Sistema de Monitoreo de Estaciones de Medición

Trabajo Práctico Final de Microarquitecturas y Softcores

Creación de dos IP Cores Transmisor y Receptor Serializadores para el intercambio de datos entre módulos

Roberto Oscar Axt

(roberto.axt@gmail.com)

14/06/2025

Historial de cambios

Versión	Fecha	Descripción	Autor	Revisores
A	14/06/2025	Versión Original	Roberto Axt	

Índice de contenido

1. Introducción **4**
1.1. Contexto 4
1.2. Propuesta del Submódulo de Transmisión 5
1.3. Propuesta del Submódulo de Recepción 5
2. Proyecto **6**
2.1. Formato de carpetas 6
2.2. IP Cores 6
2.2.1. Implementación del Transmisor 6
2.2.2. Implementación del Receptor 7
2.2.3. Posibles Mejoras 7
2.3. Proyecto Transceiver 8
2.3.1. HDL 8
2.3.2. Programación 8
3. Anexo I: Reporte de utilización **9**

1. Introducción

1.1. Contexto

El proyecto final de MIoT consiste en el desarrollo de un *Sistema de Monitoreo para Estaciones de Medición de Gas Natural* (SMEM). Estas estaciones están compuestas por shelters, en cuyo interior se alojan los sistemas de medición, energía y comunicaciones.

La mayoría de estas estaciones reciben energía eléctrica de la red, suministrada por cooperativas locales. Además, cuentan con un sistema básico de respaldo para casos de corte de suministro. Se ha identificado una necesidad clave: determinar el origen de las fallas eléctricas, es decir, si estas se deben a problemas en la red de suministro externa o a inconvenientes internos en la estación.

El sistema SMEM estará compuesto por dos módulos:

- **Módulo Central de Control**, en adelante denominado **MCC**, ubicado dentro del shelter.
- **Módulo de Supervisión Remota**, en adelante denominado **MSR,** instalado en el pilar de energía.

El módulo MSR tendrá dos sensores de presencia de 220 V: uno colocado antes y otro después del disyuntor o térmica principal de la estación. Además, contará con dos sensores tipo reed switch para la detección de manipulaciones (*tampering*). Estas cuatro señales serán serializadas y enviadas al módulo MCC a través de un submódulo de transmisión mediante un enlace de 433 MHz.

El módulo MCC recibirá los 4 bits de un submódulo de recepción, el cual se encarga de la decodificación de los datos seriales y su verificación.

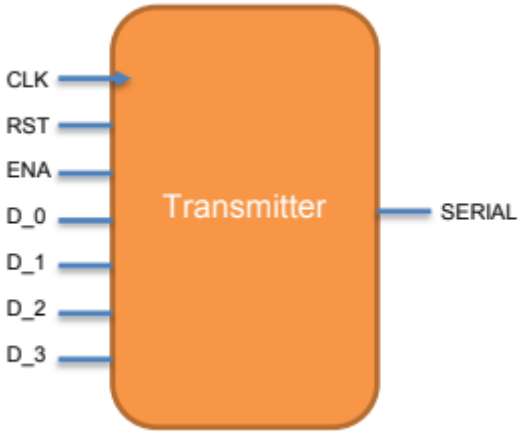
EM&R El Chaja

1.2. Propuesta del Submódulo de Transmisión

El submódulo de transmisión recibe los 4 bits en paralelo de las señales mencionada y se encarga de serializar estos datos. Además, agrega una secuencia de 3 bits denominada preámbulo para mejorar las características de sincronización y otra secuencia de 3 bits de CRC-3 para poder validar que los datos fueron recibidos correctamente.

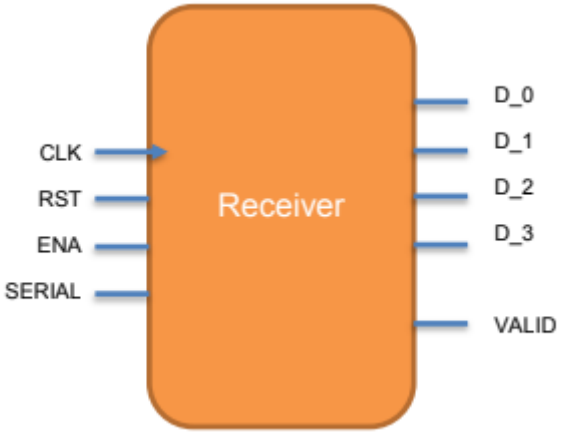
Donde los datos seriales de salida, tendrán la siguiente forma

Preámbulo			Datos				CRC			Idle
0	1	0	X	X	X	X	Y	Y	Y	1



1.3. Propuesta del Submódulo de Recepción

El submódulo de recepción recibe la secuencia mencionada. El mismo inicia esperando la secuencia del preámbulo para sincronizar el payload, que son los datos y el CRC-3. Una vez recibidos estos 7 bits verifica que la secuencia de CRC sea correcta y envía los datos a la salida junto con una señal de validación.



2. Proyecto

2.1. Formato de carpetas

El proyecto se encuentra en el siguiente repositorio

https://github.com/RobAxt/MyS_workspace/tree/main/TPFinal

Está compuesto por las carpetas

- **Documentos:** carpeta con este documento y reporte de utilización.
- **Fuentes**
  - **Implementación:** carpeta con los códigos fuentes de la implementación del transmisor y receptor para ser encapsulados en los IP cores respectivamente. También se encuentra el archivo de restricciones y el código fuente en C para la ejecución de los IP cores.
  - **Receptor:** carpeta con el código fuente del receptor y un testbench donde se simulan casos exitosos y de falla del conjunto transmisor receptor, denominado transceiver.
  - **Transmisor:** carpeta con el código fuente del transmisor y un testbench donde se valida la trama de salida.
- **Síntesis:** proyecto completo con la integración de los IP Cores del repositorio.
- **Repositorio:** IP Cores del transmisor y del receptor.

2.2. IP Cores

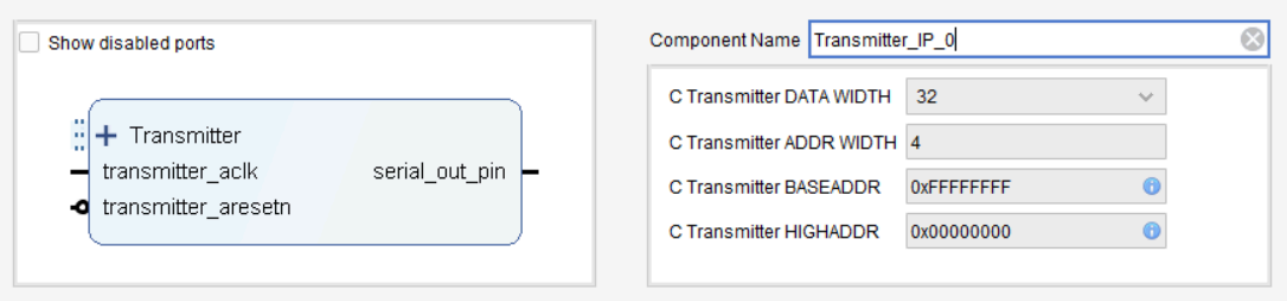
El objetivo del trabajo final es encapsular los submódulos mencionados dentro de un repositorio local en dos IP cores distintos, uno para el transmisor y otro para el receptor.

2.2.1. Implementación del Transmisor

A diferencia de la implementación del TP anterior, en este submódulo se removieron los metaharden ya que los datos de entrada van a provenir de los registros del bus axi y no de periféricos externos.

Las entradas del transmisor de "*rst*" y "*ena*", se encuentran harcodeadas, mientras que el clk se toma de bus axi y los datos de entrada son recibidos a partir del registro "*slv_reg0*".

Para poder salir al exterior con la señal serializada "*serial_out_pin*" se agrega dicha señal a la lista de puertos de las entidades: "*Transmitter_IP_v1_0_Transmitter*" y "*Transmitter_IP_v1_0*".



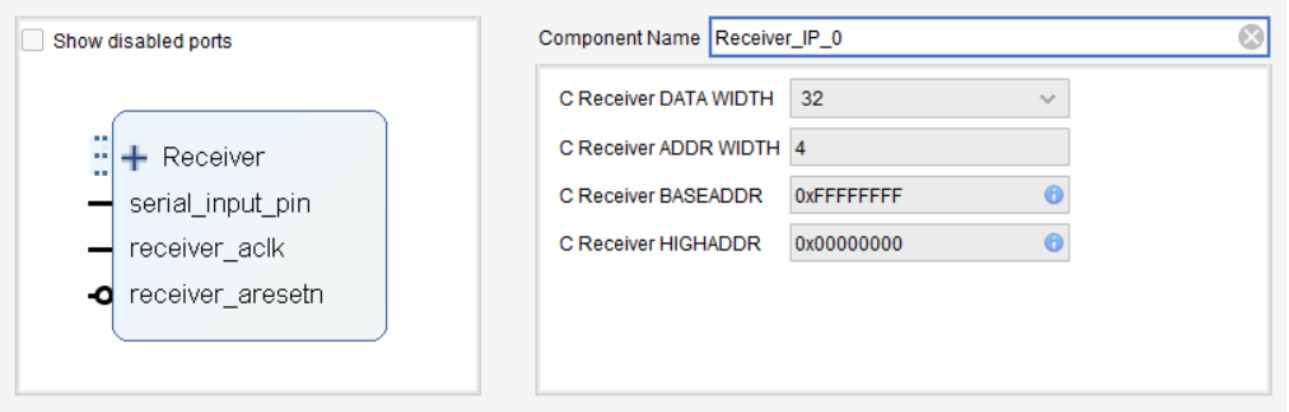
2.2.2. Implementación del Receptor

En esta implementación se deja solamente el metaharden del pin "*serial_input_pin*", ya que este proviene del exterior.

De la misma manera que en el transmisor se harcodean las señales de "*rst"* y "*ena*".

Para la lectura de los datos recibidos se crea la señal "*slv_out_reg0*", asignándole offset cero para el registro de bus axi. En la misma los bits de nibble lsb son usados para transportar los datos (*slv_out_reg0(3 downto 0)*) y la señal de validación ocupa el siguiente bit del otro nibble (*slv_out_reg0(4)*)

Para poder salir al exterior con la señal serializada "*serial_out_pin*" se agrega dicha señal a la lista de puertos de las entidades: "*Receiver _IP_v1_0_ Receiver*" y "*Receiver_IP_v1_0*".



2.2.3. Posibles Mejoras

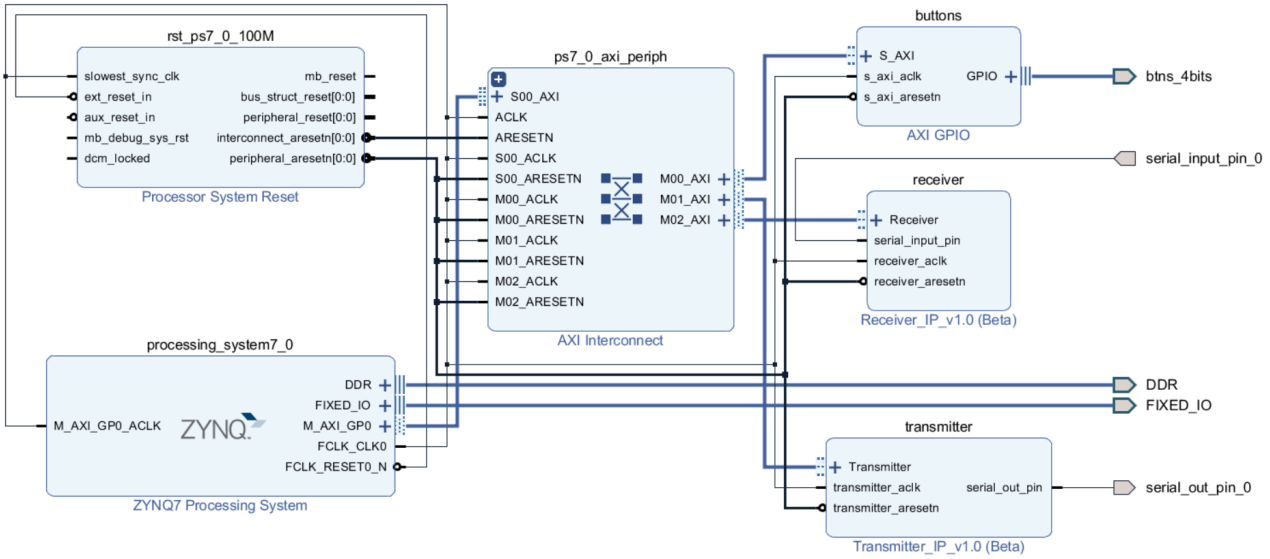
- No hardcodear los estados de "rst" y "ena"
    - En el transmisor se pudo haber usado dos bits del nibble msl del primer byte del registro "*slv_reg0*" o bien usar dos bits del registro "*slv_reg1*".

- En el receptor se pudo haber usado dos bits del registro "*slv_reg1*"

2.3. Proyecto Transceiver

2.3.1. HDL

En la figura se muestra el diagrama en bloques del sistema transceiver, el cual esta compuesto por microcontrolador Zynq7, la matriz de interconexión del bus axi, el clk del sistema y los dispositivos axi para la lectura de los botones, la transmisión del estado de los botones y la recepción de los mismos.



2.3.2. Programación

La aplicación cargada en el microcontrolador obtiene el estado de los botones a partir del módulo AXI_GPIO, los cuales son enviados por otro maestro axi al transmisor. Los datos serializados son recibidos por el receptor, quien envía al microcontrolador por otro bus axi los datos para decodificar.

```
Serial: (COM4, 115200, 8, 1, None, None - CONNECTED)
Buttons Status A
Data Transmitted: A
Raw Data Received: 1A
Data Valid: true and Data Received A
```

```
Serial: (COM4, 115200, 8, 1, None, None - CONNECTED)
Buttons Status A
Data Transmitted: A
Raw Data Received: 0
Data Valid: false and Data Received 0
```

3. Anexo I: Reporte de utilización

```
Copyright 1986-2018 Xilinx, Inc. All Rights Reserved.
------------------------------------------------------------------------------------
------------------------------------------------
| Tool Version : Vivado v.2018.1 (win64) Build 2188600 Wed Apr  4 18:40:38 MDT
2018
| Date         : Sat Jun 14 18:18:52 2025
| Host         : NB459408 running 64-bit major release  (build 9200)
| Command      : report_utilization -file
C:/Xilinx/MyS_workspace/TPFinal/Documentacion/reporteUtilizacion.txt -name
utilization_1
| Design       : transceiver_wrapper
| Device       : 7z010clg400-1
| Design State : Routed
------------------------------------------------------------------------------------
------------------------------------------------

Utilization Design Information

Table of Contents
-----------------
1. Slice Logic
1.1 Summary of Registers by Type
2. Slice Logic Distribution
3. Memory
4. DSP
5. IO and GT Specific
6. Clocking
7. Specific Feature
8. Primitives
9. Black Boxes
10. Instantiated Netlists

1. Slice Logic
--------------

+----------------------------+------+-------+-----------+-------+
|          Site Type         | Used | Fixed | Available | Util% |
+----------------------------+------+-------+-----------+-------+
| Slice LUTs                 |  734 |     0 |     17600 |  4.17 |
|   LUT as Logic             |  672 |     0 |     17600 |  3.82 |
|   LUT as Memory            |   62 |     0 |      6000 |  1.03 |
|     LUT as Distributed RAM |    0 |     0 |           |       |
|     LUT as Shift Register  |   62 |     0 |           |       |
| Slice Registers            | 1178 |     0 |     35200 |  3.35 |
|   Register as Flip Flop    | 1178 |     0 |     35200 |  3.35 |
|   Register as Latch        |    0 |     0 |     35200 |  0.00 |
| F7 Muxes                   |    0 |     0 |      8800 |  0.00 |
| F8 Muxes                   |    0 |     0 |      4400 |  0.00 |
+----------------------------+------+-------+-----------+-------+


1.1 Summary of Registers by Type
--------------------------------

+-------+--------------+-------------+--------------+
| Total | Clock Enable | Synchronous | Asynchronous |
+-------+--------------+-------------+--------------+
| 0     |              _ |           - |            - |
```

```
| 0     |              _ |            - |          Set |
| 0     |              _ |            - |        Reset |
| 0     |              _ |          Set |            - |
| 0     |              _ |        Reset |            - |
| 0     |            Yes |            - |            - |
| 0     |            Yes |            - |          Set |
| 0     |            Yes |            - |        Reset |
| 63    |            Yes |          Set |            - |
| 1115  |            Yes |        Reset |            - |
+-------+----------------+--------------+--------------+


2. Slice Logic Distribution
---------------------------

+-------------------------------------------+------+-------+-----------+-------+
|                 Site Type                 | Used | Fixed | Available | Util% |
+-------------------------------------------+------+-------+-----------+-------+
| Slice                                     |  358 |     0 |      4400 |  8.14 |
|   SLICEL                                  |  227 |     0 |           |       |
|   SLICEM                                  |  131 |     0 |           |       |
| LUT as Logic                              |  672 |     0 |     17600 |  3.82 |
|   using O5 output only                    |    0 |       |           |       |
|   using O6 output only                    |  498 |       |           |       |
|   using O5 and O6                         |  174 |       |           |       |
| LUT as Memory                             |   62 |     0 |      6000 |  1.03 |
|   LUT as Distributed RAM                  |    0 |     0 |           |       |
|   LUT as Shift Register                   |   62 |     0 |           |       |
|     using O5 output only                  |    0 |       |           |       |
|     using O6 output only                  |   58 |       |           |       |
|     using O5 and O6                       |    4 |       |           |       |
| LUT Flip Flop Pairs                       |  416 |     0 |     17600 |  2.36 |
|   fully used LUT-FF pairs                 |  127 |       |           |       |
|   LUT-FF pairs with one unused LUT output |  274 |       |           |       |
|   LUT-FF pairs with one unused Flip Flop  |  260 |       |           |       |
| Unique Control Sets                       |   75 |       |           |       |
+-------------------------------------------+------+-------+-----------+-------+
* Note: Review the Control Sets Report for more information regarding control
sets.


3. Memory
---------

+----------------+------+-------+-----------+-------+
|    Site Type   | Used | Fixed | Available | Util% |
+----------------+------+-------+-----------+-------+
| Block RAM Tile |    0 |     0 |        60 |  0.00 |
|   RAMB36/FIFO* |    0 |     0 |        60 |  0.00 |
|   RAMB18       |    0 |     0 |       120 |  0.00 |
+----------------+------+-------+-----------+-------+
* Note: Each Block RAM Tile only has one FIFO logic available and therefore can
accommodate only one FIFO36E1 or one FIFO18E1. However, if a FIFO18E1 occupies a
Block RAM Tile, that tile can still accommodate a RAMB18E1


4. DSP
------

+-----------+------+-------+-----------+-------+
| Site Type | Used | Fixed | Available | Util% |
+-----------+------+-------+-----------+-------+
| DSPs      |    0 |     0 |        80 |  0.00 |
+-----------+------+-------+-----------+-------+
```

```
5. IO and GT Specific
---------------------

+-----------------------------+------+-------+-----------+---------+
|          Site Type          | Used | Fixed | Available |  Util%  |
+-----------------------------+------+-------+-----------+---------+
| Bonded IOB                  |    6 |     6 |       100 |    6.00 |
|   IOB Master Pads           |    4 |       |           |         |
|   IOB Slave Pads            |    2 |       |           |         |
| Bonded IPADs                |    0 |     0 |         2 |    0.00 |
| Bonded IOPADs               |  130 |   130 |       130 |  100.00 |
| PHY_CONTROL                 |    0 |     0 |         2 |    0.00 |
| PHASER_REF                  |    0 |     0 |         2 |    0.00 |
| OUT_FIFO                    |    0 |     0 |         8 |    0.00 |
| IN_FIFO                     |    0 |     0 |         8 |    0.00 |
| IDELAYCTRL                  |    0 |     0 |         2 |    0.00 |
| IBUFDS                      |    0 |     0 |        96 |    0.00 |
| PHASER_OUT/PHASER_OUT_PHY   |    0 |     0 |         8 |    0.00 |
| PHASER_IN/PHASER_IN_PHY     |    0 |     0 |         8 |    0.00 |
| IDELAYE2/IDELAYE2_FINEDELAY |    0 |     0 |       100 |    0.00 |
| ILOGIC                      |    0 |     0 |       100 |    0.00 |
| OLOGIC                      |    0 |     0 |       100 |    0.00 |
+-----------------------------+------+-------+-----------+---------+


6. Clocking
-----------

+------------+------+-------+-----------+-------+
|  Site Type | Used | Fixed | Available | Util% |
+------------+------+-------+-----------+-------+
| BUFGCTRL   |    1 |     0 |        32 |  3.13 |
| BUFIO      |    0 |     0 |         8 |  0.00 |
| MMCME2_ADV |    0 |     0 |         2 |  0.00 |
| PLLE2_ADV  |    0 |     0 |         2 |  0.00 |
| BUFMRCE    |    0 |     0 |         4 |  0.00 |
| BUFHCE     |    0 |     0 |        48 |  0.00 |
| BUFR       |    0 |     0 |         8 |  0.00 |
+------------+------+-------+-----------+-------+


7. Specific Feature
-------------------

+-------------+------+-------+-----------+-------+
|  Site Type  | Used | Fixed | Available | Util% |
+-------------+------+-------+-----------+-------+
| BSCANE2     |    0 |     0 |         4 |  0.00 |
| CAPTUREE2   |    0 |     0 |         1 |  0.00 |
| DNA_PORT    |    0 |     0 |         1 |  0.00 |
| EFUSE_USR   |    0 |     0 |         1 |  0.00 |
| FRAME_ECCE2 |    0 |     0 |         1 |  0.00 |
| ICAPE2      |    0 |     0 |         2 |  0.00 |
| STARTUPE2   |    0 |     0 |         1 |  0.00 |
| XADC        |    0 |     0 |         1 |  0.00 |
+-------------+------+-------+-----------+-------+


8. Primitives
-------------

+----------+------+----------------------+
```

Ref Name	Used	Functional Category
FDRE	1115	Flop & Latch
LUT3	281	LUT
LUT6	194	LUT
LUT5	165	LUT
LUT4	138	LUT
BIBUF	130	IO
FDSE	63	Flop & Latch
LUT2	52	LUT
SRLC32E	47	Distributed Memory
CARRY4	40	CarryLogic
SRL16E	19	Distributed Memory
LUT1	16	LUT
IBUF	5	IO
PS7	1	Specialized Resource
OBUF	1	IO
BUFG	1	Clock

9. Black Boxes

Ref Name	Used

10. Instantiated Netlists

Ref Name	Used
transceiver_xbar_0	1
transceiver_rst_ps7_0_100M_0	1
transceiver_processing_system7_0_0	1
transceiver_axi_gpio_0_0	1
transceiver_auto_pc_0	1
transceiver_Transmitter_IP_0_0	1
transceiver_Receiver_IP_0_0	1