Transceivers

Xilinx

Por 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
gtwiz_reset_clk_freerun_in	In	1	Entrada de reloj del sistema, nunca superior a 250MHz
gtwiz_userclk_tx_active_in	In	1	Entrada de activación de Tx. Esta entrada tiene que estar sincronizada. Activa a nivel alto.
gtwiz_userclk_rx_active_in	In	1	Entrada de activación de Rx. Esta entrada tiene que estar sincronizada. Activa a nivel alto.
gtwiz_reset_all_in	In	1	Reset global del Transceiver
gtwiz_userdata_tx_in	In	32/40 x N.º canal	Salida de datos por el Transceiver
gtwiz_userdata_rx_out	Out	32/40 x N.º canal	Entrada de datos por el transceiver
gtrefclk00_in	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.
gthrxn_in	In	1	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)
gthrxp_in	In	1	
gthtxn_out	Out	1	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)
gthtxp_out	Out	1	

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

txctrl0_in*	In	8 x N.º Transceiver	Señales para alineación. No son necesarias, se ponen a '0'.
txctrl1_in*	In	8 x N.º Transceiver	
txctrl2_in*	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".
rxctrl0_out*	Out	8 x N.º Transceiver	Señales de recepción de la coma. No son necesarias, se dejan abiertas (*open*)
rxctrl1_out*	Out	8 x N.º Transceiver	
rxctrl2_out*	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. *Ejemplo: si se recibe un "0100", quiere decir que tienes los datos partidos en dos.*
rxctrl3_out*	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.

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

Relojes

- **Reloj para Freerun**

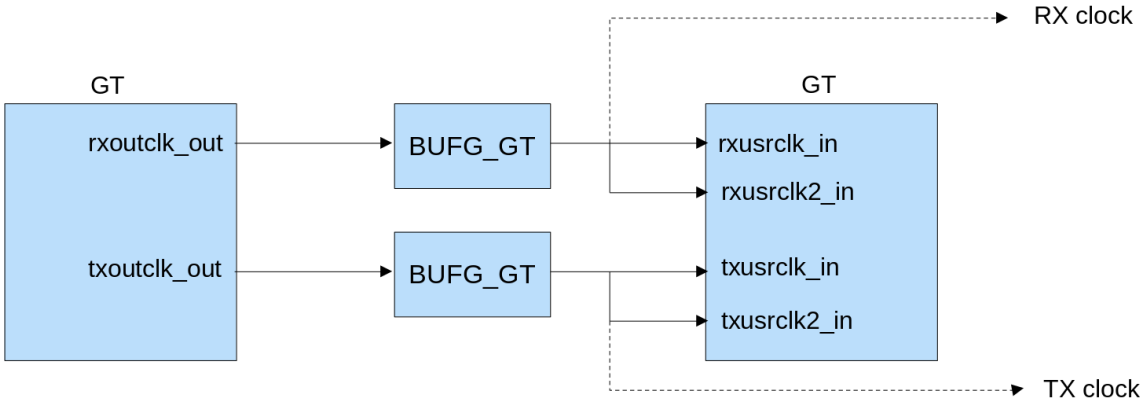
  Con un reloj diferencial se utiliza un IBUFGDS que convierte la señal en una señal de una línea, después, se pasa por un PLL para rebajarla a 100MHz, y por último, se utiliza un BUFG para conseguir la señal CLK_FREERUN.

- **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 r***xusrclk_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.
- Si el dato es “0010”(2) quiere decir
- 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(31:16), dato_ant(15:0)>,<dato_sig(31:16), dato_act(15:0)>, …
- Si el dato es “1000”(8) quiere decir

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.

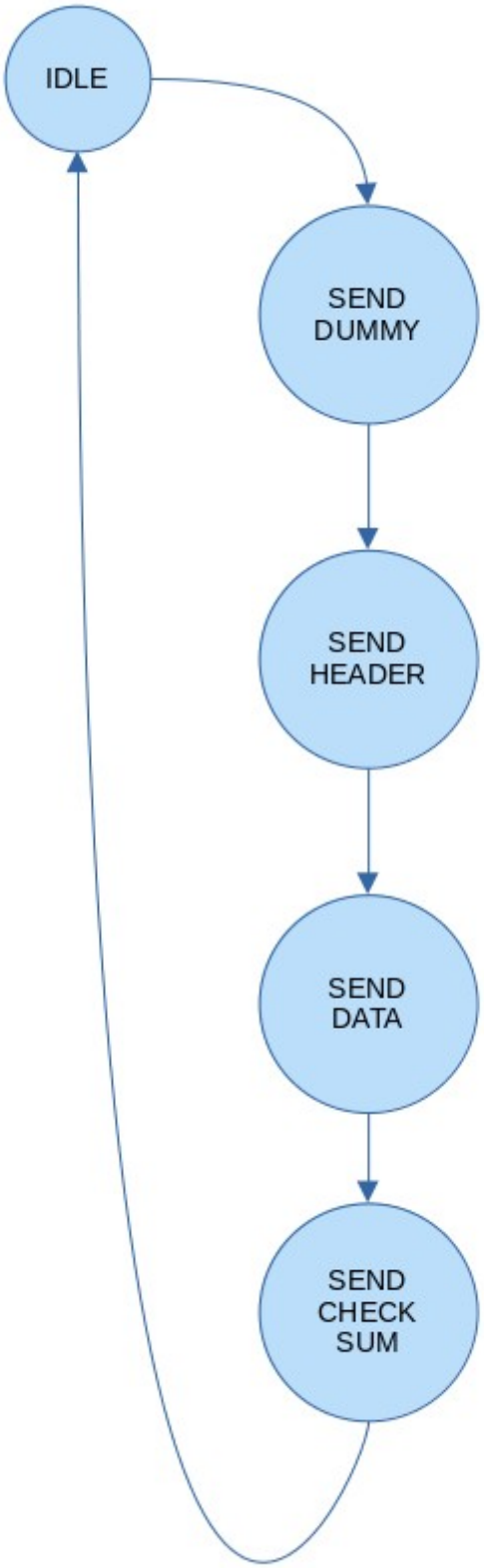
También, es necesario tener en cuenta que cuando se recibe el carácter coma, se puede producir un desalineamiento de la cabecera, yéndose parte del dato a dato previo a la recepción del carácter, por lo que se hace necesario plantearse utilizar un dato nulo o **dummy** para poder realinear bien los datos.

Envío

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

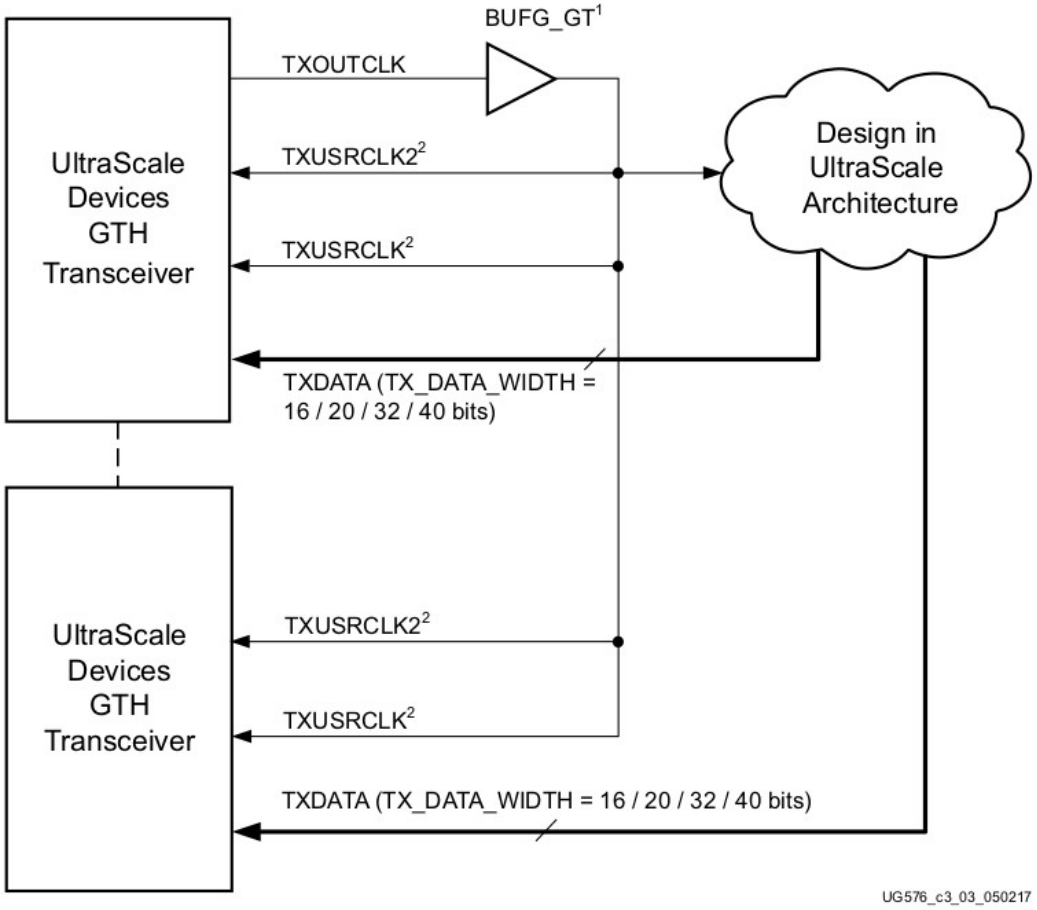
Esta máquina de estados manda en el estado **SEND_DUMMY** el dato *x”FF000055”, que es un dato que se va perder en la comunicación*. En el estado **SEND_HEADER** manda el dato *x”FF0000BC”* y también manda el dato *“0001”* por la línea control *(el resto de estados mantienen esta línea a cero)*, que va al puerto, *rxctrl2_out.*

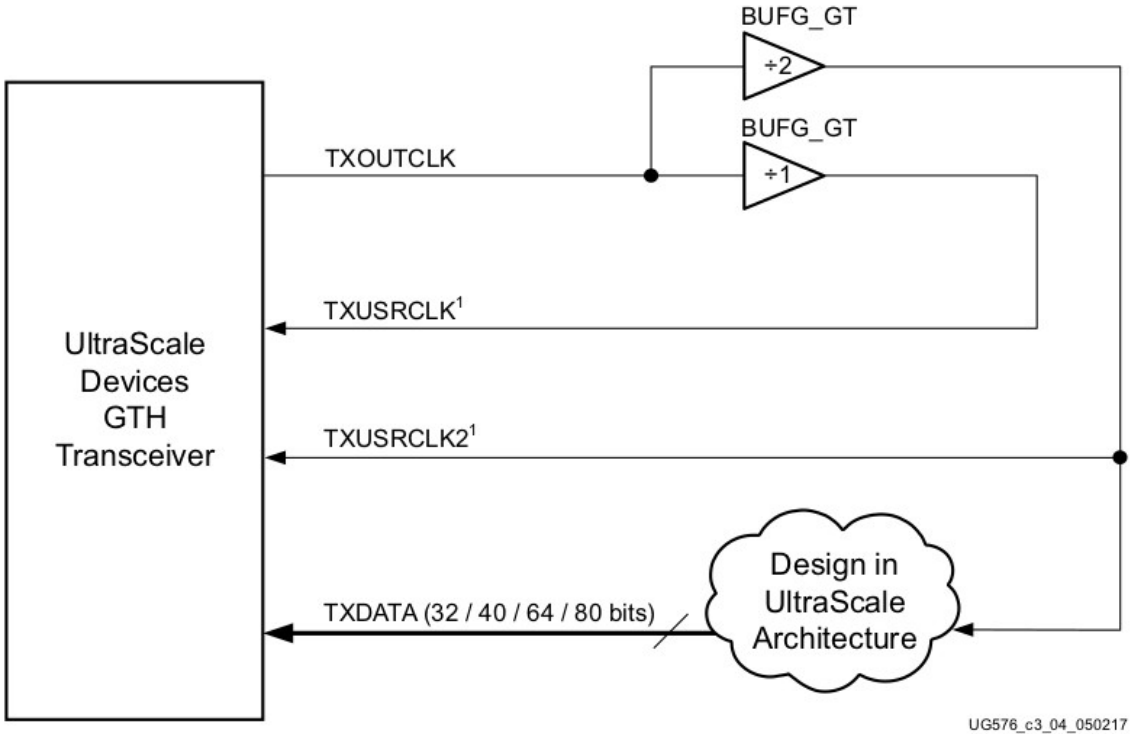
Después en **SEND_DATA** se mandan los datos que se quiere transmitir, y por último el **SEND_CHECKSUM** para garantizar que la alineación no ha sido alterada durante la transmisión de los datos.

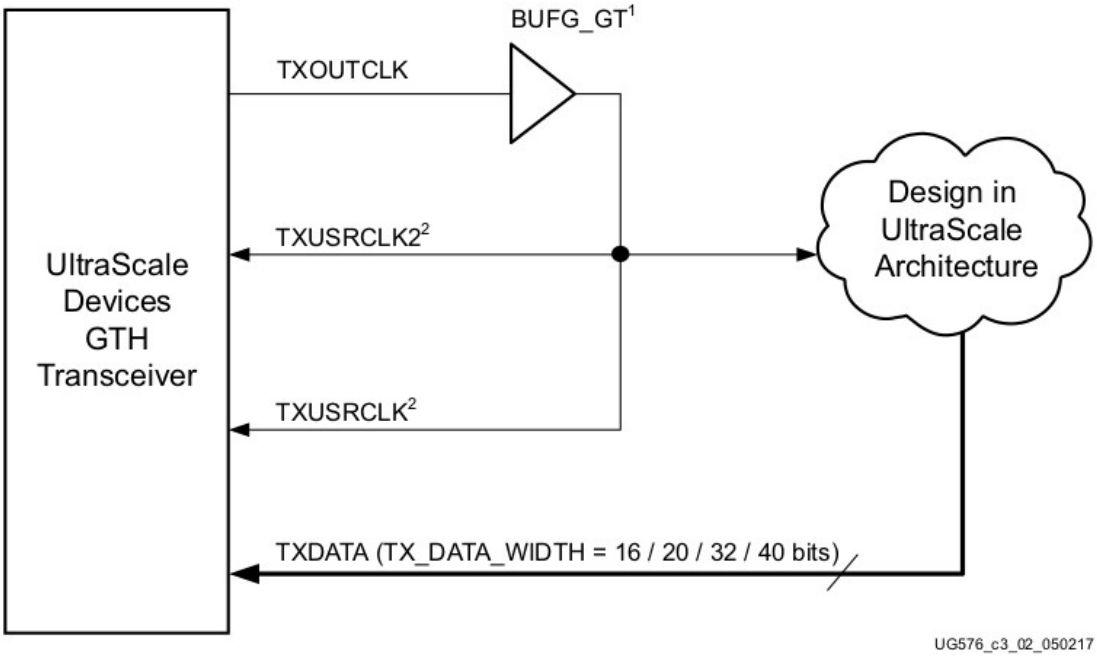

IDLE
SEND DUMMY
SEND HEADER
SEND DATA
SEND CHECK SUM

Enables

Ejemplos de fabricante para relojes(documentación):




BUFG_GT
÷2
BUFG_GT
÷1
TXOUTCLK
TXUSRCLK[1]
TXUSRCLK2[1]
UltraScale Devices GTH Transceiver
Design in UltraScale Architecture
TXDATA (32 / 40 / 64 / 80 bits)
UG576_c3_04_050217


BUFG_GT[1]
TXOUTCLK
TXUSRCLK2[2]
TXUSRCLK[2]
UltraScale Devices GTH Transceiver
Design in UltraScale Architecture
TXDATA (TX_DATA_WIDTH = 16 / 20 / 32 / 40 bits)
UG576_c3_02_050217