Transmisor

Documento que indica todo lo necesario para usar el transmisor **IEEE_8021513_TX**.

El IP Core se encuentra en esta carpeta: IP Core Tx.

El proyecto ejemplo de Vivado donde se corrió la simulación: Ejemplo Tx.

Funcionalidades probadas

- Máximo tamaño transmisible por trama: 4096 bytes.
- Acepta transmisión de múltiples tramas consecutivas, sin necesidad de reset. La señal de nuevo frame no debe recibirse mientras la señal de "new_msg_ready" esté en "0".

Clocks

- **clk_dac**: [125 MHz]. Clock físico del DAC, conectado a la entrada del clocking wizard.
- **clk_tx**: [125 MHz]. Clock del transmisor, salida del clocking wizard. Sincrónico con la salida del transmisor.
- **clk_fifo_m**: [15.625 MHz]. Clock para sacar datos de la FIFO, salida del clocking wizard. Sincrónico con la entrada del transmisor.

Entradas

- **IPCORE_CLK**: [clk]. Señal de clock de 125 MHz.
- **IPCORE_RESETN**: [bool]. Señal de reset ACTIVE LOW ('0' para resetear).
- **new_frame_in**: [bool]. Indica que hay un nuevo mensaje a transmitir. Lee solamente el flanco ascendente de la señal.
- **[reg0, reg1, reg2, reg3]**: [uint32_t]. Registros de configuración.
- **data_in**: [uint8_t]. Datos a transmitir. Se espera que sean recibidos de una interfaz AXI4 Stream de 8bits.
- **valid_in**: [bool]. Momento en que los datos recibidos son válidos (señal de AXI4 Stream).

Registros

Al recibir la señal "new_frame_in", se van a leer los registros de 32bits (reg0, reg1, reg2 y reg3) durante solamente un ciclo de clock, por lo que los registros pueden cambiar de valor durante la transmisión de un mensaje. Los registros quedan definidos como sigue:

Register	**31**	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
reg0	x	x	x	x	x	x	x	x	p23	p22	p21	p20	p19	p18	p17	p16
reg1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
reg2	x	x	x	x	x	concat2	concat1	concat0	x	x	x	x	x	rep2	rep1	rep0
reg3	x	x	mimon2	mimon1	mimon0	mimos2	mimos1	mimos0	x	x	x	x	x	cp2	cp1	cp0

Register	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x00	p15	p14	p13	p12	p11	p10	p9	p8	p7	p6	p5	p4	p3	p2	p1	p0
0x04	m15	m14	m13	m12	m11	m10	m9	m8	m7	m6	m5	m4	m3	m2	m1	m0
0x08	x	x	x	x	x	rate2	rate1	rate0	x	x	x	x	x	x	block1	block0
0x0c	x	x	x	bat4	bat3	bat2	bat1	bat0	x	x	x	x	si3	si2	si1	si0

- **p[23:0]**: *psduSize*. Tamaño en bytes del mensaje a transmitir.
- **m[15:0]**: *messageDuration*. En vez de usarse para indicar el tiempo que demora la transmisión, este parámetro se usa para indicar la cantidad de bytes "extra" agregados en la transmisión, para que sea múltiplo de "payloadBitsPerBlock0 = 21".

  Por ejemplo: si su mensaje es de 30 bytes, entonces (psduSize = 30; messageDuration = 9).

  Si su mensaje es de 300 bytes, entonces (psduSize = 300; messageDuration = 6).

  Si bien el mensaje "real" tiene un tamaño fijo, el mensaje escrito en la FIFO de entrada debe ser un múltiplo de 21 bytes, agregando bytes nulos para completar el múltiplo de 21.

- **block[1:0]**: *blockSize*. Siempre "00".
- **rate[2:0]**: *fecRate*. Siempre "001".
- **rep[2:0]**: *repetitionNumber*. Siempre "001".
- **concat[2:0]**: *fecConcatenationFactor*. Siempre "000".
- **si[3:0]**: *scramblerInitialization*. Cualquier valor (testeado con "1111").
- **bat[4:0]**: *batId*. Siempre "00010".
- **cp[2:0]**: *cyclicPrefixId*. Cualquier valor menos "000". (testeado con "001").
- **mimos[2:0]**: *explicitMimoPilotSymbolCombSpacing*. Cualquier valor (se puede usar para cualquier cosa).
- **mimon[2:0]**: *explicitMimoPilotSymbolNumber*. Cualquier valor (se puede usar para cualquier cosa).

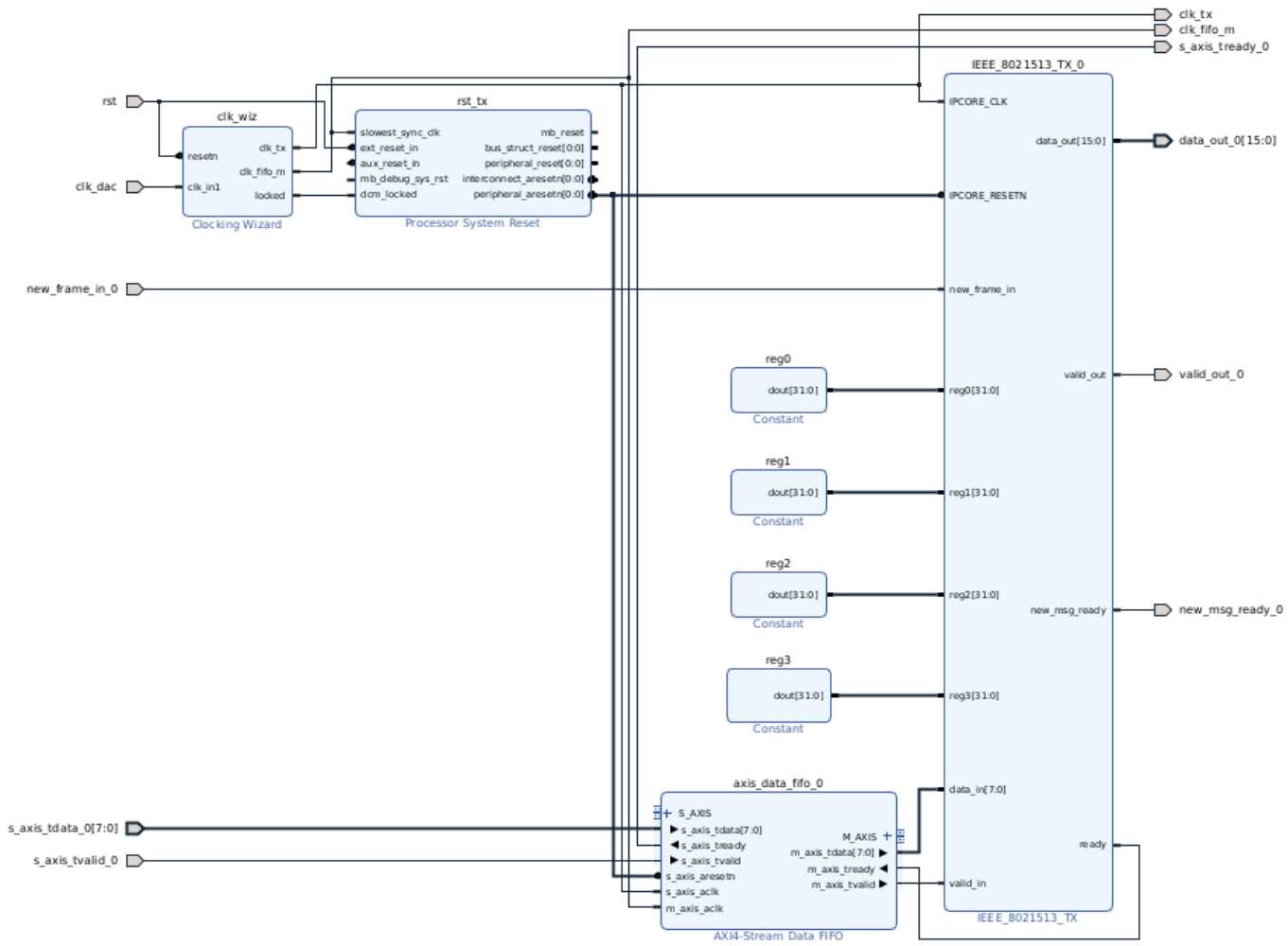
Outputs

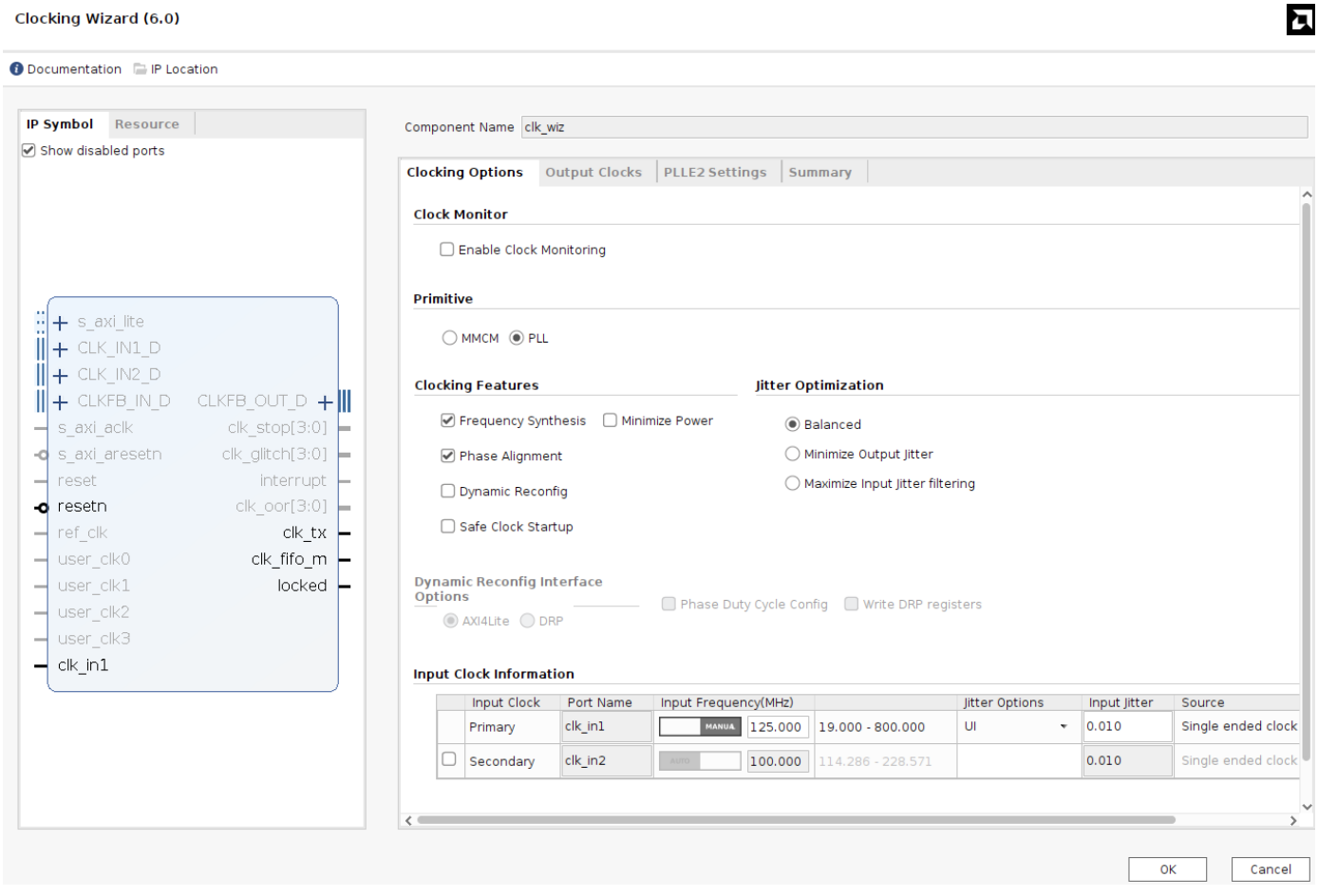
- **data_out**: [int16]. Valores de salida, para el DAC. Los dos bits MSB no se usan, por lo que toma valores entre [-8192; 8191].
- **valid_out**: [bool]. Indica que el valor de salida es válido.
- **new_msg_ready**: [bool]. Indica que el bloque está preparado para recibir un nuevo mensaje. Esta salida responde al siguiente comportamiento:
  - Luego del reset, empieza en "1".
  - Al recibir en la entrada un `new_frame_in`, se pone en "0".
  - Cuando se termine de enviar el símbolo OFDM actual, se pone en "1" nuevamente, indicando que está listo para recibir un nuevo símbolo.
- **ready**: [bool]. Señal del AXI4-Stream. Indica que está listo para leer de la FIFO los datos del payload.

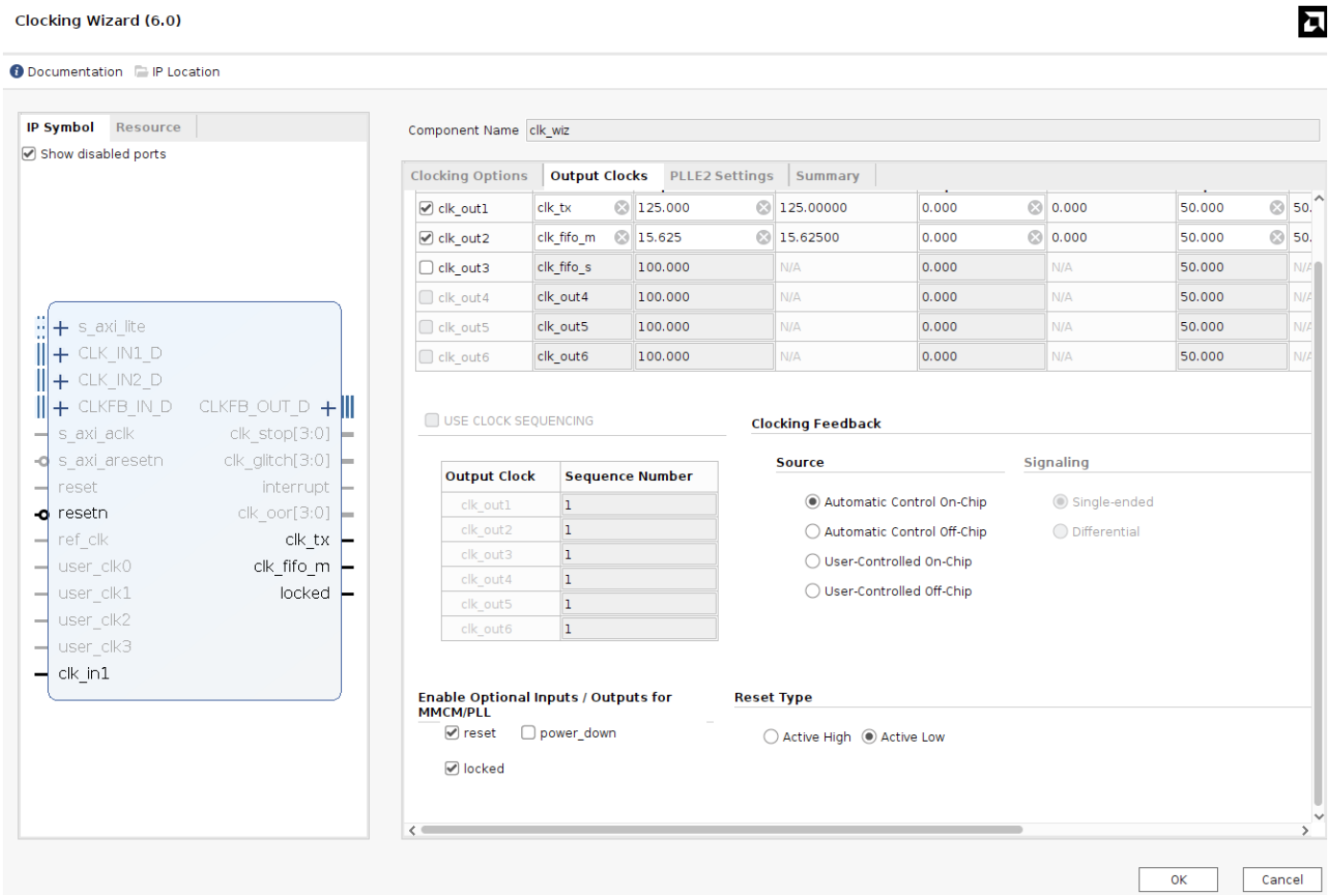
Modo de uso

1. Escribir en la FIFO el mensaje a transmitir. Si bien el mensaje puede ser de "x" bytes (incluyendo 0 bytes), tenga en cuenta que lo que se escriba en la FIFO debe ser un múltiplo de 21 bytes (completar con '0' de ser necesario).
2. Setear los registros reg0, reg1, re2 y reg3.
3. Leer la señal `new_msg_ready`, y esperar hasta que esté en "1".
4. Levantar la señal `new_frame_in` durante un ciclo de clock de "clk_fifo_s". A partir de este punto, los registros pueden ser modificados sin problemas.
5. Esperar mientras se procesan el preámbulo y encabezado.
6. Se va a levantar la señal de `ready` y va a empezar a leer la FIFO la cantidad de bytes indicada por los registros.
7. Esperar mientras se forma el símbolo OFDM.
8. Se envía a la salida una señal continua de 125MHz lista para conectarse al DAC. Se indica su validez con la señal `valid_out`.
9. No se puede levantar otra señal de `new_frame_in` hasta que `new_msg_ready` valga '1'.

Block Design


clk_tx
clk_fifo_m
s_axis_tready_0
IEEE_8021513_TX_0
IPCORE_CLK
data_out[15:0]
data_out_0[15:0]
IPCORE_RESETN
rst
clk_wiz
resetn
clk_in1
clk_tx
clk_fifo_m
locked
Clocking Wizard
clk_dac
rst_tx
slowest_sync_clk
ext_reset_in
aux_reset_in
mb_debug_sys_rst
dcm_locked
mb_reset
bus_struct_reset[0:0]
peripheral_reset[0:0]
interconnect_aresetn[0:0]
peripheral_aresetn[0:0]
Processor System Reset
new_frame_in_0
new_frame_in
reg0
dout[31:0]
Constant
reg0[31:0]
valid_out
valid_out_0
reg1
reg1[31:0]
reg2
reg2[31:0]
new_msg_ready
new_msg_ready_0
reg3
reg3[31:0]
axis_data_fifo_0
data_in[7:0]
S_AXIS
s_axis_tdata[7:0]
s_axis_tready
s_axis_tvalid
s_axis_aresetn
s_axis_aclk
m_axis_aclk
M_AXIS
m_axis_tdata[7:0]
m_axis_tready
m_axis_tvalid
ready
valid_in
s_axis_tdata_0[7:0]
s_axis_tvalid_0
AXI4-Stream Data FIFO
IEEE_8021513_TX


Clocking Wizard (6.0)
Documentation
IP Location
IP Symbol
Resource
Show disabled ports
s_axi_lite
CLK_IN1_D
CLK_IN2_D
CLKFB_IN_D
CLKFB_OUT_D
s_axi_aclk
clk_stop[3:0]
s_axi_aresetn
clk_glitch[3:0]
reset
interrupt
resetn
clk_oor[3:0]
ref_clk
clk_tx
user_clk0
clk_fifo_m
user_clk1
locked
user_clk2
user_clk3
clk_in1
Component Name
clk_wiz
Clocking Options
Output Clocks
PLLE2 Settings
Summary
Clock Monitor
Enable Clock Monitoring
Primitive
MMCM
PLL
Clocking Features
Frequency Synthesis
Minimize Power
Phase Alignment
Dynamic Reconfig
Safe Clock Startup
Jitter Optimization
Balanced
Minimize Output Jitter
Maximize Input Jitter filtering
Dynamic Reconfig Interface Options
AXI4Lite
DRP
Phase Duty Cycle Config
Write DRP registers
Input Clock Information
Input Clock
Port Name
Input Frequency(MHz)
Jitter Options
Input Jitter
Source
Primary
clk_in1
MANUA
125.000
19.000 - 800.000
UI
0.010
Single ended clock
Secondary
clk_in2
AUTO
100.000
114.286 - 228.571
0.010
Single ended clock
OK
Cancel



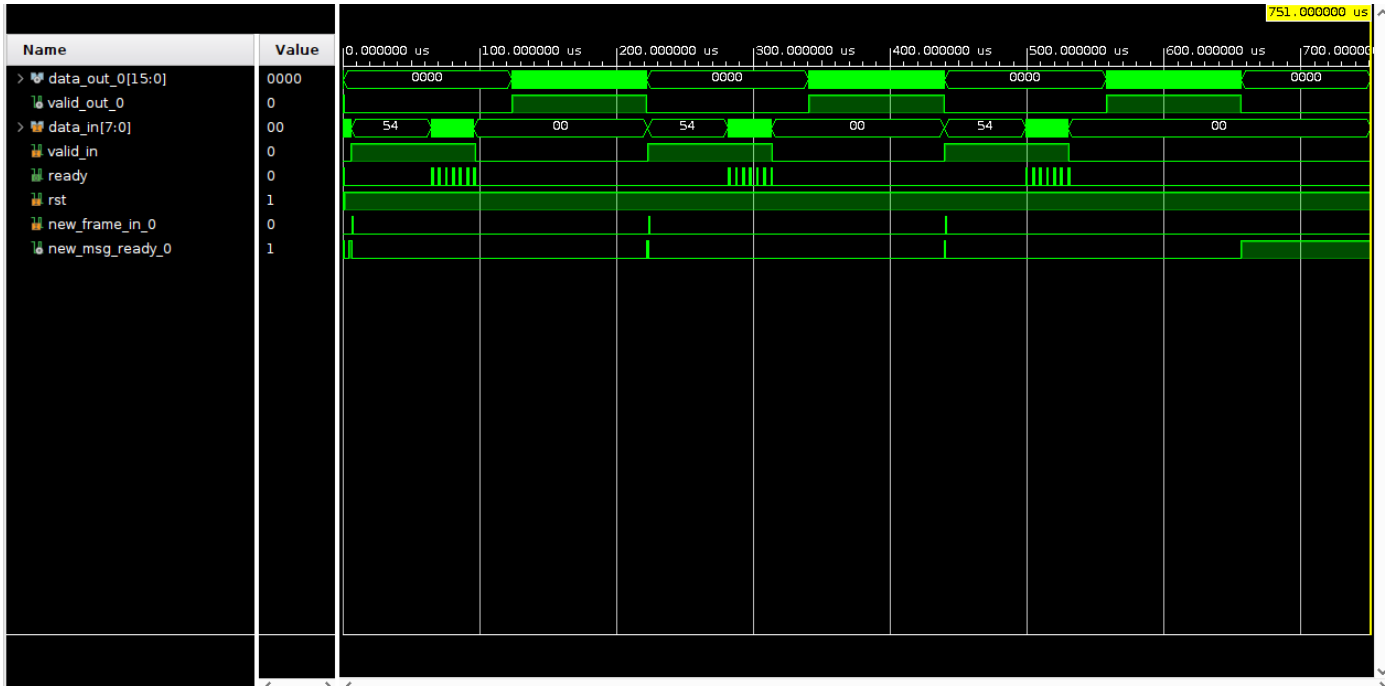
Resets separados para la FIFO y para el IP-Core.

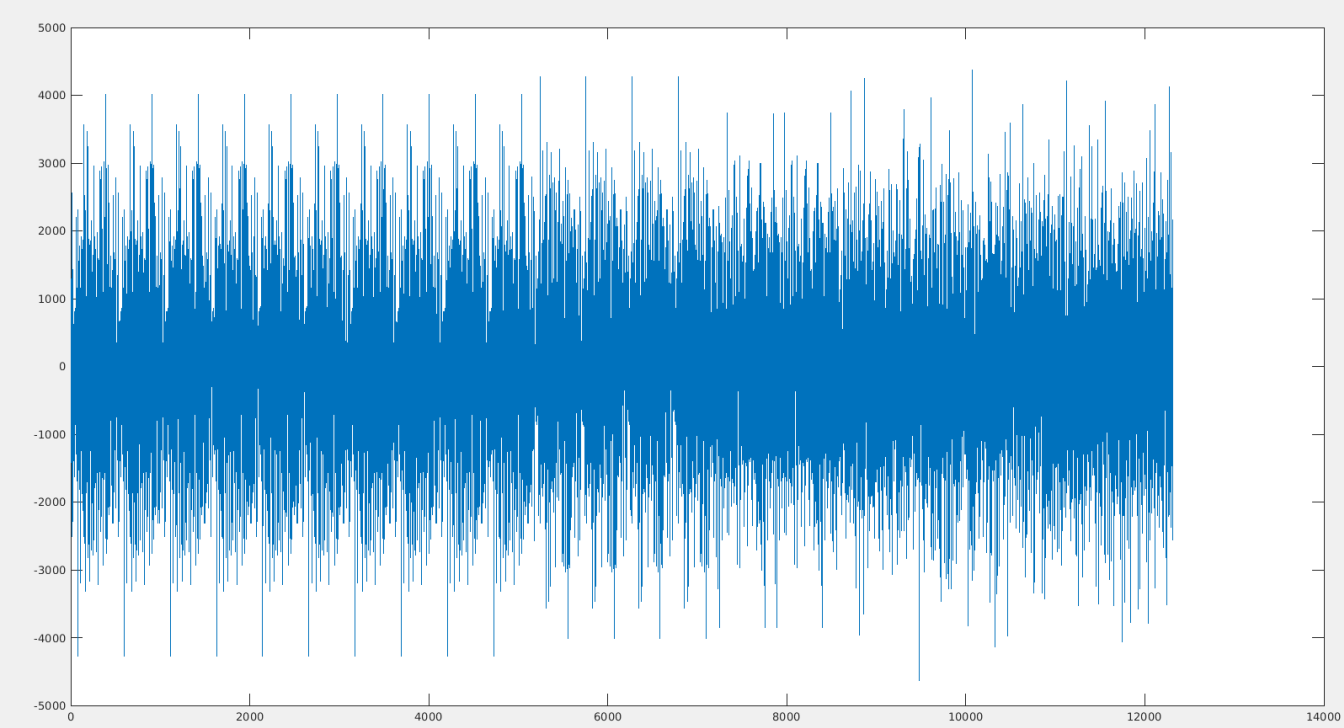
Simulación

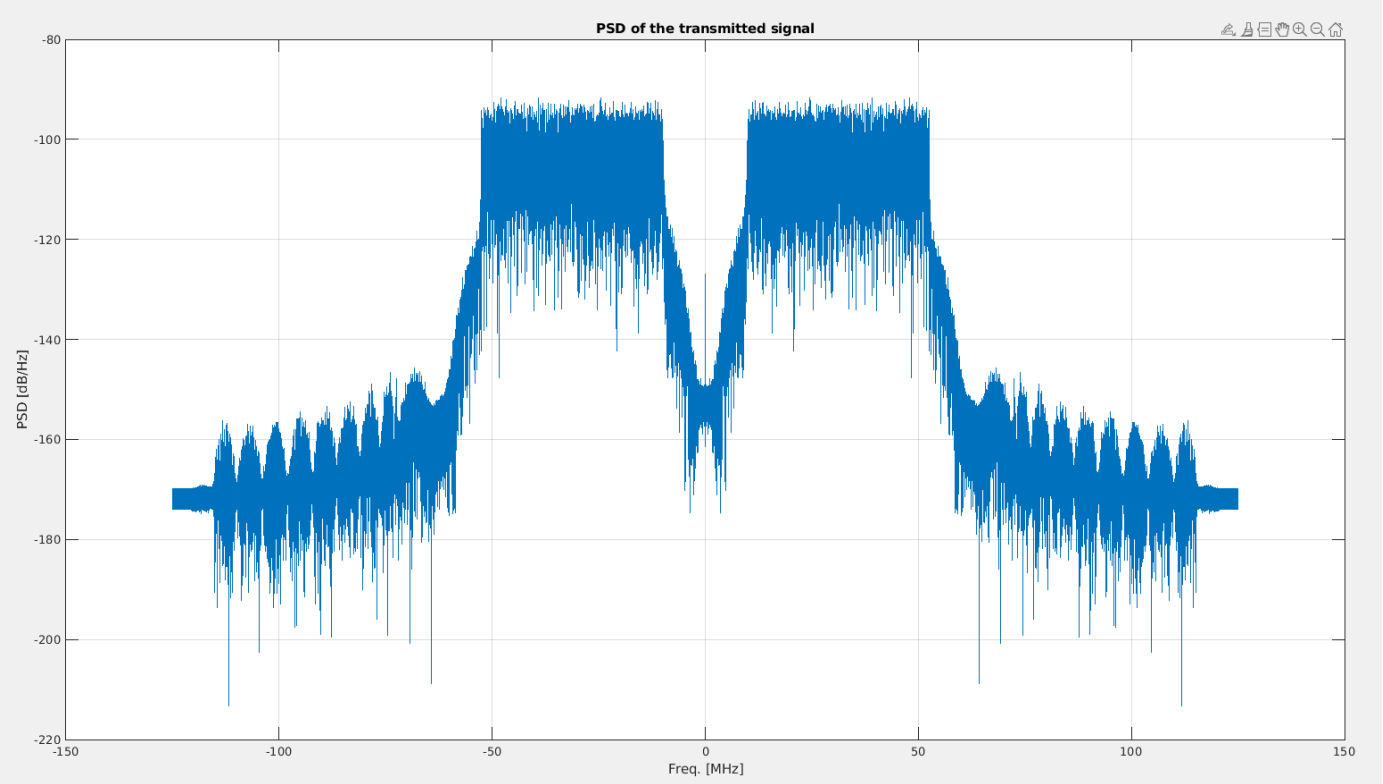
Critical warnings: 0.

La simulación fue realizada utilizando archivos adjuntos `data_in.mem` y `data_out.mem`, y con los siguientes valores de registros:

- msg = "This is an example message used to test the transmitter. It is made large on purpose to test for a large message being transmitted "
- reg0 = 147
- reg1 = 17
- reg2 = 65792
- reg3 = 66063





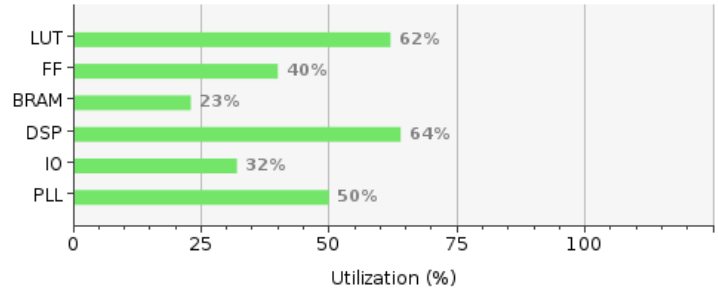


Sintesis

Importante: agregar el archivo de constraints al proyecto de Vivado, de nombre`ieee_constraints.xdc`.

Critical warnings: 0.

Resource	Utilization	Available	Utilization %
LUT	10870	17600	61.76
FF	14216	35200	40.39
BRAM	13.50	60	22.50
DSP	51	80	63.75
IO	32	100	32.00
PLL	1	2	50.00



Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	2.387 ns	Worst Hold Slack (WHS):	-0.100 ns	Worst Pulse Width Slack (WPWS):	2.000 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	-0.281 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	3	Number of Failing Endpoints:	0
Total Number of Endpoints:	43067	Total Number of Endpoints:	43025	Total Number of Endpoints:	15148

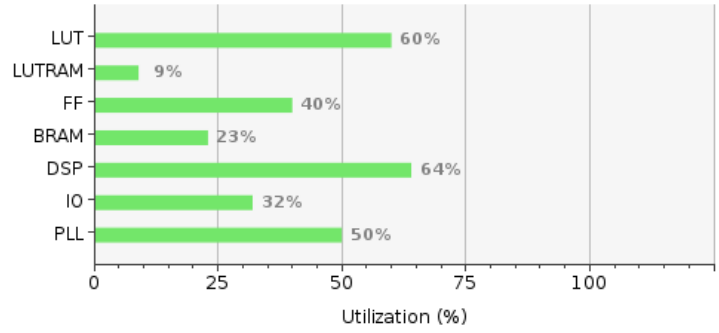
Timing constraints are not met.

Implementacion

Critical warnings: 2 (relacionadas con no haber definido la ubicación física de los pines).

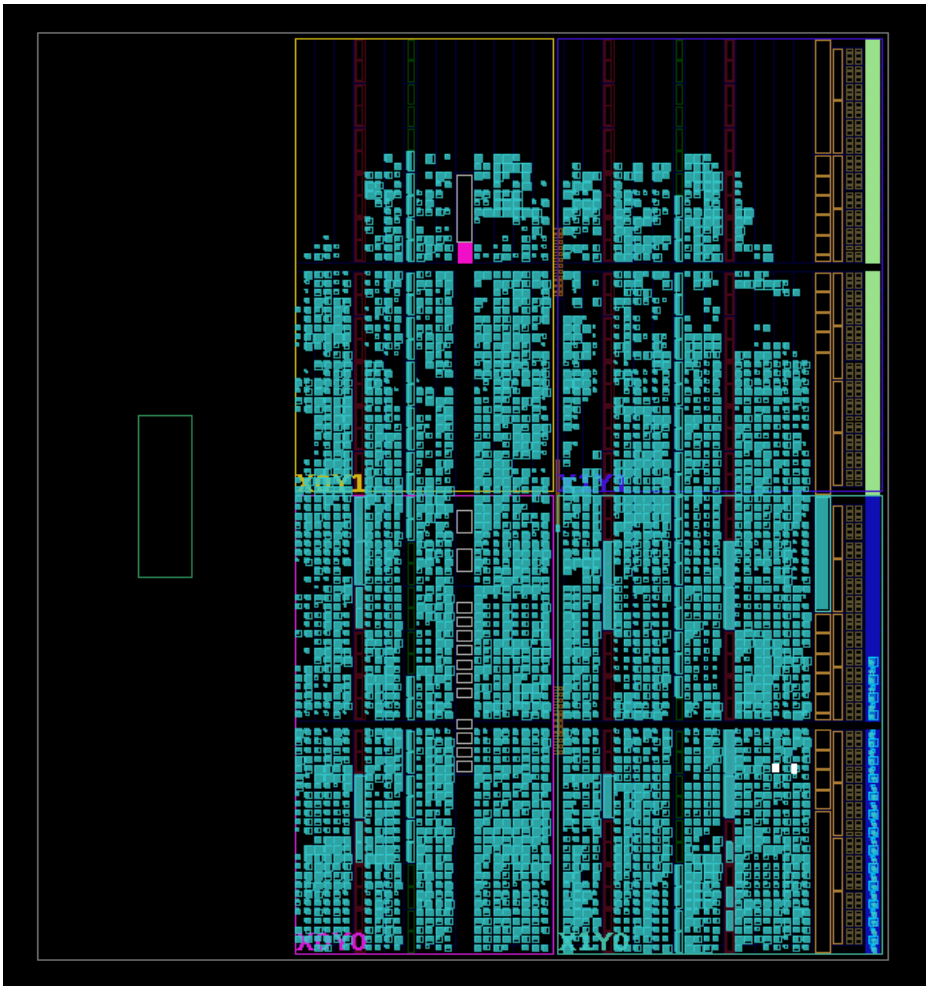
Warnings: 4. Una dice que n se uso el ZYNQ. Y 3 hacen referencia a DSPs inferidos, que se sugieren que tengan al menos dos registros a la salida (lo cual todos los multiplicadores cumplen).

Resource	Utilization	Available	Utilization %
LUT	10631	17600	60.40
LUTRAM	519	6000	8.65
FF	14098	35200	40.05
BRAM	13.50	60	22.50
DSP	51	80	63.75
IO	32	100	32.00
PLL	1	2	50.00



Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.546 ns	Worst Hold Slack (WHS):	0.027 ns	Worst Pulse Width Slack (WPWS):	2.000 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	42948	Total Number of Endpoints:	42928	Total Number of Endpoints:	15030

All user specified timing constraints are met.



Historial de versiones

v5.0

- Se divide a la mitad el valor de salida (anteriormente, había un logical shift izquierdo, que ahora se eliminó para evitar que el ringing de salida sature el DAC).
- Se agrega una FIFO a la entrada del payload, para evitar problemas de sincronización.

v4.0

- Ahora la simulación RTL y la simulación post implementación coinciden. Se modificó la lógica de bloques con "enable", y se hicieron los subsistemas sincrónicos.
- Se reduce el ancho de banda máximo de la señal y se hace el filtro de interpolación más fino.

v3.0

- Ahora solamente se lee el flanco ascendente de la señal `new_frame_in`.
- La salida es ahora un int16, que toma valores entre [-8192; 8191].
- Se agrega la nueva salida `new_msg_ready`, para sincronizar la recepción de nuevos símbolos desde el software.

v2.0

- Se elimina la señal "Tlast" a la entrada del Ip Core.
- La salida ahora es de 16 bits, con los dos bits LSB ignorados.

v1.2

Se agrega mención al archivo de constraints.

v1.1

Se agrega especificación del tamaño de "120 bytes" múltiplo del mensaje.

v1.0

Creación inicial del documento