

Receiver

Documento que indica todo lo necesario para usar el receptor, compuesto de tres Ip Cores distintos:

- **rx_rf_ip (v3.0)**: Encargado de comunicarse con el ADC y decimar las muestras.
- **rx_demod_ip (v3.0)**: Encargado de realizar la demodulación OFDM y QAM.
- **rx_decoder (v3.0)**: Encargado de realizar la decodificación LDPC y lógicas digitales.

Los IP Cores se encuentran en esta carpeta: [IP Core Rx](#).

El proyecto ejemplo de Vivado donde se corrió la simulación: [Ejemplo Rx](#).

Funcionalidades probadas

- Máximo tamaño de trama recibida: 4096 bytes.
- Mínimo tamaño de trama recibida: 64 bytes.
- Acepta la recepción de múltiples tramas consecutivas, sin necesidad de reset.
- Señal de entrada mínima: [-0.1; 0.1]. Señal de entrada máxima: [-1.0; 1.0].
- Máximo error de frecuencia entre el clock del transmisor y el receptor: 15 kHz.
- Delay aleatorio entre señal recibida y transmitida.

Clocks

- **clk_adc**: [125 MHz]. Clock físico del ADC, conectado a la entrada del clocking wizard.
- **clk_125**: [125 MHz]. Clock usado para alimentar **rx_rf_ip** y **rx_decoder**.
- **clk_625**: [62.5 MHz]. Clock usado para alimentar **rx_demod_ip**.

Inputs

- **IPCORE_CLK**: [clk]. Señal de clock de 125 MHz.
- **IPCORE_RESETN**: [bool]. Señal de reset ACTIVE LOW ('0' para resetear).
- **data_in**: [int16]. Datos de entrada del ADC. Los dos bits MSB son descartados, por lo que acepta valores entre [-8192; 8191].
- **valid_in**: [bool]. Cuando vale "0", la salida de la etapa de RF es "0".

Outputs

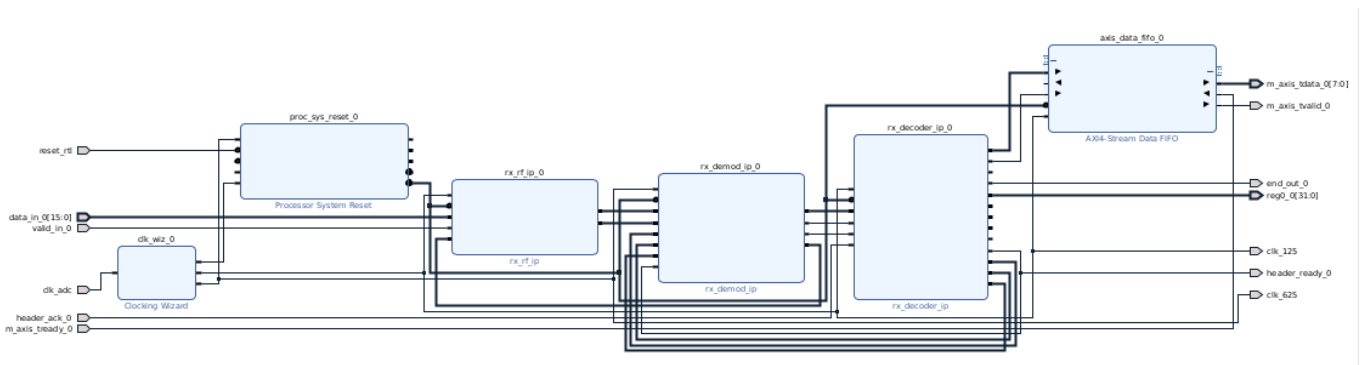
- **data_out**: [uint8_t]. Palabras recibidas del payload. Solamente se reciben las palabras "efectivas". Recuerde que para el transmisor, se transmitían palabras extras para llenar el múltiplo de 21. Esas palabras extras son descartadas automáticamente por el receptor.

- **valid_out**: [bool]. Vale "1" mientras las palabras del payload sean válidas. Señal de AXI4-Stream.
- **start_out**: [bool]. Vale "1" mientras sea válida la primera palabra del payload.
- **end_out**: [bool]. Vale "1" mientras sea válida la última palabra del payload, indicando el final de la recepción.
- **[reg0, reg1, reg2, reg3]**: [uint32_t]. Registros leídos del encabezado. Su valor es válido luego de haber recibido la señal de **header_ready**.
- **header_ready_fifo**: [bool]. Pulso que vale "1" una vez que fueron leídos los registros del encabezado, indicando que se detectó una nueva transmisión.
- **header_error**: [bool]. Se pone en "1" y queda levantado. Indica que falló el CRC del encabezado. Se mantiene prendido hasta que se resetee el Ip Core. No se envía **header_ready** ni se envían datos de payload si se levantó este flag.

Modo de funcionamiento

1. Se reciben continuamente datos del DAC. Solamente se detectará un símbolo OFDM mientras la señal **valid_in** de la etapa de RF esté en "1".
2. Luego de que se haya detectado el preámbulo OFDM de un mensaje, el receptor seguirá trabajando para demodular el encabezado. Una vez que el encabezado haya sido leído, se pondrá en "1" la señal **header_ready**, y se actualizarán los valores de los cuatro registros. En caso de que haya habido un error en la decodificación, se levantará la señal **header_error**:
 - En caso de que la señal **header_error** se haya levantado, reiniciar todo el IP Core, e ignorar el símbolo OFDM actual.
 - Poner en "1" la señal de entrada **header_ack** una vez leídos los registros, para poner la señal **header_ready** en "0".
3. Pasado un tiempo, se levantará la señal **start_out**, indicando que están llegando los primeros bytes del payload a la FIFO, junto con la señal **valid_out**. Los datos del payload se escriben de manera continua.
4. Una vez que se termina de recibir el payload, se levanta la señal **end_out** durante la última palabra. Esta señal es equivalente a un TLAST de AXI4-Stream, si se desea usar.

Block design

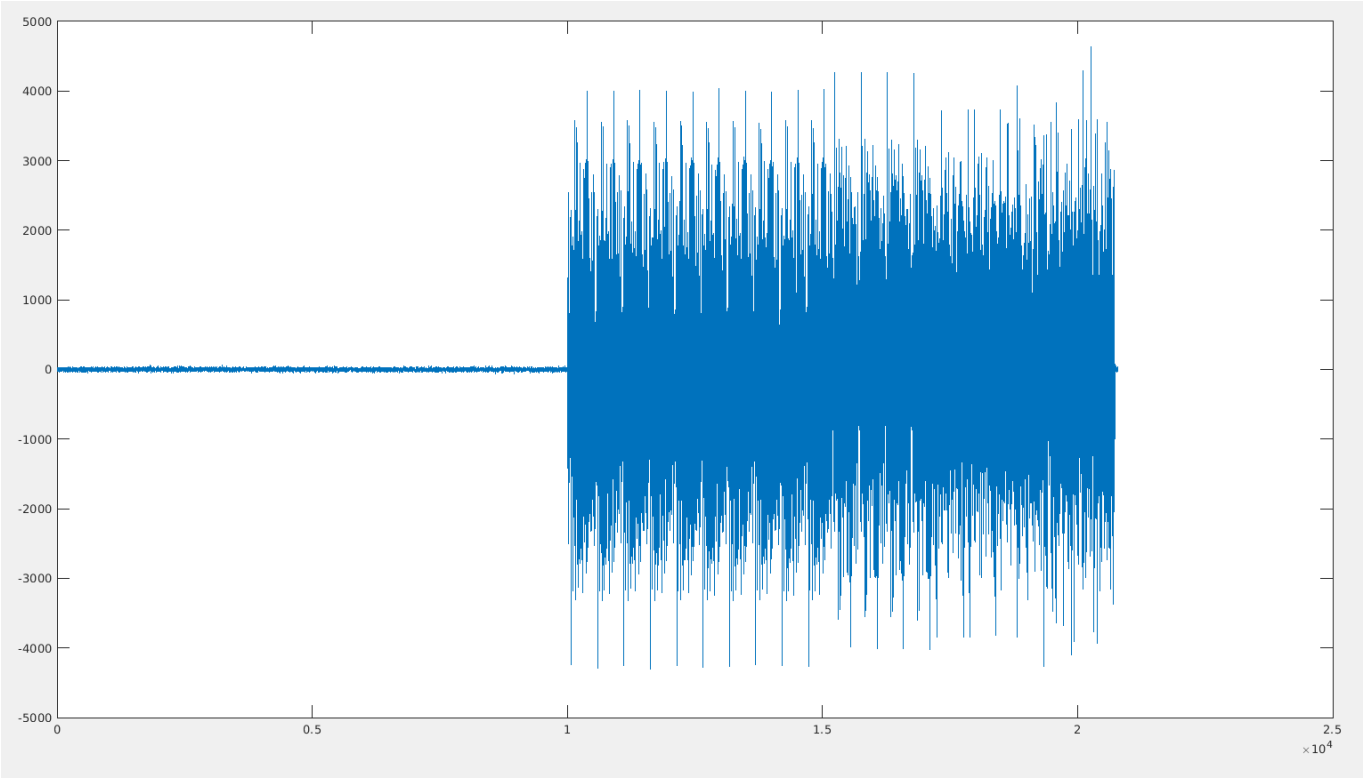
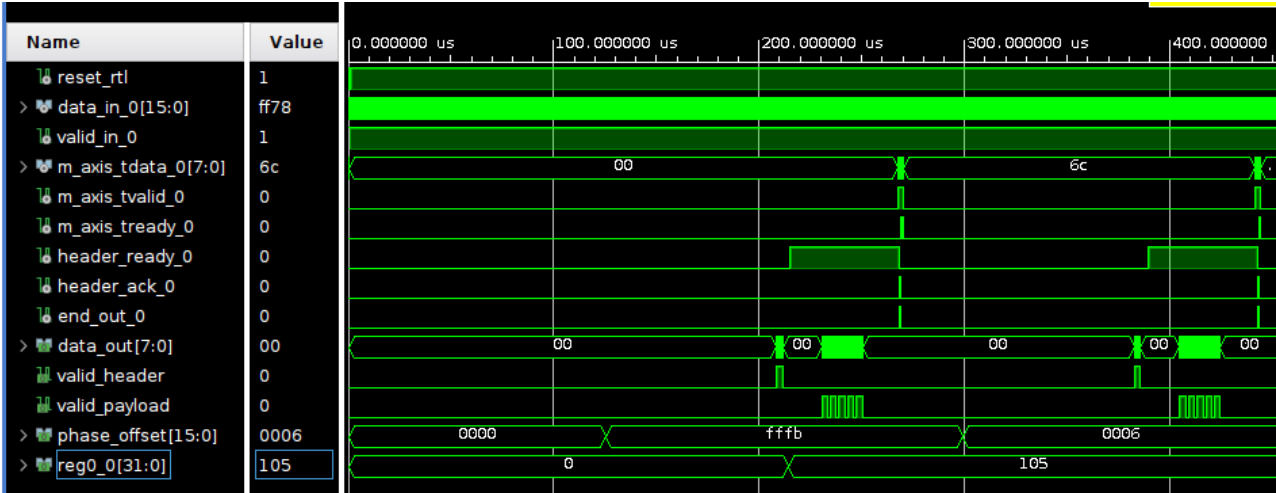


Simulation

Critical warnings: 0.

Valores de registros usados:

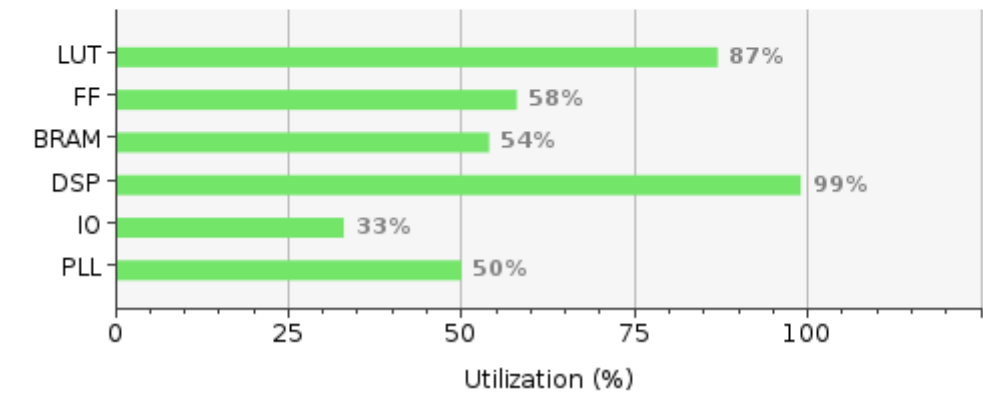
- Mensaje recibido = 'This is a test of the RX for the UTN VLC Project! Some extra text need to be added for it to work well'
- reg0 = 105
- reg1 = 3
- reg2 = 65792
- reg3 = 66063



Sintesis

Critical warnings: 0.

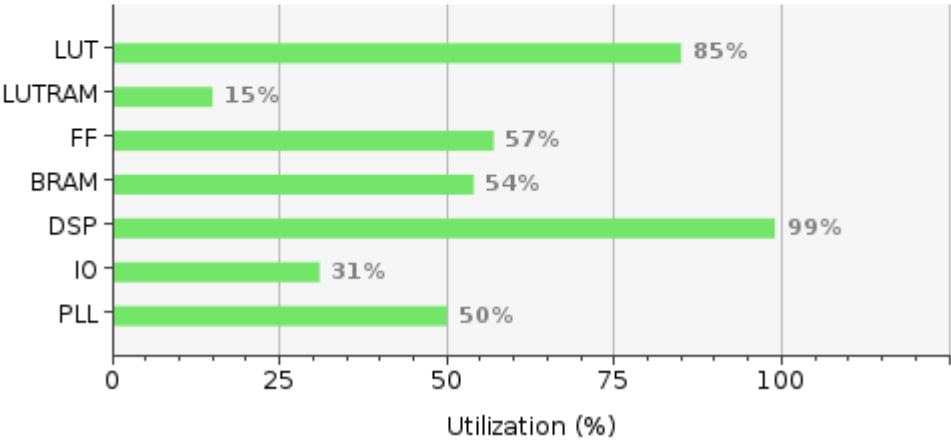
Resource	Utilization	Available	Utilization %
LUT	15303	17600	86.95
FF	20363	35200	57.85
BRAM	32.50	60	54.17
DSP	79	80	98.75
IO	33	100	33.00
PLL	1	2	50.00



Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 2.081 ns	Worst Hold Slack (WHS): -0.151 ns	Worst Pulse Width Slack (WPWS): 2.000 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): -1.406 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 10	Number of Failing Endpoints: 0
Total Number of Endpoints: 64646	Total Number of Endpoints: 64604	Total Number of Endpoints: 20982
Timing constraints are not met.		

Implementation

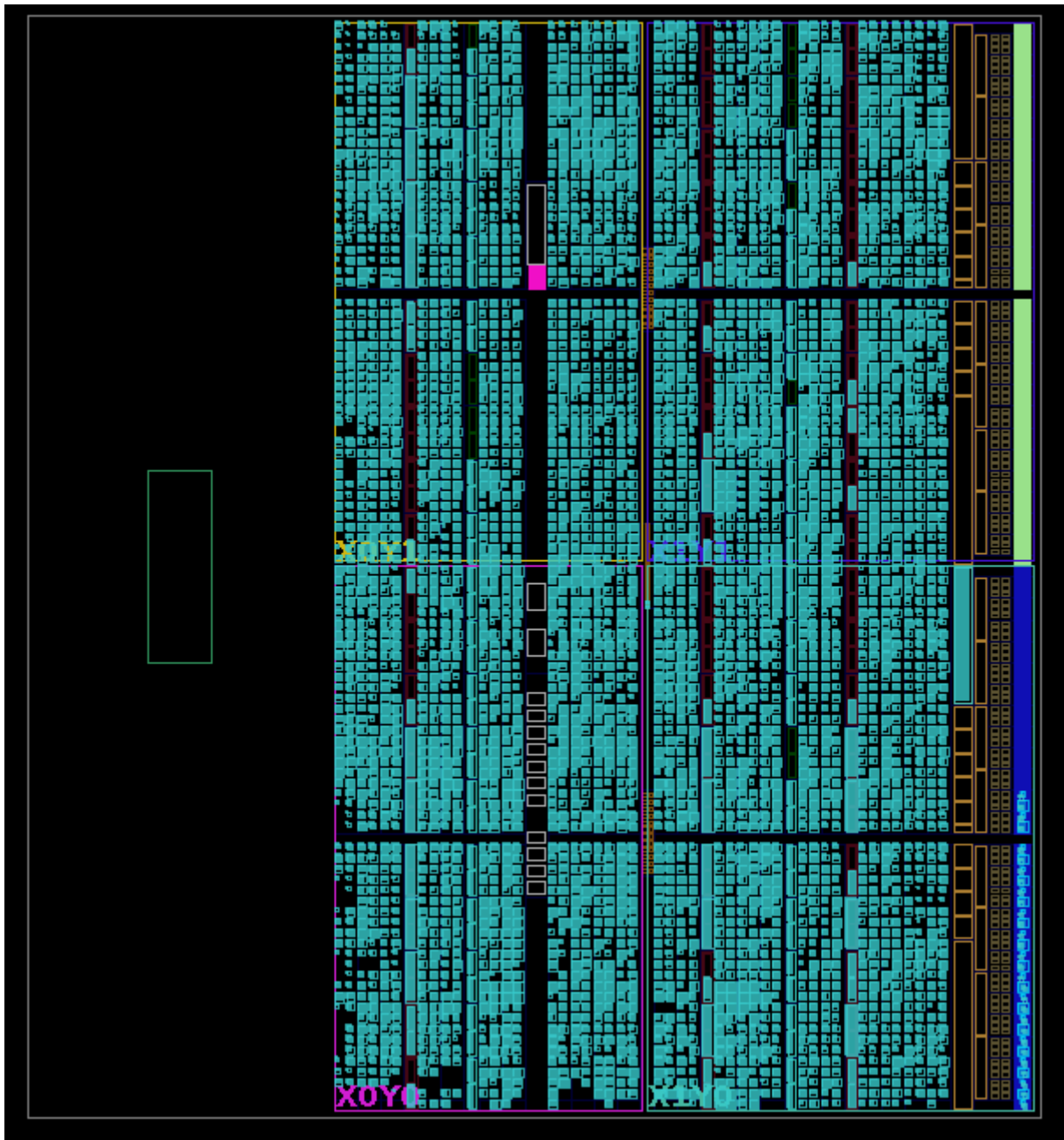
Resource	Utilization	Available	Utilization %
LUT	14990	17600	85.17
LUTRAM	906	6000	15.10
FF	20232	35200	57.48
BRAM	32.50	60	54.17
DSP	79	80	98.75
IO	31	100	31.00
PLL	1	2	50.00



Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 0.502 ns	Worst Hold Slack (WHS): 0.022 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: 67458	Total Number of Endpoints: 67438	Total Number of Endpoints: 21749

All user specified timing constraints are met.



Versionado

v3.1

- LDPC decoder con 8 iteraciones.
- Se cambia la lógica de header ready y header error. En caso de error, header ready no se levanta. Ahora, header ready es un pulso, se espera que se almacenen los registros en una fifo usando este pulso como valid, así se puede almacenar muchos frames a la vez, sin necesidad de leerlos todos juntos.

v3.0

- Se separa el receptor en tres IP Cores distintos.
- Se elimina el archivo de constraints.

v2.0

- Se agrega la entrada `header_ack`, y se mantiene en nivel la salida `header_ready`.
- Se modifica el rango de entrada a un int16, entre los valores [-8192; 8191].
- Se modifica el filtro pasabajos del decimador, para hacerlo más estricto en frecuencia (fstop de 37MHz a fstop de 33.5 MHz).
- Se agregan delays en el demodulador de OFDM, y se modifican las constraints de MCP para que incluyan sólo el demodulador OFDM.

v1.0

- Emisión inicial del documento.