

# 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 "valid" de la salida esté activa.

## 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.
- **[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 (testado con "1111").
- **bat[4:0]:** *batId*. Siempre "00010".
- **cp[2:0]:** *cyclicPrefixId*. Cualquier valor menos "000". (testado 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).

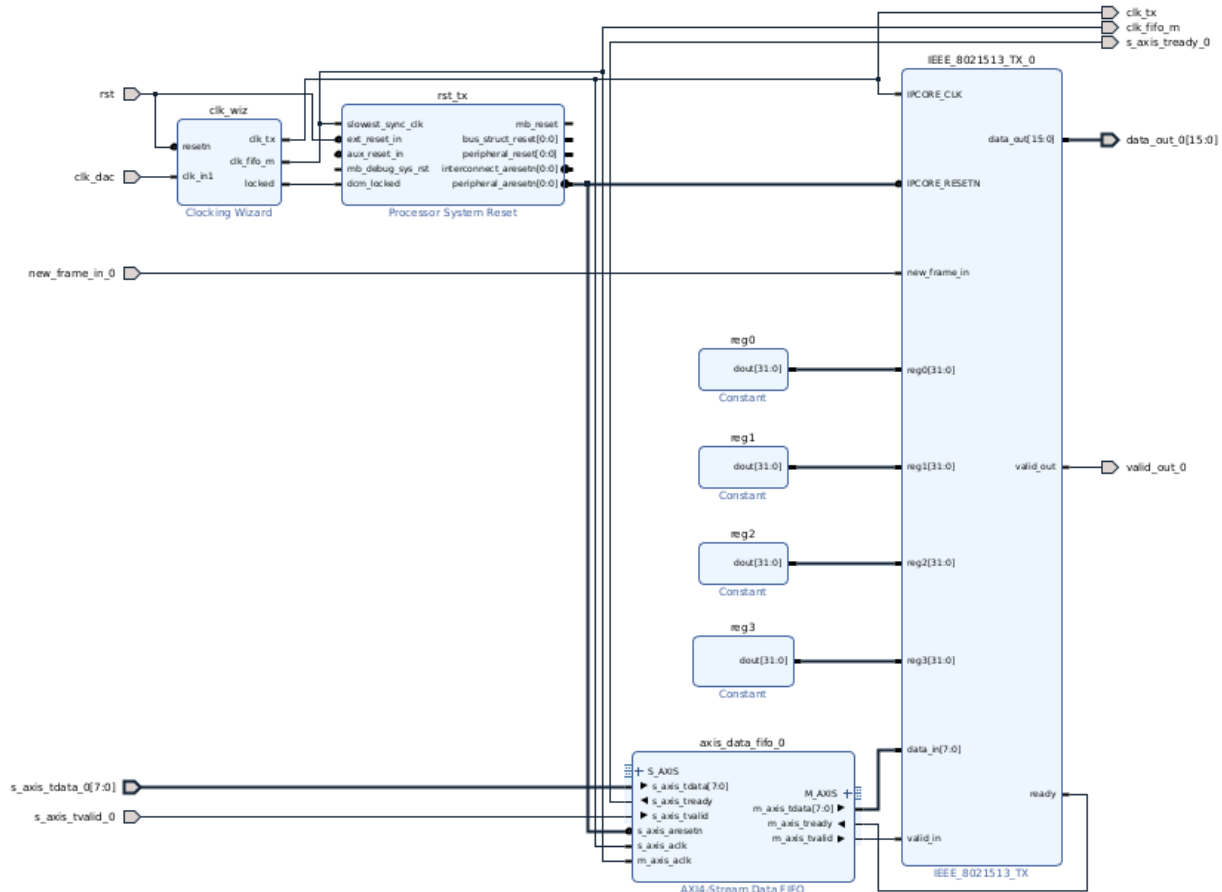
## Outputs

- **data\_out:** [fi(1,16,15)]. Valores de salida, para el DAC. Los dos bits LSB no se usan.
- **valid\_out:** [bool]. Indica que el valor de salida es válido.
- **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. 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.
4. Esperar mientras se procesan el preámbulo y encabezado.
5. Se va a levantar la señal de **ready** y va a empezar a leer la FIFO la cantidad de bytes indicada por los registros.
6. Esperar mientras se forma el símbolo OFDM.
7. Se envía a la salida una señal continua de 125MHz lista para conectarse al DAC. Se indica su validez con la señal **valid\_out**.
8. No se puede levantar otra señal de **new\_frame\_in** hasta el falling\_edge de la señal **valid\_out**.

## Block Design



### 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
<input checked="" type="checkbox"/>	Primary	clk_in1	<input type="text" value="125.000"/> <div>MANUAL</div>	19.000 - 800.000	UI	0.010	Single ended clock
<input type="checkbox"/>	Secondary	clk_in2	<input type="text" value="100.000"/> <div>AUTO</div>	114.286 - 228.571		0.010	Single ended clock

OK

Cancel

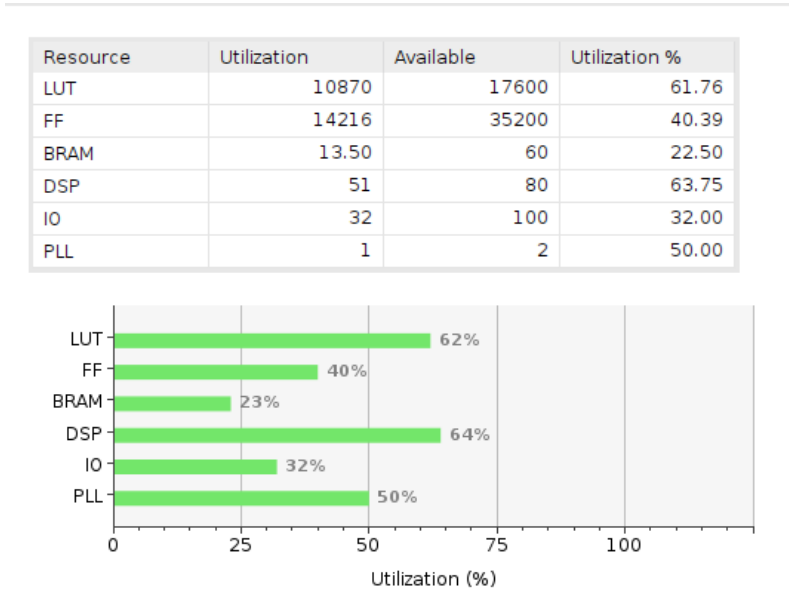


Resets separados para la FIFO y para el IP-Core.

Critical warnings: 0.



Critical warnings: 0.



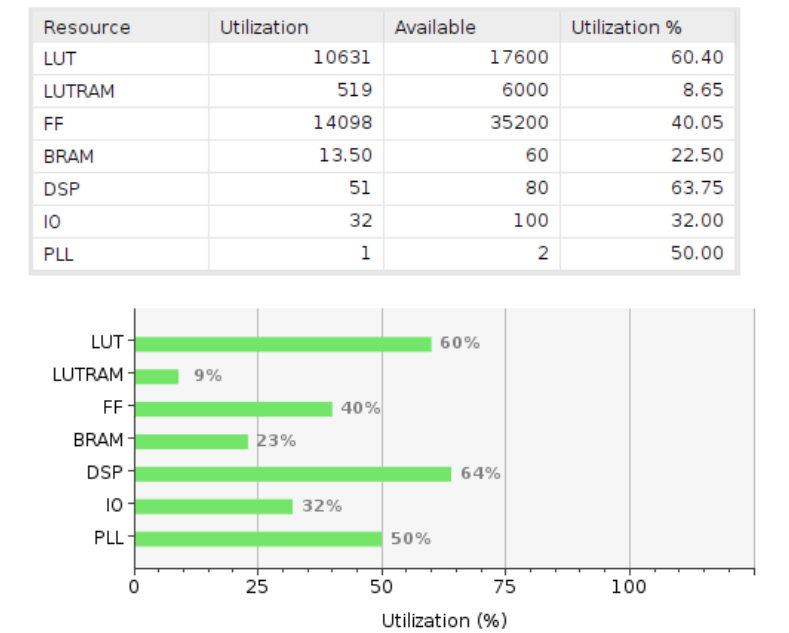
Design Timing Summary

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.		

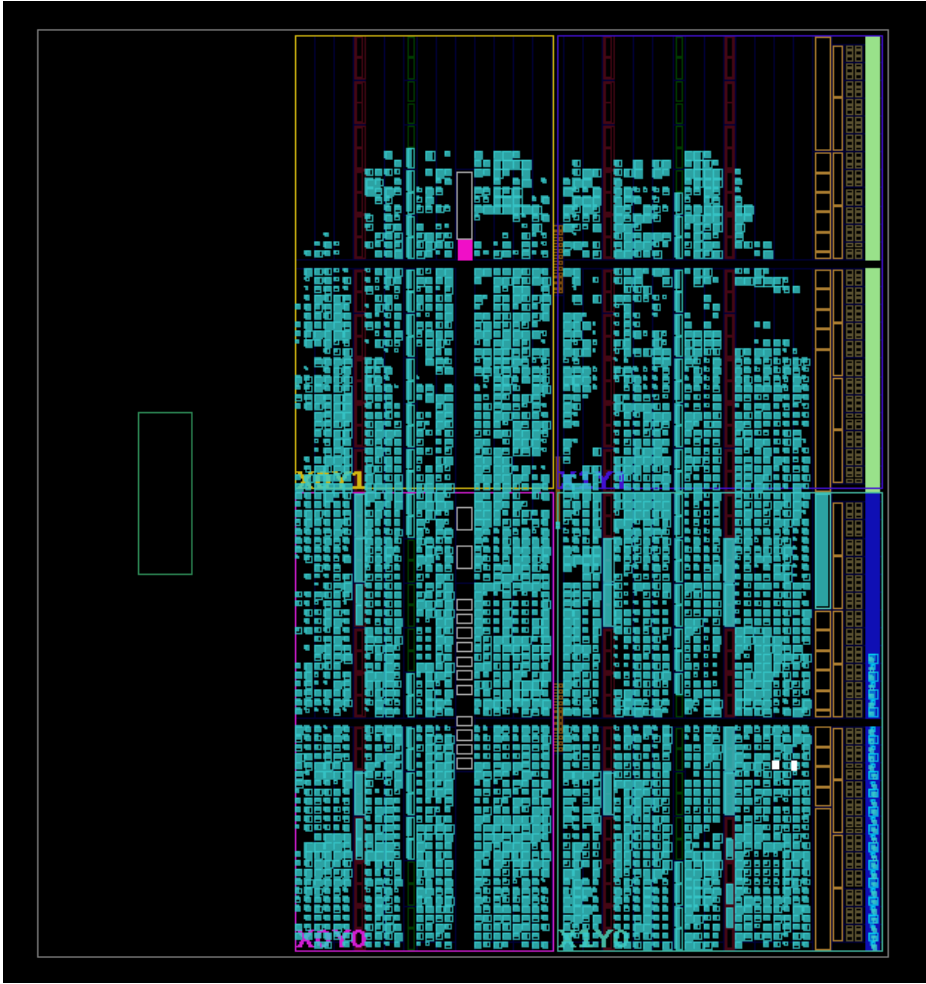
Implementacion

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).



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

### 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