremento del contador de modulación (delta\_pinc) que corresponde a la pendiente de la rampa que se muestra en la imagen siguiente:



Para poder operar en conjunto con el modo pulsado, el contador de modulación es habilitado/deshabilitado por el contador de pulsos en caso que el modo sea: "Pulsado con modulación en frecuencia".

### Modulación en fase

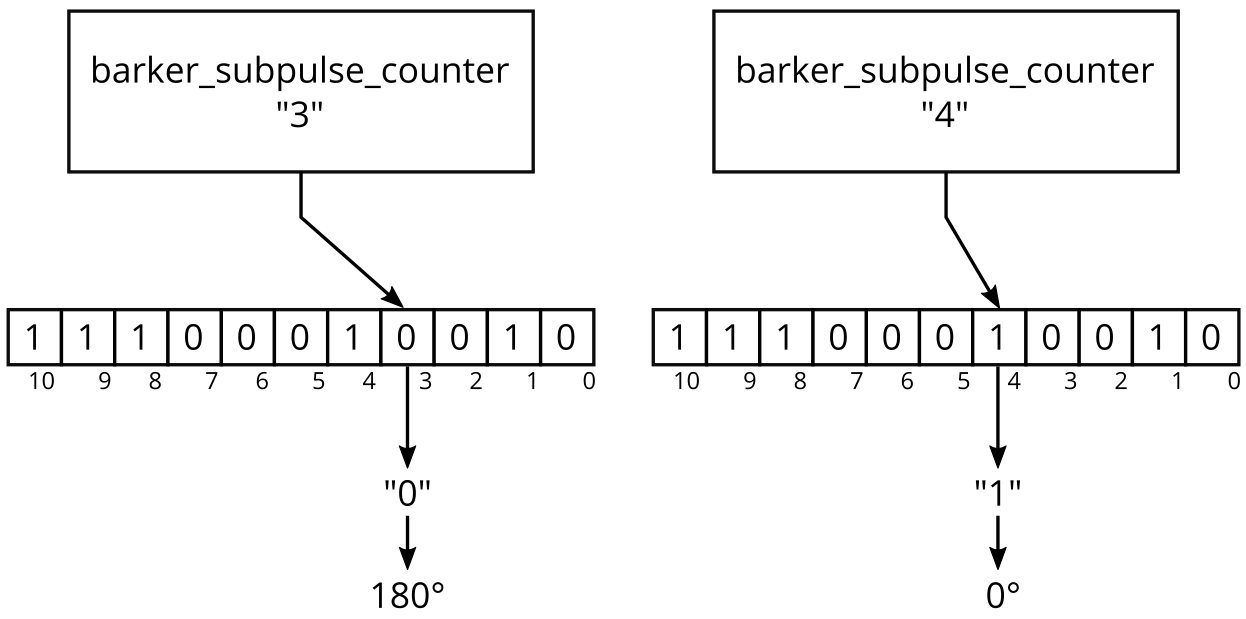


La modulación en fase requiere generar cambios de fase de 180° según una secuencia de bits que el módulo recibe a través de una de las entradas de configuración. Como el DDS Compiler cuenta con una entrada de OFFSET específica para aplicar el defasaje, el modulador únicamente debe colocar en la estructura de datos del bus AXI la fase correcta en cada instante de la secuencia de modulación.

Se necesita poder generar todas las secuencias barker vistas anteriormente, por lo que al ser de longitud variable, el modulador recibe además del código correspondiente a la secuencia, un valor que indica qué número de secuencia es (2, 3, 4, 5, 7, 11, ó 13), y la duración en tiempo (ciclos de reloj) que se desea para cada "subpulso" de la secuencia. Por ejemplo, una secuencia de longitud 7 bits, posee 7 subpulsos. Estableciendo una longitud de subpulso de 2 microsegundos, la secuencia total tendrá una duración de 14 microsegundos.

Para saber qué defasaje aplicar se implementó un contador para saber cuál es el subpulso actual, extraer el valor del bit de la secuencia correspondiente y en función de ese valor aplicar o no el cambio de fase.

Este tercer contador es habilitado/deshabilitado una vez que se cumple el tiempo correspondiente a cada subpulso. Se tomó la decisión de utilizar el mismo contador para contar la longitud de los subpulsos que para la modulación en frecuencia (modulation\_counter), dado que ambas modulaciones no pueden estar activas simultáneamente.



## Tests en simulación

## Test con ILA