

**Transceivers**

**Xilinx**

Por [ing\\_davidrubio@outlook.com](mailto:ing_davidrubio@outlook.com) | [Blog](#)

# Introducción

Este documento tiene como objetivo explicar los conceptos básicos sobre el uso de transceivers de Xilinx.

## Puertos básicos

Nombre	Dirección	Tamaño	Función
<i>gtwiz_reset_clk_freerun_in</i>	In	1	Entrada de reloj del sistema, nunca superior a 250MHz
<i>gtwiz_userclk_tx_active_in</i>	In	1	Entrada de activación de Tx. Esta entrada tiene que estar sincronizada. Activa a nivel alto.
<i>gtwiz_userclk_rx_active_in</i>	In	1	Entrada de activación de Rx. Esta entrada tiene que estar sincronizada. Activa a nivel alto.
<i>gtwiz_reset_all_in</i>	In	1	Reset global del Transceiver
<i>gtwiz_userdata_tx_in</i>	In	32/40 x N.º transceivers	Salida de datos por el Transceiver
<i>gtwiz_userdata_rx_out</i>	Out	32/40 x N.º transceivers	Entrada de datos por el transceiver
<i>gtrefclk00_in</i>	In	1	Reloj de referencia, este reloj no tiene mucho valor. Viene de la salida de un IBUFGDS_GTE3 con entrada de los relojes del MGT.
<i>gthrxn_in</i>	In	1 x N.º transceivers	Entradas que vienen del exterior. No se define el pin en el XDC de este puerto. (es posible que está sea la entrada exterior del transceiver)
<i>gthrxp_in</i>	In	1 x N.º transceivers	
<i>gthtxn_out</i>	Out	1 x N.º transceivers	Salidas que vienen del exterior. No se define el pin en el XDC de este puerto. (es posible que está sea la salida exterior del transceiver)
<i>gthtxp_out</i>	Out	1 x N.º transceivers	

<b><i>rxoutclk_out</i></b>	Out	1 x N.º transceivers	Reloj de salida de Rx, va a un BUFG_GT
<b><i>txoutclk_out</i></b>	Out	1 x N.º transceivers	Reloj de salida de Tx, va a un BUFG_GT
<b><i>rxusrclk_in</i></b>	In	1 x N.º transceivers	Reloj de Rx. Este reloj proviene de un BUFG_GT del puerto de salida <i>rxoutclk_out</i> . El reloj útil para el trabajo en FPGA es el <i>rxusrclk2_in</i>
<b><i>rxusrclk2_in</i></b>	In	1 x N.º transceivers	
<b><i>txusrclk_in</i></b>	In	1 x N.º transceivers	Reloj de Tx. Este reloj proviene de un BUFG_GT del puerto de salida <i>txoutclk_out</i> . El reloj útil para el trabajo en FPGA es el <i>txusrclk2_in</i>
<b><i>txusrclk2_in</i></b>	In	1 x N.º transceivers	
<b>&lt;señales de entrada&gt;</b>	In	<>	Señales activas a '1', se pueden meter a un VIO, con el reloj de <i>txusrclk2_in</i> o <i>rxusrclk2_in</i>
<b>&lt;señales de salida&gt;</b>	Out	<>	Señales activas a '1', se pueden meter a un ILA o a un VIO, con el reloj de <i>txusrclk2_in</i> o <i>rxusrclk2_in</i>
<b><i>rx8b10ben_in*</i></b>	In	1 x N.º transceivers	Señal de habilitación de la codificación 8b/10b para RX, activa a nivel alto
<b><i>tx8b10ben_in*</i></b>	In	1 x N.º transceivers	Señal de habilitación de la codificación 8b/10b para TX, activa a nivel alto
<b><i>rxcommadeten_in**</i></b>	In	1 x N.º transceivers	Señales de habilitación de la alineación de los datos recibidos. Por norma general siempre a '1'.
<b><i>rxmcommaalignen_in**</i></b>	In	1 x N.º transceivers	
<b><i>rxpcommaalignen_in**</i></b>	In	1 x N.º transceivers	
<b><i>rxbyteisaligned_out**</i></b>	Out	1 x N.º transceivers	Señales de comprobación de la alineación de los datos. Se pueden dejar en abierto.
<b><i>rxbyterealign_out**</i></b>	Out	1 x N.º transceivers	
<b><i>rxcommadet_out**</i>, (2) ***</b>	Out	1 x N.º transceivers	

<b><i>txctrl0_in*</i></b>	In	16 x N.º Transceiver	Señales para alineación.
<b><i>txctrl1_in*</i></b>	In	16 x N.º Transceiver	No son necesarias, se ponen a '0'.
<b><i>txctrl2_in*</i></b>	In	8 x N.º Transceiver	Señal de transmisión de coma al receptor. Se utiliza el reloj de transmisión para enviarlo. Solo es necesario transmitir en los 4 primeros bits (el resto se obvian), solo tiene que transmitir durante la cabecera un "0001".
<b><i>rxctrl0_out*</i></b>	Out	16 x N.º Transceiver	Señales de recepción de la coma. No son necesarias, se dejan abiertas ( <i>open</i> )
<b><i>rxctrl1_out*</i></b>	Out	16 x N.º Transceiver	
<b><i>rxctrl2_out*</i></b>	Out	8 x N.º Transceiver	Señal de recepción de la coma. Esta señal es fundamental para conocer el desfase de los datos. El transmisor emite un "0001", y dependiendo de cómo se recibe se conoce el desalineamiento. <i>Ejemplo: si se recibe un "0100", quiere decir que tienes los datos partidos en dos.</i>
<b><i>rxctrl3_out*</i></b>	Out	8 x N.º Transceiver	Señal de recepción de los datos de control. No es necesaria, se puede dejar al aire.

\* Estas señales aparecen al activar la codificación 8b/10b del transceiver.

\*\* Estas señales aparecen al activar la detección de coma del transceiver.

\*\*\* Esta señal es activa a nivel alto durante dos ciclos de reloj, el primero es el ciclo de reloj anterior a la coma y el segundo es el de la coma, por lo que se puede utilizar para manejar la recepción de datos si fuera necesario.

*NOTA: no se puede activar una señal de control sin activar previamente el 8b/10b*

# Relojes

- **Reloj para Freerun**

Con un reloj diferencial se utiliza un IBUFGDS que convierte la señal en una señal de una línea, después, se pasa por un PLL para rebajarla a 100MHz, y por último, se utiliza un BUFG para conseguir la señal CLK\_FREERUN.



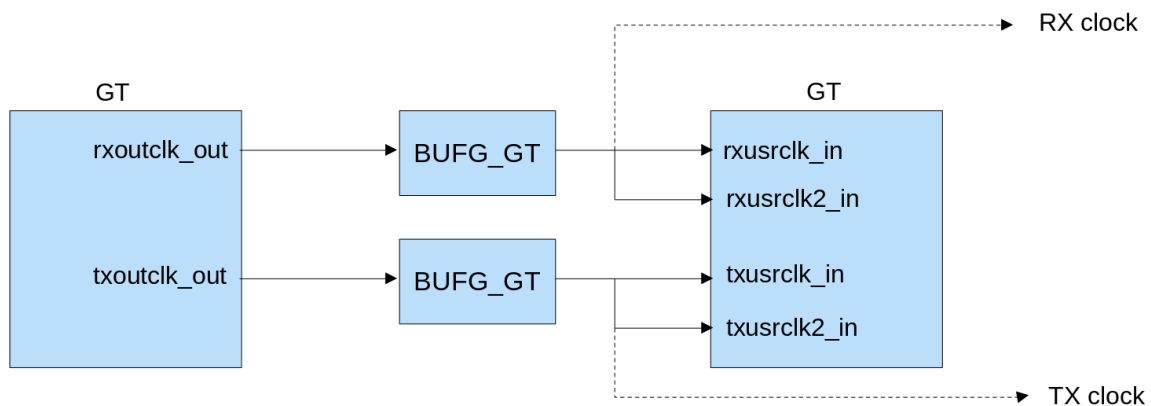
- **Reloj para referencia**

Para conseguir el reloj de referencia *gtrefclk00\_in* se tiene que coger las señales de reloj diferenciales del transceiver a 125MHz y se pasan por un IBUFGDS\_GTE3 (IBUFGDS\_GTE4 para los MPSoCs).



- **Relojes de trabajo**

Los relojes de trabajo y fundamentales del transceiver provienen de los puertos de salida del propio transceiver, estos puertos son el ***rxoutclk\_out*** y el ***txoutclk\_out***. Estos relojes se pasan cada uno por un **BUFG\_GT**, dónde se pueden dividir sus frecuencias en caso de ser necesario. Estos relojes son también entradas del propio transceiver por los puertos ***rxusrclk\_in***, ***rxusrclk2\_in***, ***txusrclk\_in*** y ***txusrclk2\_in***. Estos relojes tienen que estar sincronizados entre '1' y '2' (*aunque no tengan la misma frecuencia*). El reloj útil para la FPGA es el '2'.



# Alineamiento

Para el sistema de alineamiento dependen de varios parámetros.

El primero y fundamental para empezar es la señal **rxctrl2\_out**, esta señal de 8 bits (*de los que los 4 primeros NO se utilizan*) nos indica cuál es el desalineamiento de los datos de entrada. Hay que tener en cuenta que el transmisor solo emite en el envío de la cabecera el dato "0001". Bien, pues al recibir este dato se comprueba cuál es el desalineamiento.

- Si el dato es "0001"(1) quiere decir que no existe desalineación del dato recibido.  
..., <dato\_ant(31:0)> , <dato\_act(31:0)> , <dato\_sig(31:0)> , ...
- Si el dato es "0010"(2) quiere decir que los tres primeros bytes pertenecen al dato, y el último byte pertenece al dato anterior, siendo este el primer byte del dato anterior.  
..., <dato\_act(23:0), dato\_ant(31:24)>, <dato\_sig(23:0), dato\_act(31:24)>, ...
- Si el dato es "0100"(4) quiere decir que el dato recibido es la mitad del dato real, la otra mitad del dato viene con el dato siguiente tal que así:  
..., <dato\_act(15:0), dato\_ant(31:16)>, <dato\_sig(15:0), dato\_act(31:16)>, ...
- Si el dato es "1000"(8) quiere decir  
..., <dato\_act(7:0), dato\_ant(31:8)>, <dato\_sig(7:0), dato\_act(31:8)>, ...

Es importante tener en cuenta la alineación de los datos recibidos, pero también que los datos se pueden desalinear sobretodo al principio, o al recibir una coma entre medio de la comunicación. Por ello es necesario valorar la opción de añadir un **checksum** al final del envío de cada trama.

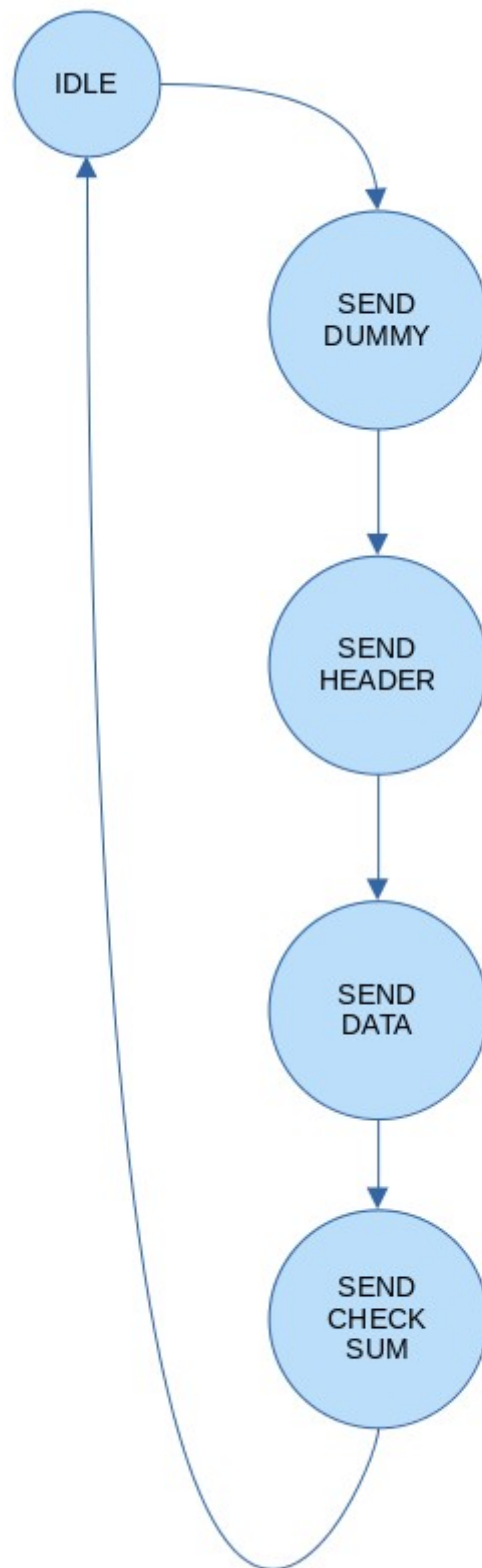
También, es necesario tener en cuenta que cuando se recibe el carácter coma, se puede producir un desalineamiento de la cabecera, yéndose parte del dato a dato previo a la recepción del carácter, por lo que se hace necesario plantearse utilizar un dato nulo o **dummy** para poder realinear bien los datos.

## Envío

Para hacer el envío se recomienda el uso de una máquina de estados como la siguiente.

Esta máquina de estados manda en el estado **SEND\_DUMMY** el dato *x"FF000055"*, *que es un dato que se va perder en la comunicación*. En el estado **SEND\_HEADER** manda el dato *x"FF0000BC"* y también manda el dato "0001" por la línea control (*el resto de estados mantienen esta línea a cero*), que va al puerto, **rxctrl2\_out**.

Después, en **SEND\_DATA** se mandan los datos que se quiere transmitir, y por último el **SEND\_CHECKSUM** para garantizar que la alineación no ha sido alterada durante la transmisión de los datos.

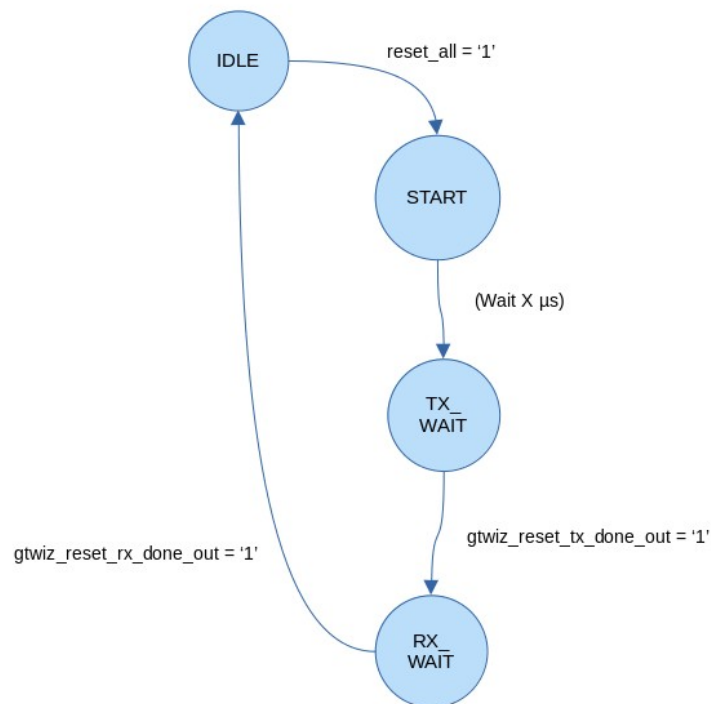


# Enables

- **gtwiz\_reset\_tx\_done\_out** : señal de salida de reset del transceiver para el envío de datos. Esta señal se pone a nivel alto cuando los demás resets de transmisión están a '0'.
- **gtwiz\_reset\_rx\_done\_out** : señal de salida de reset del transceiver para la recepción de datos. Esta señal se pone a nivel alto cuando los demás resets de recepción están a '0'.
- **gtwiz\_reset\_tx\_pll\_and\_datapath\_in** : señal de entrada de reset para tx, al activarlo la señal *gtwiz\_reset\_tx\_done\_out* se pone a nivel bajo
- **gtwiz\_reset\_tx\_datapath\_in** : señal de entrada de reset para tx, al activarlo la señal *gtwiz\_reset\_tx\_done\_out* se pone a nivel bajo
- **gtwiz\_reset\_all\_in** : esta señal de entrada solo afecta a la configuración inicial
- **gtwiz\_reset\_rx\_datapath\_in** : esta señal no influye en dejar de recibir datos
- **gtwiz\_reset\_rx\_pll\_and\_datapath\_in** : esta señal no influye en dejar de recibir datos

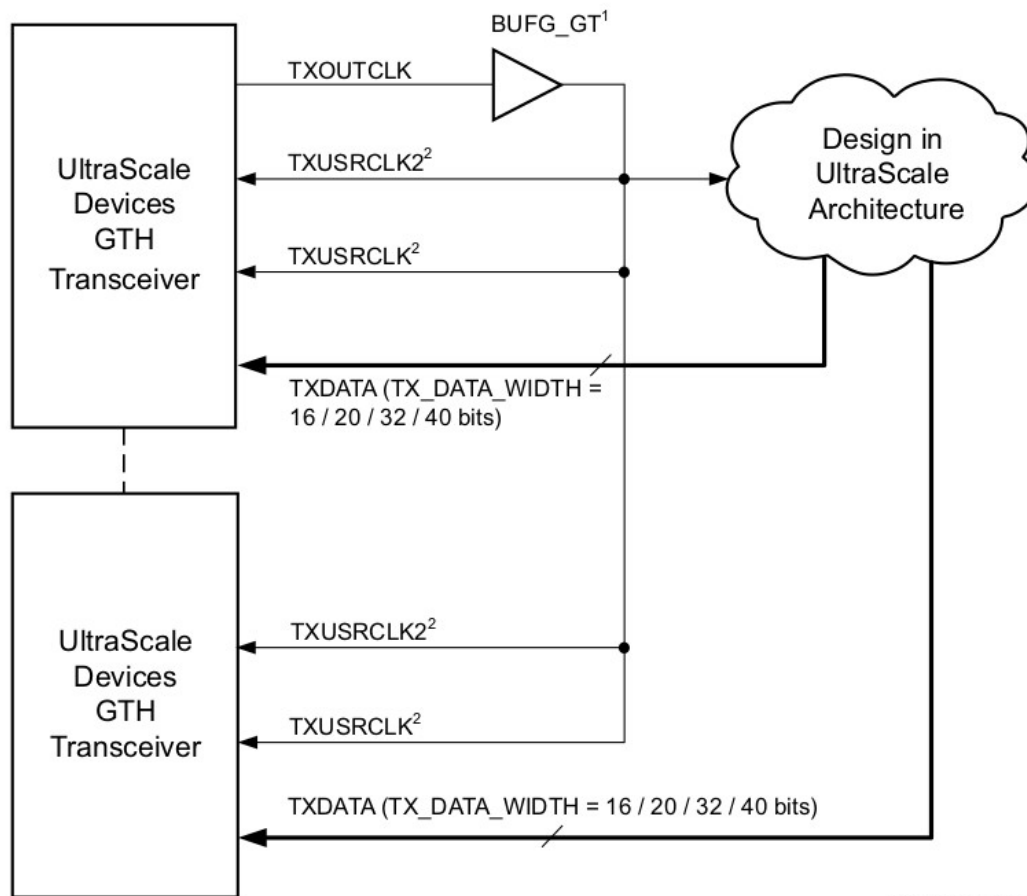
# Arraque

Para el arranque se puede utilizar una máquina de estados como la siguiente, esta espera a que todo este correcto para empezar la comunicación.





Ejemplos de fabricante para relojes(documentación):



UG576\_c3\_03\_050217

