

Transmisor

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

Funcionalidades probadas

- Máximo tamaño transmisible por trama: mínimo (TODO probar).

Clocks

- **clk_ext**: [100 MHz]. Clock del procesador ZYNQ (puede tomar otro valor).
- **clk_tx**: [125 MHz]. Clock del transmisor. Sincrónico con la salida del transmisor.
- **clk_fifo_s**: [100 MHz]. Clock para cargar datos a la FIFO (puede tomar otro valor).
- **clk_fifo_m**: [15.625 MHz]. Clock para sacar datos de la FIFO. 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).
- **last_frame**: Señal *TLAST* de AXI4 Stream. Indica que es el último elemento de este paquete de transmisión.

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	15	14
0x00	x	x	x	x	x	x	x	x	p23	p22	p21	p20	p19	p18	p17	p16	p15	p14
0x04	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	m15	m14
0x08	x	x	x	x	x	concat2	concat1	concat0	x	x	x	x	x	rep2	rep1	rep0	x	x
0x0c	x	x	mimon2	mimon1	mimon0	mimos2	mimos1	mimos0	x	x	x	x	x	cp2	cp1	cp0	x	x

- **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 = 120".

Por ejemplo: si su mensaje es de 100 bytes, entonces (psduSize = 100; messageDuration = 20).

Si su mensaje es de 300 bytes, entonces (psduSize = 300; messageDuration = 60).

Si bien el mensaje "real" tiene un tamaño fijo, el mensaje escrito en la FIFO de entrada debe ser un múltiplo de 120 bytes.

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

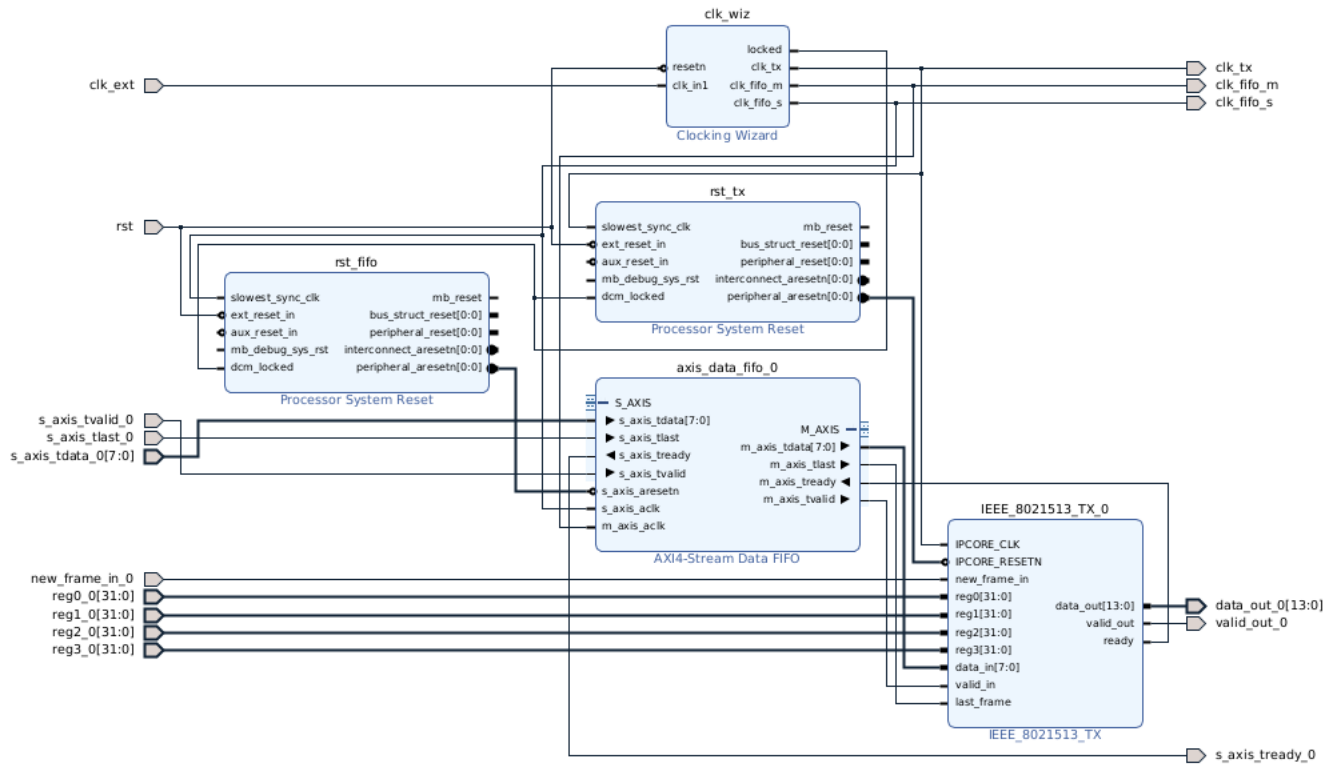
Outputs

- **data_out**: [fi(1,14,13)]. Valores de salida, para el DAC.
- **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 120 bytes (completar con '0' de ser necesario).
2. Setear los registros reg0, reg1, reg2 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 preambulo 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)

DocumentationIP Location

IP SymbolResource

Show disabled ports

resetn

clk_in1

clk_tx

clk_fifo_m

clk_fifo_s

locked

Component Nameclk_wiz

Clocking OptionsOutput ClocksMMCM SettingsSummary

Clock Monitor

Enable Clock Monitoring

Primitive

MMCM

PLL

Clocking FeaturesJitter Optimization

Frequency SynthesisPhase AlignmentDynamic ReconfigSafe Clock Startup

Minimize PowerSpread SpectrumDynamic Phase Shift

BalancedMinimize Output JitterMaximize Input Jitter filtering

Dynamic Reconfig Interface Options

AXI4LiteDRP

Phase Duty Cycle ConfigWrite DRP registers

Input Clock Information

	Input Clock	Port Name	Input Frequency(MHz)		Jitter Options	Input Jitter	Source
<input checked="" type="checkbox"/>	Primary	clk_in1	<div>AUTO100.000</div>	10.000 - 800.000	UI	0.010	Single er
<input type="checkbox"/>	Secondary	clk_in2	<div>AUTO100.000</div>	60.000 - 120.000		0.010	Single er

Clocking OptionsOutput ClocksMMCM SettingsSummary

Output Clock	Port Name	Output Freq (MHz)	Phase (degrees)	Duty Cycle (%)
		RequestedActual	RequestedActual	RequestedActual
<input checked="" type="checkbox"/> clk_out1	clk_tx	125.000125.00000	0.0000.000	50.00050.000
<input checked="" type="checkbox"/> clk_out2	clk_fifo_m	15.62515.62500	0.0000.000	50.00050.000
<input checked="" type="checkbox"/> clk_out3	clk_fifo_s	100.000100.00000	0.0000.000	50.00050.000
<input type="checkbox"/> clk_out4	clk_out4	100.000N/A	0.000N/A	50.000N/A
<input type="checkbox"/> clk_out5	clk_out5	100.000N/A	0.000N/A	50.000N/A
<input type="checkbox"/> clk_out6	clk_out6	100.000N/A	0.000N/A	50.000N/A
<input type="checkbox"/> clk_out7	clk_out7	100.000N/A	0.000N/A	50.000N/A

☐ USE CLOCK SEQUENCING

Output Clock	Sequence Number
clk_out1	1
clk_out2	1
clk_out3	1
clk_out4	1
clk_out5	1
clk_out6	1
clk_out7	1

Enable Optional Inputs / Outputs for MMCM/PLL

resetpower_downinput_clk_stoppedlocked

clkfbstopped

Clocking Feedback

SourceSignaling

Automatic Control On-ChipAutomatic Control Off-ChipUser-Controlled On-ChipUser-Controlled Off-Chip

Single-endedDifferential

Reset Type

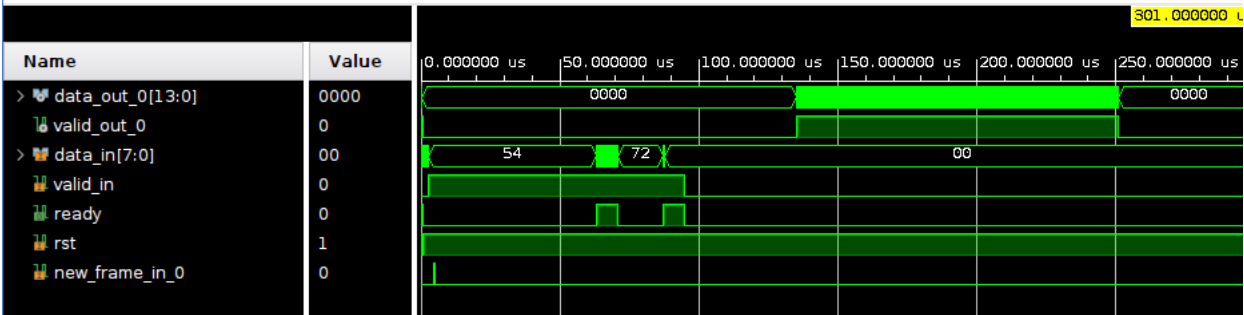
Active HighActive Low

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

3 / 4

Simulación

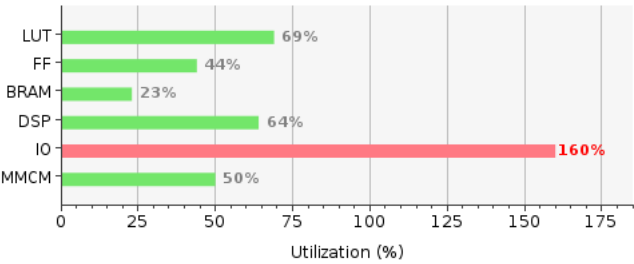
Critical warnings: 0.



Síntesis

Critical warnings: 0.

Resource	Utilization	Available	Utilization %
LUT	12149	17600	69.03
FF	15515	35200	44.08
BRAM	13.50	60	22.50
DSP	51	80	63.75
IO	160	100	160.00
MMCM	1	2	50.00



Para el timing, no cumple el tiempo de hold de la FIFO. Esto debería revisarse una vez que se haga la implementación. En el peor de los casos, la frecuencia de clock del esclavo de la FIFO debería de disminuirse.

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 2.387 ns	Worst Hold Slack (WHS): -0.085 ns	Worst Pulse Width Slack (WPWS): 2.750 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): -0.085 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 1	Number of Failing Endpoints: 0
Total Number of Endpoints: 47871	Total Number of Endpoints: 47829	Total Number of Endpoints: 16845

Timing constraints are not met.

Implementación

Error: ""

Historial de versiones

- v1.1
 - Se agrega especificación del tamaño de "120 bytes" múltiplo del mensaje.
- v1.0
 - Creación inicial del documento