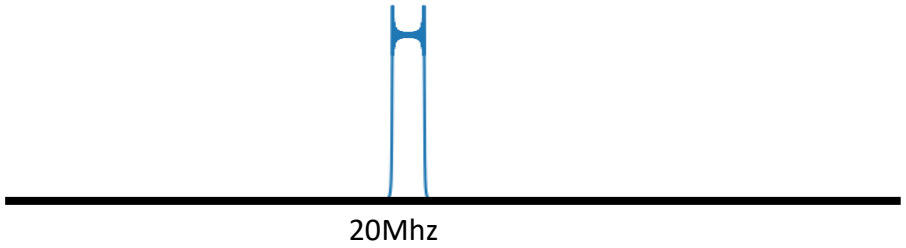
Modulador FS/4

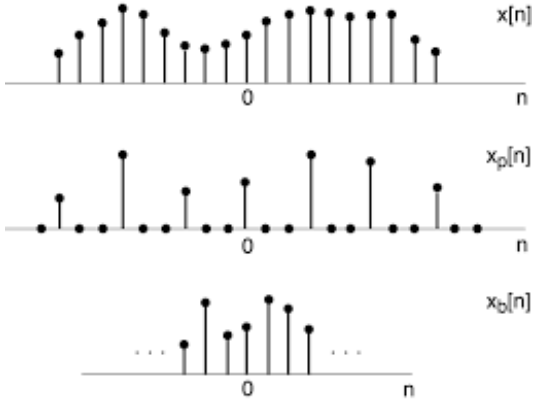
BACH. CARLOS HERRERA

Objetivo

Reducción de la frecuencia a la que se está trabajando con la señal

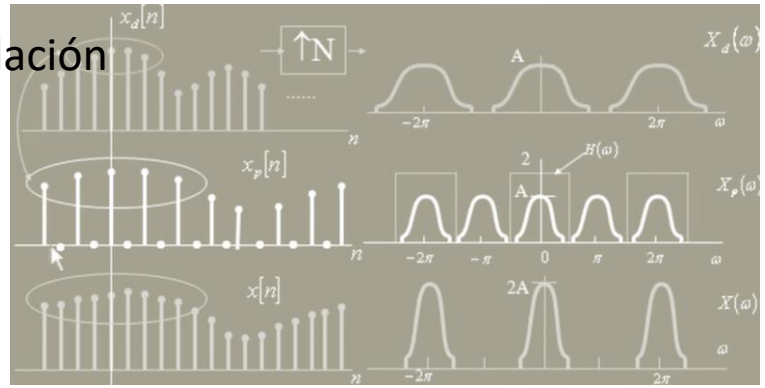


$$I'(N) = I(N)\cos(w * t) - Q(N)\sin(w * t)$$
$$Q'(N) = I(N)\sin(w * t) + Q(N)\cos(w * t)$$

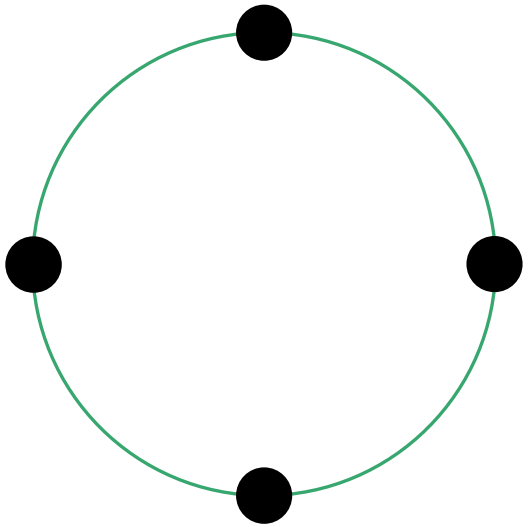


Decimación

Filtro interpolación



Valores sumas y restas



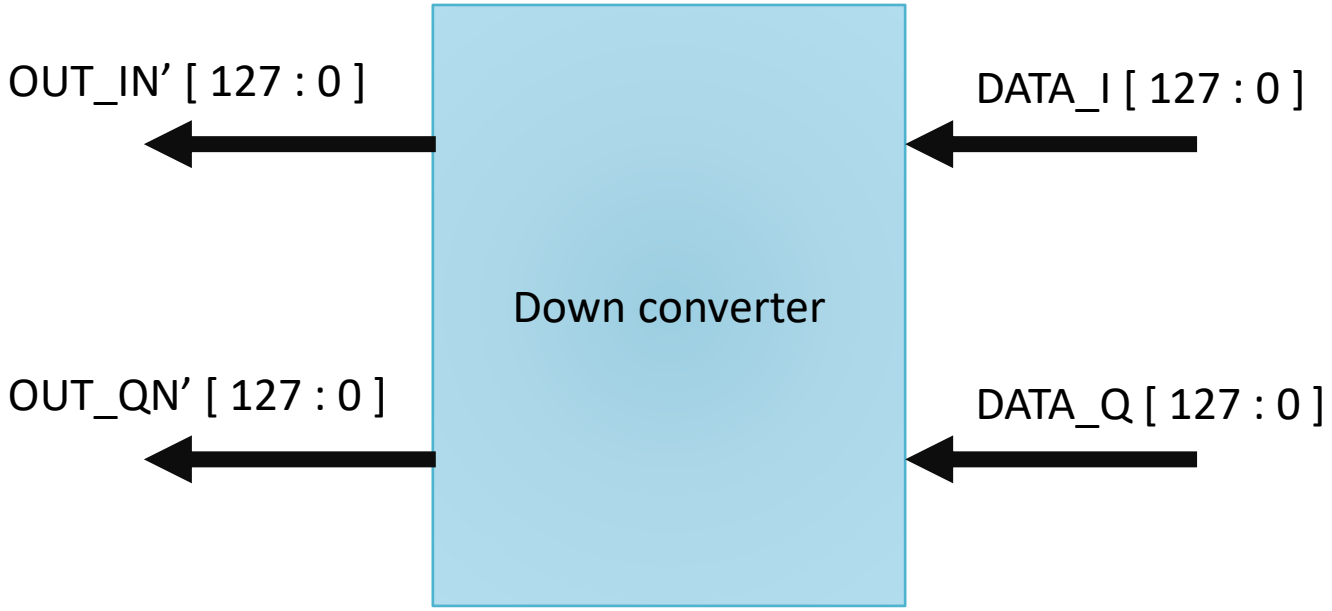
	$\mathbf{0}$	$\mathbf{\pi/_2}$	$\boldsymbol{\pi}$	$\mathbf{3\pi/_2}$
Seno	0	1	0	-1
Coseno	1	0	-1	0

$$I'(N) = I(N)\cos(w * t) - Q(N)\sin(w * t)$$
$$Q'(N) = I(N)\sin(w * t) + Q(N)\cos(w * t)$$

	$I'(N)$	$Q'^{(N)}$
$\mathbf{0}$	$I(N) * 1 - Q(N) * 0$	$I(N) * 0 + Q(N) * 1$
$\mathbf{\pi/_2}$	$I(N) * 0 - Q(N) * 1$	$I(N) * 1 + Q(N) * 0$
$\boldsymbol{\pi}$	$I(N) * -1 - Q(N) * 0$	$I(N) * 0 + Q(N) *$ -1
$\mathbf{3\pi/_2}$	$I(N) * 0 - Q(N) * -1$	$I(N) * -1 + Q(N) * 0$

Diagrama de bloques

8 muestras de 16bits tipo int16 de una misma señal llega hacia el bloque de procesmiento



Esquemático Vivado

Entradas:

clk_i

reset_i

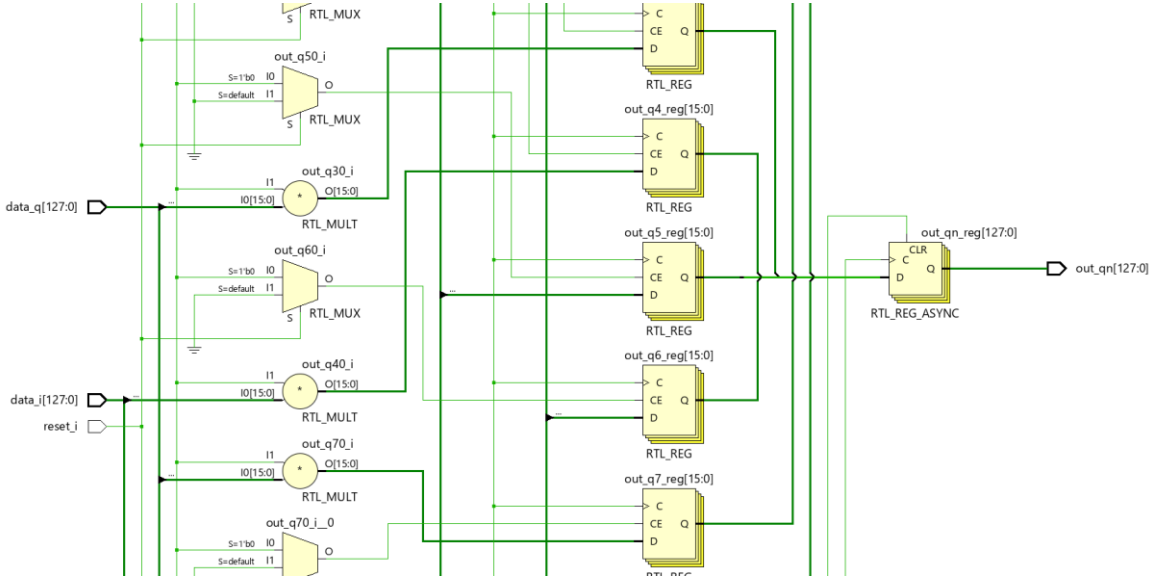
data_i

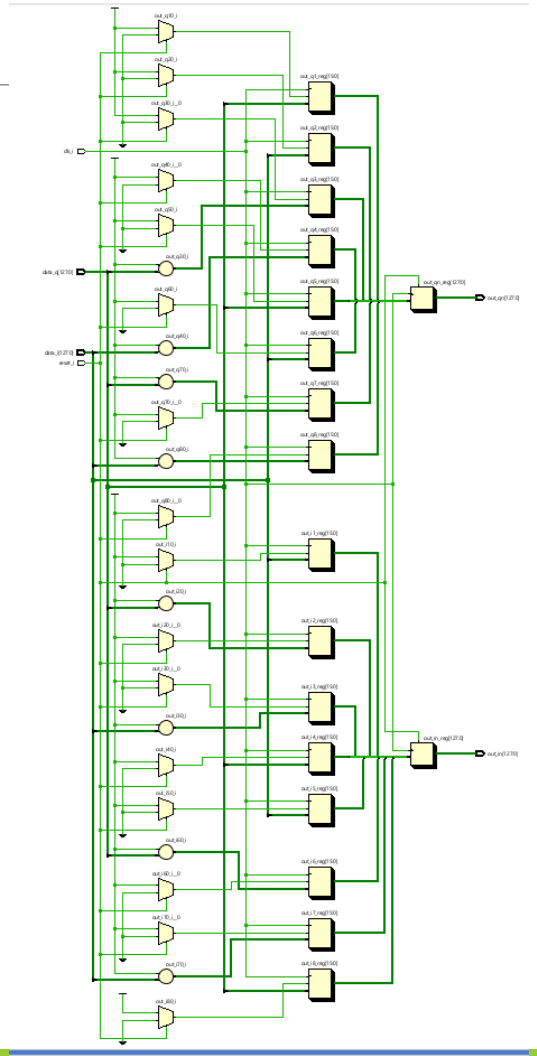
data_q

Salidas

out_in

out_qn

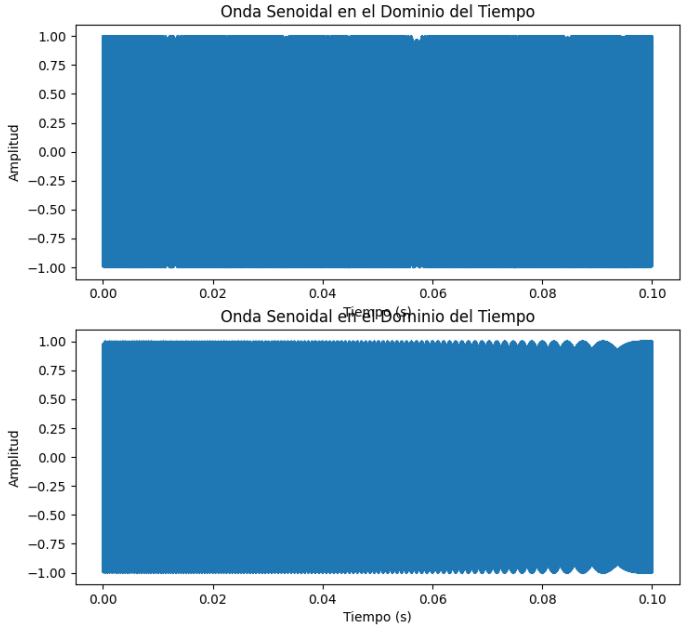


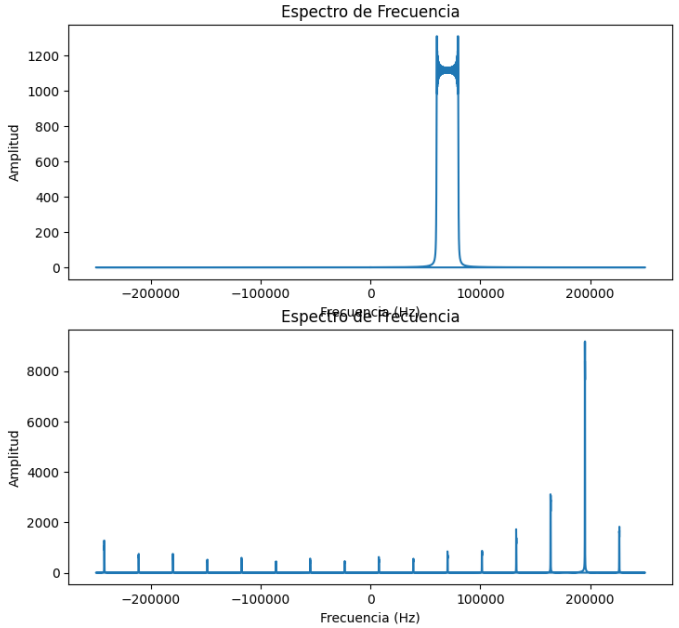


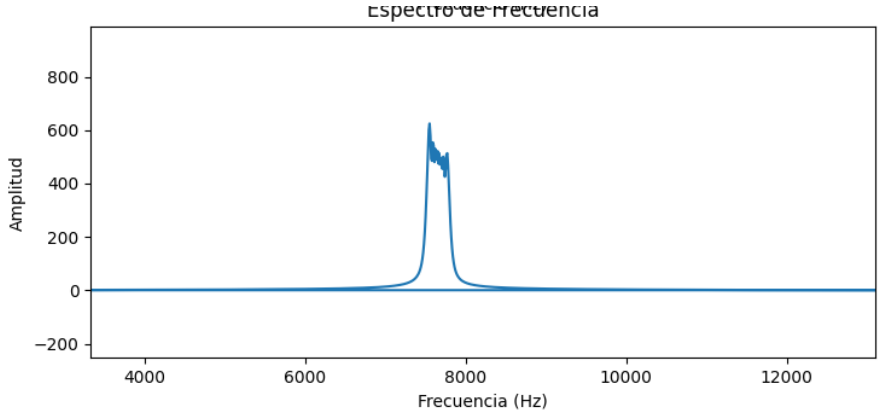
VHDL

```
-- 0 GRADOS
-- out_i1 <= std_logic_vector(to_signed( (to_integer(signed(sample_i1)) * 1 - to_integer(signed(sample_q1)) * 0) , DATA_IN_WIDTH ));
-- out_q1 <= std_logic_vector(to_signed( (to_integer(signed(sample_i1)) * 0 + to_integer(signed(sample_q1)) * 1) , DATA_IN_WIDTH ));
out_i1 <= std_logic_vector(to_signed(to_integer(signed(data_i(127 downto 112))), DATA_IN_WIDTH));
out_q1 <= std_logic_vector(to_signed(to_integer(signed(data_q(127 downto 112))), DATA_IN_WIDTH));

-- pi/2
-- out_i2 <= std_logic_vector(to_signed( (to_integer(signed(sample_i1)) * 0 - to_integer(signed(sample_q1)) * 1) , DATA_IN_WIDTH ));
-- out_q2 <= std_logic_vector(to_signed( (to_integer(signed(sample_i1)) * 1 + to_integer(signed(sample_q1)) * 0) , DATA_IN_WIDTH ));
out_i2 <= std_logic_vector(to_signed( to_integer(signed(data_q ( 111 downto 96))) * to_integer(to_signed(-1,DATA_IN_WIDTH)), DATA_IN_
out_q2 <= std_logic_vector(to_signed( to_integer(signed(data_i ( 111 downto 96))) , DATA_IN_WIDTH ));

-- pi
-- out_i3 <= std_logic_vector(to_signed( (to_integer(signed(sample_i1)) * -1 - to_integer(signed(sample_q1)) * 0) , DATA_IN_WIDTH ));
-- out_q3 <= std_logic_vector(to_signed( (to_integer(signed(sample_i1)) * 0 + to_integer(signed(sample_q1)) * -1) , DATA_IN_WIDTH ));
out_i3 <= std_logic_vector(to_signed( to_integer(signed(data_i ( 95 downto 80))) *  to_integer(to_signed(-1,DATA_IN_WIDTH)) , DATA_IN_
out_q3 <= std_logic_vector(to_signed( to_integer(signed(data_q ( 95 downto 80))) *  to_integer(to_signed(-1,DATA_IN_WIDTH)) , DATA_IN_

-- 3pi/2
-- out_i4 <= std_logic_vector(to_signed( (to_integer(signed(sample_i1)) * 0 - to_integer(signed(sample_q1)) * -1) , DATA_IN_WIDTH ));
-- out_q4 <= std_logic_vector(to_signed( (to_integer(signed(sample_i1)) * -1 + to_integer(signed(sample_q1)) * 0) , DATA_IN_WIDTH ));
out_i4 <= std_logic_vector(to_signed( to_integer(signed(data_q ( 79 downto 64))) , DATA_IN_WIDTH ));
out_q4 <= std_logic_vector(to_signed( to_integer(signed(data_i ( 79 downto 64))) *  to_integer(to_signed(-1,DATA_IN_WIDTH))  , DATA_I
```

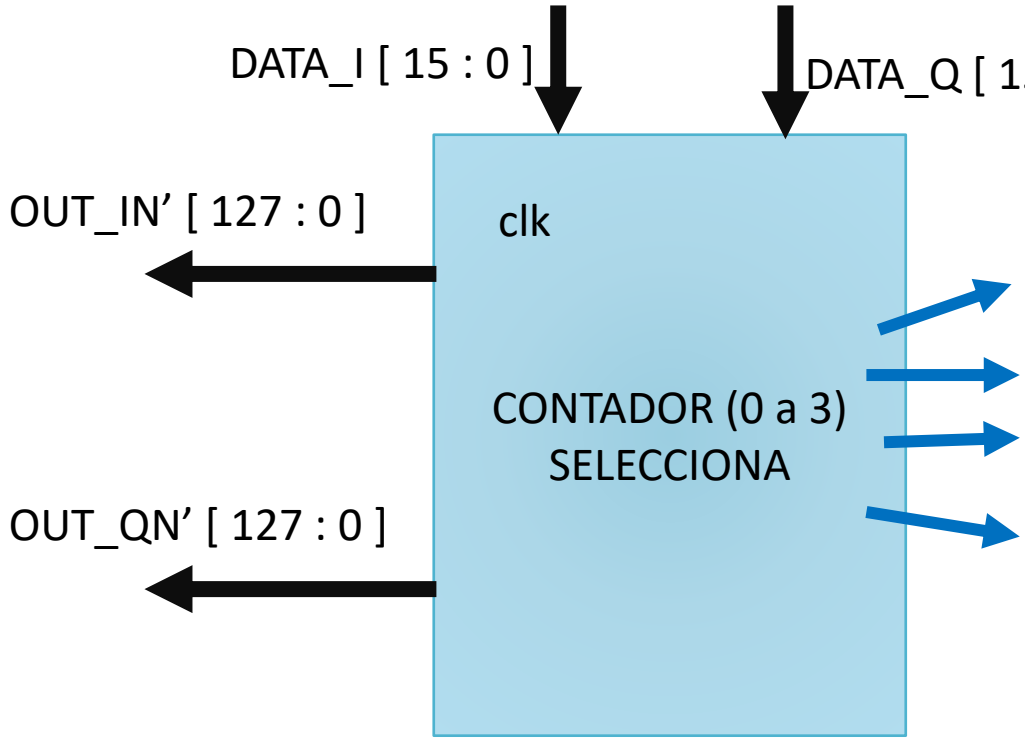
Grafica de la señal







Alternativa



		$I'(N)$	$Q'^{(N)}$
00	$\mathbf{0}$	$I(N) * 1 - Q(N) * 0$	$I(N) * 0 + Q(N) * 1$
01	$\pi/_2$	$I(N) * 0 - Q(N) * 1$	$I(N) * 1 + Q(N) * 0$
10	$\boldsymbol{\pi}$	$I(N) * -1 - Q(N) * 0$	$I(N) * 0 + Q(N) * -1$
11	$3\pi/_2$	$I(N) * 0 - Q(N) * -1$	$I(N) * -1 + Q(N) * 0$