

MÁSTER OFICIAL EN INGENIERÍA ELECTRÓNICA



VNIVERSITAT
DE VALÈNCIA

TRABAJO DE FIN DE MÁSTER

**SISTEMA DE TRANSFERENCIA DE DATOS EN EL
INSTRUMENTO TILEPPR DEL PROYECTO TILECAL**

AUTOR:

**DANIEL ALEJANDRO FLORES
PÉREZ**

TUTOR:

**JOSÉ FRANCISCO
JORDÁN MARTÍNEZ**

CO-TUTOR:

**JOSÉ ALBERTO
VALERO BIOT**

SEPTIEMBRE, 2017



MÁSTER OFICIAL EN INGENIERÍA ELECTRÓNICA

TRABAJO DE FIN DE MÁSTER

SISTEMA DE TRANSFERENCIA DE DATOS EN EL INSTRUMENTO TILEPPR DEL PROYECTO TILECAL

AUTOR:

**DANIEL ALEJANDRO FLORES
PÉREZ**

TUTOR:

**JOSÉ FRANCISCO
JORDÁN MARTÍNEZ**

CO-TUTOR:

**JOSÉ ALBERTO
VALERO BIOT**

TRIBUNAL:

PRESIDENTE/A:

VOCAL 1:

VOCAL 2:

FECHA DE DEFENSA:

CALIFICACIÓN:

Declaración

Este trabajo final de Máster es fruto de mi propio esfuerzo y labor, salvo cuando se hace referencia explícita al trabajo de otros y que no se ha presentado para otra calificación para esta u otra universidad.

Daniel Alejandro Flores Pérez

Agradecimientos

Quiero expresar mi gratitud a todas las personas e instituciones que me brindaron su apoyo y confianza para llevar a cabo este trabajo de investigación.

En primer lugar quiero agradecer al Dr. José Francisco Jordán Martínez por su confianza y valiosa dirección y al Dr. Alberto Valero Biot por la recepción en el grupo de investigación del TileCal y el apoyo en el desarrollo de este proyecto. Sin vosotros, esto no habría sido posible.

Agradezco al IFIC-Valencia por su acogida y el apoyo recibido a lo largo del periodo en el cual he desarrollado mi labor investigadora. También me complace agradecer la acogida, el apoyo y los medios recibidos de la Universitat de València - Escola Técnica Superior d'Enginyeria, donde he cursado mi maestría.

Debo un especial reconocimiento a la Universidad Nacional Autónoma de Honduras, su Facultad de Ingeniería y a su Departamento de Ingeniería Eléctrica por la confianza que mostraron en mí al concederme una beca del Programa de Relevo Docente a través de la Vicerrectoría de Relaciones Exteriores de la UNAH, la cual me ha facilitado la movilidad necesaria y estadía para poder desarrollar mi maestría y mi trabajo de investigación.

Gracias a los compañeros del equipo TileCal en el IFIC-Valencia por la ayuda prestada. También agradezco a mis compañeros de máster por el tiempo compartido, trabajo en equipo y por hacer de los momentos más pesados algo más llevadero.

Mi más profundo agradecimiento a mi familia, principalmente a mi madre Glenda Pérez, a mi padre German Flores y a mis hermanos German y Raúl. Este también es su premio.

A todos ustedes, mi mayor reconocimiento y gratitud.

Contenido

LISTA DE FIGURAS.....	IX
LISTA DE TABLAS	XV
I. OBJETIVOS Y RESUMEN DEL PROYECTO	1
I. 1. Objetivos.....	1
I. 2. Resumen	2
II. ESPECIFICACIONES DE DISEÑO	7
II. 1. Equipo disponible para pruebas del sistema FELIX en el IFIC.....	7
II. 2. Especificaciones de la programación hardware del TilePPr para el sistema FELIX	9
II. 2. 1 <i>E-link</i>	10
II. 2. 2 <i>Especificaciones del módulo hardware del integrador</i>	10
II. 2. 3 <i>Especificaciones del módulo hardware del FE-I4</i>	10
III. INTRODUCCIÓN.....	11
III. 1. CERN e IFIC-Valencia.....	11
III. 1. 1 <i>ATLAS y el modelo estándar</i>	11
III. 1. 2 <i>La interacción entre el CERN y el IFIC</i>	12
III. 2. Instrumentos del experimento ATLAS.....	13
III. 2. 1 <i>TileCal: Hadronic Tile Calorimeter</i>	13
III. 2. 2 <i>El TilePPr</i>	15
III. 2. 3 <i>Los demostradores</i>	15
III. 2. 4 <i>Sistema de lectura FELIX</i>	16
III. 2. 5 <i>Sistema de lectura del integrador</i>	17
III. 2. 6 <i>Sistema de lectura del FE-I4</i>	18
IV. SOLUCIONES EXISTENTES.....	21
IV. 1. FELIX en el TileDAQ: El enrutador central.....	21

IV. 2.	FELIX y el TilePPr: Señales de reloj.....	23
IV. 3.	FELIX y el TilePPr: La interfaz electrónica E-link.....	24
IV. 4.	FELIX y el TilePPr: Vista superficial de su programación VHDL.....	25
IV. 4. 1	<i>Processor (sROD_processing_module.vhd)</i>	26
IV. 4. 2	<i>FLX.Felix (tile_felix_interface.vhd)</i>	27
IV. 4. 3	<i>System (sROD_system_module.vhd)</i>	28
IV. 5.	El módulo integrator_receiver en el TilePPr.....	28
IV. 6.	Propuesta del FE-I4 y el sistema FELIX.....	31
V.	ESTUDIO PREVIO DE LAS SOLUCIONES POSIBLES.....	33
V. 1.	Puesta en marcha del FLX.Felix.....	33
V. 2.	Software para la detección de errores más específicos.....	34
V. 3.	Programación hardware del módulo que comunique el integrator_receiver.vhd al servidor local.....	34
V. 4.	Programación hardware del sistema de comunicación entre el FE-I4 y el servidor local.....	35
VI.	SOLUCIÓN PROPUESTA Y ESPECIFICACIONES FINALES.....	37
VI. 1.	Puesta en marcha del FLX.Felix.....	37
VI. 1. 1	<i>Programación hardware adicional y verificación con Chipscope.</i>	37
VI. 1. 2	<i>Configuración del e-link en el lado del servidor.</i>	46
VI. 1. 3	<i>Conexión estable entre TilePPr y TileDAQ</i>	46
VI. 2.	Software para la detección de errores más específicos.....	47
VI. 3.	Programación hardware del módulo que comunique el integrador al servidor local.....	53
VI. 3. 1	<i>Propuesta de la trama de datos para el integrador</i>	53
VI. 3. 2	<i>Módulo INT2HOST (integrator2host_top.vhd)</i>	55
VI. 3. 3	<i>INT2FIFO_MD1 (integrator2fifo_top.vhd)</i>	56
VI. 3. 3. 1	<i>EXDATA_S2F_MD1 (example_data_generator.vhd)</i>	57
VI. 3. 3. 2	<i>SS2F_D1 y SS2F_D2 (source_integrator2fifoMD1.vhd)</i>	58
VI. 3. 3. 3	<i>INT_MD1 (integrator_receiver.vhd)</i>	58
VI. 3. 3. 4	<i>IWEAclk80toclk40 e IWEBclk80toclk40 (Integ_wAB_clk80toclk40.vhd)</i>	58
VI. 3. 3. 5	<i>P2F_MD1 (integrator_pack2fifo_16bits)</i>	59
VI. 3. 3. 6	<i>S2F_MD1 (integrator_side2fifo_16bits)</i>	60
VI. 3. 4	<i>elink_tx_INT2HOST (FIFO2Elink.vhd)</i>	61
VI. 3. 5	<i>FIFO2GBT (Interface2_gbt_example_design.vhd)</i>	62
VI. 4.	Sistema de comunicación del FE-I4.....	62
VI. 4. 1	<i>Propuesta de la trama de datos para el FE-I4</i>	62

VI. 4. 2	<i>Tx y Rx del FE-I4 en la VC707 y la KC705</i>	64
VI. 4. 3	<i>Bloques IP de la librería de VIVADO</i>	64
VI. 4. 4	<i>Bloques IP creados en este trabajo</i>	65
VI. 4. 5	<i>Estandar de comunicación: LVDS y LVDS_25</i>	68
VII.	NORMATIVA	69
VII. 1.	ANSI/VITA 57.1-2008.....	69
VII. 2.	Dimensiones de la PCB	70
VII. 3.	Interfaz FMC	71
VII. 4.	Pines y señales	71
VIII.	SIMULACIONES	73
VIII. 1.	FLX.Felix.....	73
VIII. 2.	INT2HOST	73
VIII. 3.	FE-I4	76
VIII. 3. 1	<i>Placa electrónica para pruebas FMC-SCSI-HDMI-DisplayPort</i>	76
IX.	ELECCIÓN DE COMPONENTES Y MOTIVACIÓN	81
IX. 1.	FLX.Felix.....	81
IX. 2.	INT2HOST	81
IX. 3.	FE-I4	82
IX. 4.	Interfaces	82
IX. 5.	Terminales diferenciales programables	83
IX. 6.	Bloques IO de la FPGA	84
IX. 7.	Reloj y los IOSERDES2	87
IX. 8.	Elección de FIFO	88
X.	INTERCONEXIÓN Y SOFTWARE	91
X. 1.	Conexiones existentes antes del proyecto	91
X. 2.	Programas	92
XI.	DISEÑO DE LA PLACA DE PRUEBAS	93
XI. 1.	Condiciones de diseño eléctrico.....	93
XI. 2.	Condiciones de diseño mecánico.....	96
XI. 3.	Condiciones de diseño térmico	97
XII.	PRUEBAS Y RESULTADOS EXPERIMENTALES	99
XII. 1.	Resultados antes de las modificaciones al felix_interface.vhd y elaboración de procedimiento	99
XII. 2.	Independencia del INT2HOST en el TilePPr: configuración inicial.....	106
XII. 3.	BER: Tasa de error binario.....	110
XII. 4.	Resultados experimentales del FE-I4.....	111
XII. 4. 1	<i>Configuración de FIFO</i>	111

XII. 4. 2	<i>Lazo cerrado</i>	112
XII. 4. 3	<i>Conteo de cabeceras y colas: EMC en el cable SCSI</i>	113
XII. 4. 4	<i>Diferencia entre la frecuencia de la KC705 y la VC707</i>	115
XIII.	PRUEBAS DE VALIDACIÓN Y CERTIFICACIÓN	119
XIII. 1.	FLX.Felix	120
XIII. 1. 1	<i>Respuesta del sistema ante distintas frecuencias de L1A</i>	120
XIII. 2.	INT2HOST	122
XIII. 3.	FE-I4.....	123
XIV.	MANUAL DE USUARIO	125
XV.	COSTES DE FABRICACIÓN	127
XVI.	CONCLUSIONES Y TRABAJOS FUTUROS	129
XVII.	REFERENCIAS BIBLIOGRÁFICAS	133
XVIII.	PLANOS Y ANEXOS	137
ANEXO I.	VISTA DEL ESQUEMÁTICO DE LA PLACA DE PRUEBAS	139
ANEXO II.	INTERFACE FE-I4 EN EL VC707: DOS CANALES RX Y DOS CANALES TX	141
ANEXO III.	INTERFACE FE-I4 EN EL VC707: VALIDACIÓN DE DOS CANALES RX	143
ANEXO IV.	INTERFACE FE-I4 EN EL KC705: DOS CANALES RX Y DOS CANALES TX	145
ANEXO V.	VISTA DE LAS PISTAS EN LA PLACA DE PRUEBAS (CAPTURA DE LOS ARCHIVOS GERBER).	147
ANEXO VI.	ORDEN DE COMPRA PARA LA PLACA DE PRUEBAS.....	149
XIX.	LISTA DE ACRÓNIMOS Y SIGLAS	151

Lista de Figuras

I.1	<i>Módulos de programación hardware objetivos en este trabajo.</i>	<i>1</i>
I.2	<i>Arquitectura del sistema de pruebas de FELIX en el laboratorio TileCal del IFIC-Valencia</i>	<i>2</i>
I.3	<i>Arquitectura del sistema de pruebas de FE-I4 propuesto.</i>	<i>3</i>
III.1	<i>Lecturas obtenidas del ATLAS que llevaron al anuncio de la observación de una nueva partícula en la región de masa de 126 GeV.</i>	<i>12</i>
III.2	<i>Participación del IFIC-Valencia en el proyecto ATLAS.....</i>	<i>12</i>
III.3	<i>Soportes de acero que conforman las secciones del TileCal.....</i>	<i>13</i>
III.4	<i>Ubicación de la electrónica en los soportes de acero.</i>	<i>14</i>
III.5	<i>Diagrama de la arquitectura de lectura de TileCal para la segunda fase de actualización de ATLAS.</i>	<i>15</i>
III.6	<i>Diagrama de bloques que muestra el flujo de la señal del sistema "Demonstrator".. ..</i>	<i>16</i>
III.7	<i>Diagrama de bloques que muestra el flujo de la señal del sistema con la actualización de alta luminosidad.</i>	<i>16</i>
III.8	<i>Arquitectura del FELIX.....</i>	<i>16</i>
III.9	<i>Vista funcional del FELIX.....</i>	<i>17</i>
III.10	<i>Ruta de la señal del TileCal con sus diferentes sistemas de monitoreo.</i>	<i>17</i>
III.11	<i>Adquisición de datos 27262 en el año 2012.</i>	<i>18</i>

III.12	<i>Enlaces e interfaces programados en la FPGA de la placa de evaluación KC705 para el FE-I4.....</i>	19
III.13	<i>Arquitectura que une el FE-I4 con el sistema FELIX a través del GBT programado en el TilePPr.....</i>	19
IV.1	<i>Comparación entre el sistema de lectura de TileCal y la segunda fase de actualización.....</i>	21
IV.2	<i>Captura tomada de la interfaz gráfica “FELIX E-LINK Configurator”.....</i>	22
IV.3	<i>Fibras de conexión entre el TTCex y las placas electrónicas de prueba: puerto 1 hacia la TileDAQ y el puerto 4 hacia el TilePPr.....</i>	23
IV.4	<i>Diagrama de conexión entre el TilePPr y el generador de reloj en el sistema de pruebas de IFIC-Valencia.....</i>	24
IV.5	<i>Esquema RTL: Vista de los tres módulos principales que componen a la interfaz FELIX en la FPGA del VC707.....</i>	25
IV.6	<i>Esquema RTL: Relación de algunas señales entre los módulos Processor y FLX.Felix.....</i>	26
IV.7	<i>Esquema RTL: Relación entre el módulo elink_tx y el mmcmfelixclocks dentro del módulo FLX.Felix.....</i>	27
IV.8	<i>Esquema RTL: Vista top del gbtExmplDsgn instanciado dentro del módulo FLX.Felix.....</i>	28
IV.9	<i>Propuesta para la adquisición de datos del FE-I4.....</i>	31
VI.1	<i>Montaje para el monitoreo por osciloscopio de las señales de reloj.....</i>	37
VI.2	<i>Señal rxData_from_gbtExmplDsg mostrada en esquema RTL.....</i>	39
VI.3	<i>Formas de onda obtenidas por Chipscope con frecuencia de muestreo de 160 MHz.....</i>	43
VI.4	<i>Duración de la señal trigger L1A y separación mínima antes del próximo trigger L1A.....</i>	43
VI.5	<i>Señal de FLX_DATA_IN resultante de tres pulsos consecutivos L1A.....</i>	43
VI.6	<i>Señales de reloj de la VC707 obtenidas por osciloscopio.....</i>	47
VI.7	<i>Estructura del fragmento y componentes considerados para el análisis por software.....</i>	48
VI.8	<i>Datos extraídos de cada canal con ayuda del software unboxing_v01_1.py.</i>	49
VI.9	<i>Imagen del software Unboxing v.1.1, la primer ventana brinda opciones para el análisis general del archivo binario y sus bloques.....</i>	50
VI.10	<i>Imagen del software Unboxing v.1.1, la segunda ventana brinda opciones para el análisis de los fragmentos.....</i>	51

VI.11	<i>Imagen del software Unboxing v.1.1, la tercera y última ventana brinda opciones para el análisis de los canales y crea el archivo de los datos obtenidos.....</i>	<i>52</i>
VI.12	<i>Esquema RTL Top del módulo INT2HOST.....</i>	<i>56</i>
VI.13	<i>Vista del lado A (SECTION-A) y el lado B (SECTION-B) de la placa electrónica principal.....</i>	<i>56</i>
VI.14	<i>Esquema RTL: Top del módulo INT2FIFO_MD1</i>	<i>57</i>
VI.15	<i>Esquema RTL: generador de datos para las pruebas y los módulos SS2F.</i>	<i>57</i>
VI.16	<i>Esquema RTL: selector de fuente SS2F_D1.</i>	<i>58</i>
VI.17	<i>Esquema RTL: prolongadores de duración de las señales integ_weA e integ_weB.</i>	<i>59</i>
VI.18	<i>Esquema RTL: Top del P2F_MD1</i>	<i>59</i>
VI.19	<i>Esquema RTL: S2F_MD1 para un solo E-Proc de 2 bits</i>	<i>60</i>
VI.20	<i>Esquema RTL: elink_tx_INT2HOST.</i>	<i>61</i>
VI.21	<i>Esquema RTL: FIFO2GBT</i>	<i>62</i>
VI.22	<i>Esquema sobre el manejo del ISERDES según Xilinx.</i>	<i>63</i>
VI.23	<i>Bloques IP de VIVADO: SelectIO Interface Wizard para OSERDES2</i>	<i>64</i>
VI.24	<i>Bloque IP de VIVADO: FIFO Generator.....</i>	<i>65</i>
VI.25	<i>Bloques IP de VIVADO: Clocking Wizard y Processor System Reset.....</i>	<i>65</i>
VI.26	<i>Bloque IP creado en este trabajo: controlBitslip_v1_0.</i>	<i>65</i>
VI.27	<i>Bloque IP sacado del código verilog del SelectIO Interface Wizard: IO_SERDES_IP_selectio_wiz_0_v1_0.....</i>	<i>66</i>
VI.28	<i>Bloque IP creado en este trabajo: DataGenerator_example_ip_V1_0.</i>	<i>66</i>
VI.29	<i>Bloque IP creado en este trabajo: emptynot_FIFO_ip_v1_0.....</i>	<i>66</i>
VI.30	<i>Bloque IP creado en este trabajo: MUX_2to1_ip_v1_0.....</i>	<i>67</i>
VI.31	<i>Bloque IP creados en este trabajo: cannel_reg_ip_v1_0.</i>	<i>67</i>
VI.32	<i>Bloque IP creado en este trabajo: ctrl_kintex_ip_V1_0.....</i>	<i>68</i>
VII.1	<i>Mecánica del módulo FMC de anchura simple de grado comercial según ANSI/VITA 57.1-2008.....</i>	<i>70</i>
VII.2	<i>Regiones de la PCB para módulo FMC según VITA 57.1.....</i>	<i>71</i>
VII.3	<i>Etiquetado del conector del módulo FMC (vista del lado del componente).</i>	<i>71</i>

VIII.1	<i>Resultados de uso post-síntesis: FELIX en el VC707 con el INT2HOST para un solo E-Proc de 2 bits.....</i>	73
VIII.2	<i>Resultados de uso post-síntesis: INT2HOST para un solo e-proc de 2 bits.....</i>	74
VIII.3	<i>Resultados de uso post-síntesis: INT2HOST para dos e-proc de 2 bits.</i>	74
VIII.4	<i>Formas de onda por ChipScope capturadas: trama de datos generados por el módulo ExampleData del INT2HOST para un solo E-proc de 2.....</i>	75
VIII.5	<i>Simulación con SigXplorer: modelo IBIS exportados de Vivado.....</i>	76
VIII.6	<i>Modelo IBIS del pin de entrada.</i>	76
VIII.7	<i>Modelo IBIS del terminal diferencial en las entradas del XC7VX485T.....</i>	77
VIII.8	<i>Simulación con SigXplorer: Especificación de las cuatro capas en el simulador.</i>	77
VIII.9	<i>Simulación con SigXplorer: Canal con la señal de reloj LA01_CC.....</i>	78
VIII.10	<i>Simulación con SigXplorer: Tabla resumen con los resultados de la simulación.</i>	78
VIII.11	<i>Simulación con SigXplorer: Gráficas con los valores de tensión en modo común y diferencial al generar una secuencia pseudo aleatoria de 1024 bits.</i>	79
VIII.12	<i>Simulación con SigXplorer: Diagramas de ojo para el estudio de la señal de reloj LA01_CC.....</i>	79
VIII.13	<i>Resultados de uso post-implementación: FE-I4 en la VC707.....</i>	80
VIII.14	<i>Resultados de uso post-implementación: FE-I4 en la KC705.</i>	80
IX.1	<i>Estrategia de bloques de entrada y salida para aprovechar el mínimo disponible de señales de reloj.....</i>	82
IX.2	<i>Terminación diferencial FD_100 declarada por defecto para estándar LVDS o LVDS_25 en modo conductor driver).</i>	83
IX.3	<i>Terminación diferencial declarada con DIFF_TERM para estándar LVDS o LVDS_25 en modo entrada.....</i>	84
IX.4	<i>Diagrama de señales de un FIFO con interfaz nativa</i>	89
XI.1	<i>Geometría física de tramos diferenciales.</i>	94
XI.2	<i>Descripción del Artículo ML4 para la solicitud de PCB.</i>	94
XI.3	<i>Captura de pantalla: cálculo de impedancia diferencial.</i>	95
XI.4	<i>Ejemplo de la guía de enrutados para vías según TI.....</i>	95
XI.5	<i>Vista de los anti-pads para las vías de las señales diferenciales en el conector J6 (HDMI).....</i>	95

XI.6	<i>Dimensiones de la interfaz SCSI proporcionada por el distribuidor.....</i>	96
XI.7	<i>Dimensiones de la interfaz DisplayPort proporcionada por el distribuidor</i>	96
XI.8	<i>Dimensiones de la interfaz HDMI proporcionada por el distribuidor.</i>	97
XI.9	<i>Diseño de la placa electrónica de pruebas: vista de la parte inferior del modelo en tres dimensiones de la placa.</i>	97
XI.10	<i>Diseño de la placa electrónica de pruebas: vista de las capas exteriores, vías, conectores y serigrafía.</i>	98
XI.11	<i>Placa electrónica para pruebas: vista de la cara superior.</i>	98
XII.1	<i>Resultado experimental 01: análisis fcheck de la lectura de un pulso L1A.....</i>	99
XII.2	<i>Resultado experimental 02: análisis fcheck de la lectura de ocho pulsos L1A..</i>	99
XII.3	<i>Resultado experimental 03: análisis fcheck de la segunda lectura de ocho pulsos L1A.</i>	100
XII.4	<i>Resultado experimental: presencia de pulso L1A fuera de la frecuencia asignada (resaltado por óvalo).</i>	101
XII.5	<i>Resultado experimental 04: análisis fcheck de pulsos L1A a 60 Hz.....</i>	101
XII.6	<i>Resultado experimental 05: análisis de unboxing_v1_1.py de pulsos L1A a 60 Hz</i>	101
XII.7	<i>Resultado experimental 06: archivo .txt en formato de bloques generado por unboxing_v1_1.py”.....</i>	102
XII.8	<i>Resultado experimental 07 a: señales de las FIFO capturadas por ChipScope con muestras a 80 MHz.</i>	102
XII.9	<i>Resultado experimental 07 b: señales de las FIFO capturadas por ChipScope con muestras a 80 MHz.....</i>	103
XII.10	<i>Trama de datos número 1. Imagen obtenida del terminal cuando se ejecuta la función “fcheck -F 1” mientras está en ejecución el programa setup.sh.....</i>	104
XII.11	<i>Trama de datos número 2. Imagen obtenida del terminal cuando se ejecuta la función “fcheck -F 3” mientras está en ejecución el programa setup.sh.....</i>	105
XII.12	<i>Captura del eproc8 de la prueba T_01_03 con muestras de 25 ns.....</i>	107
XII.13	<i>Captura del eproc2 de la prueba T_01_03 con muestras de 12,5 ns).....</i>	107
XII.14	<i>Monitoreo por ILA de las tramas de datos transferidas para su verificación.....</i>	111
XII.15	<i>Formas de onda por ILA capturadas: comparación para demostrar los desfases que ocurren en el envío de la trama de datos del la XC7K325T a la XC7VX485T.</i>	111
XII.16	<i>Formas de onda por ILA capturadas: conjunto de los canales con el correcto funcionamiento del control de la FIFO.....</i>	112

XII.17	<i>Formas de onda por ILA capturadas: trama de datos enviada por el canal 1 de la XC7K325T, procesada por la XC7VX485T y reenviada a la XC7K325T.</i>	112
XII.18	<i>Formas de onda por ILA capturadas: uso de cable SCSI sin protección frente a EMC.</i>	113
XII.19	<i>Formas de onda por ILA capturadas: uso de cable SCSI con protección frente a EMC</i>	114
XII.20	<i>Formas de onda por ILA capturadas: primera trama recuperada por FIFO.</i>	115
XII.21	<i>Formas de onda por ILA capturadas: tramas recuperadas de FIFO</i>	116
XII.22	<i>Formas de onda por ILA capturadas: conteo final de tramas recuperadas de FIFO</i>	117
XIII.1	<i>Vista del TilePPr desarrollado por el IFIC-Valencia y de la tarjeta electrónica principal que envía las señales del QSFP1 y del integrador.....</i>	119
XIII.2	<i>Vista de las conexiones por fibra al TilePPR.....</i>	120
XIII.3	<i>Gráfica de valores que se obtienen del módulo ADC conectado al canal 9 (lado B) de la placa electrónica principal</i>	122
XIII.4	<i>Lecturas individuales del lado A y el lado B del integrador enviadas por el sistema FELIX.....</i>	123
XIII.5	<i>Gráfica de valores que se obtienen del módulo ADC conectado al canal 9 (lado B) a través de FELIX.....</i>	123
XIII.6	<i>Bloque IP creado en este trabajo: counter_test_ip_01</i>	124
XIV.1	<i>Referencia documental para la puesta en marcha del sistema FELIX en el IFIC-Valencia.....</i>	125
XIV.2	<i>Captura con las instrucciones de las palabras clave que controlan los estados del bloque IP emptynot_FIFO.</i>	126

Lista de Tablas

IV.1	<i>Puerto de entrada del integrator_receiver.vhd que contiene información del lado A</i>	29
IV.2	<i>Puertos de salida del integrator_receiver.vhd para el lado A de la placa electrónica principal</i>	30
VI.1	<i>Configuración del e-link en la programación de la FPGA del VC707</i>	38
VI.2	<i>Nueva configuración de los botones de pulso para la tarjeta de evaluación VC707</i>	39
VI.3	<i>Trama de datos enviada por el TilePPr a través del SFP Tx sin modificación de los RunParameters</i>	41
VI.4	<i>Trama de datos enviada por el TilePPr a través del SFP Tx con modificación de los RunParameters</i>	42
VI.5	<i>Trama de datos enviada por el TilePPr a través del SFP Tx sin modificación de los RunParameters y con los doce canales</i>	44
VI.6	<i>Trama de datos enviada por el TilePPr a través del SFP Tx con modificación de los RunParameters y con los doce canales</i>	45
VI.7	<i>Configuración del E-link en el firmware de la FPGA del servidor local</i>	46
VI.8	<i>Trama de datos para ser enviada por el TilePPr a través del SFP Tx: Único e-proc de 2 bits</i>	53
VI.9	<i>Trama de datos para ser enviada por el TilePPr a través del SFP Tx: Dos e-procs de 2 bits</i>	54
VI.10	<i>Trama de datos para ser enviada entre la VC707 y la KC705 a través del conector FMC</i>	63
VI.11	<i>Especificaciones LVDS para los bloques HP de la placa electrónica VC707</i> ...	68
VI.12	<i>Especificaciones LVDS_25 para los bloques HR de la placa electrónica KC705</i>	68

VII.1	<i>Ejemplos de tiempos UI (intervalo de unidad) para velocidades de transferencia de datos.....</i>	72
IX.1	<i>Especificaciones de relojes globales y PLL/MMCM en FPGA serie 7 de Xilinx.</i>	81
IX.2	<i>Agrupación por bloques I/O de pines de FPGA para la interfaz SCSI: Rx del VC707</i>	84
IX.3	<i>Agrupación por bloques I/O de pines de FPGA para la interfaz SCSI: Tx del VC707</i>	85
IX.4	<i>Asignación de canales por SCSI para la placa electrónica VC707</i>	85
IX.5	<i>Agrupación por bloques I/O de pines de FPGA para la interfaz HDMI: Rx del VC707.....</i>	86
IX.6	<i>Agrupación por bloques I/O de pines de FPGA para la interfaz HDMI: Tx del VC707</i>	86
IX.7	<i>Asignación de canales por HDMI para la placa electrónica VC707</i>	86
IX.8	<i>Agrupación por bloques I/O de pines de FPGA para la interfaz DisplayPort: Rx del VC707.....</i>	87
IX.9	<i>Agrupación por bloques I/O de pines de FPGA para la interfaz DisplayPort: Tx del VC707.....</i>	87
IX.10	<i>Asignación de canales por DisplayPort para la placa electrónica VC707</i>	87
IX.11	<i>Conectores asignados para uso de señales diferenciales de reloj.....</i>	88
IX.12	<i>Beneficios en la configuración de memoria según la FIFO.....</i>	89
IX.13	<i>Configuraciones que se pueden utilizar para las FIFO con dominios de reloj independientes.</i>	89
XII.1	<i>Condiciones de prueba iniciales para el L1A y el INT2HOST</i>	106
XII.2	<i>Resultados de prueba iniciales para el L1A y el INT2HOST.....</i>	107
XII.3	<i>Resultados comprobados con el fcheck (servidor local):</i>	108
XII.4	<i>Condiciones para la variación de la frecuencia de L1A</i>	108
XII.5	<i>Resultados de la variación de la frecuencia de L1A</i>	109
XII.6	<i>Tasa de error binario según las pruebas experimentales</i>	110
XIII.1	<i>Condiciones para la variación de la frecuencia de L1A en pruebas de certificación</i>	120
XIII.2	<i>Resultados de la variación de la frecuencia de L1A en pruebas de validación</i>	121
XIII.3	<i>Tasa de error binario según las pruebas de validación.....</i>	122
XV.1	<i>Horas de trabajo</i>	127

XV.2	<i>Coste de los componentes para la placa electrónica de prueba del FE-I4.....</i>	<i>128</i>
XV.3	<i>Coste estimado para el equipamiento básico del laboratorio TileCal</i>	<i>128</i>

I. OBJETIVOS Y RESUMEN DEL PROYECTO

I. 1. Objetivos

Los objetivos principales del proyecto son:

- a) Poner en marcha en la FPGA XC7VX485T del Pre-procesador o TilePPr la interfaz de transferencia de datos para que sea compatible con el nuevo sistema “Enlace de intercambio en el límite frontal” o FELIX (Front- End Link eXchange).
- b) Brindar las herramientas necesarias (programación hardware y aplicaciones software) para la verificación de la comunicación integral del sistema FELIX antes y después de agregar las funcionalidades del integrador.
- c) Programar en la FPGA XC7VX485T del TilePPr un sistema de transferencia de datos provenientes del integrador del Calorímetro de Baldosas Hadrónico TileCal (Hadronic Tile Calorimeter) que se reenviarán a través del sistema FELIX.
- d) Diseñar, ensamblar y programar los componentes de un sistema de transferencia de datos serial para evaluar la conexión entre la FPGA XC7K325T del FE-I4 a la FPGA XC7VX485T del TilePPr mediante la interfaz FMC de forma que sea compatible con el sistema FELIX.

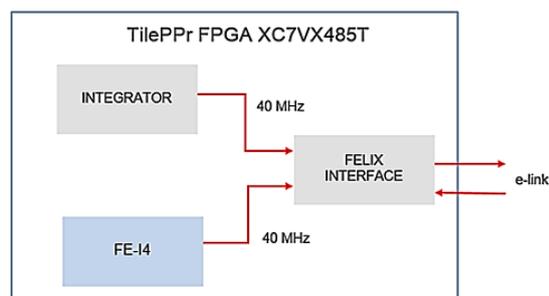


Fig. I.1 - Módulos de programación hardware objetivos en este trabajo.

I. 2. Resumen

El proyecto surge de la implementación del nuevo sistema de lectura de datos FELIX, que aprovecha los Transceptores Gigabit incorporados en las FPGA XC7VX485T y XC7VX690T del fabricante Xilinx. Estos transceptores se usan para establecer enlaces de datos a través de fibra óptica entre la FPGA del Pre-procesador del TileCal o TilePPr (XC7VX485T) y la placa electrónica que está conectada por PCIe al servidor local (XC7VX690T).

La placa electrónica con la FPGA que recibe la información del TilePPr y que va conectada por la interfaz PCIe al servidor local se le llama en este trabajo "TileDAQ", es programada directamente con el archivo .bit proveído por el CERN, siendo esta el núcleo de FELIX. El procedimiento de verificación que se lleva a cabo deja como última opción que la programación hardware de la FPGA del servidor local o "TileDAQ" sea responsable de cualquier error en la transferencia de datos. Nuevas versiones del archivo .bit son proveídas por el CERN periódicamente, siempre respetando las especificaciones compartidas a los distintos equipos de investigación que cooperan para el desarrollo del proyecto TileCal.

El sistema de pruebas que posee el laboratorio TileCal del Instituto de Física Corpuscular IFIC para la transferencia de datos entre el TilePPr y el TileDAQ se resume en la siguiente figura:

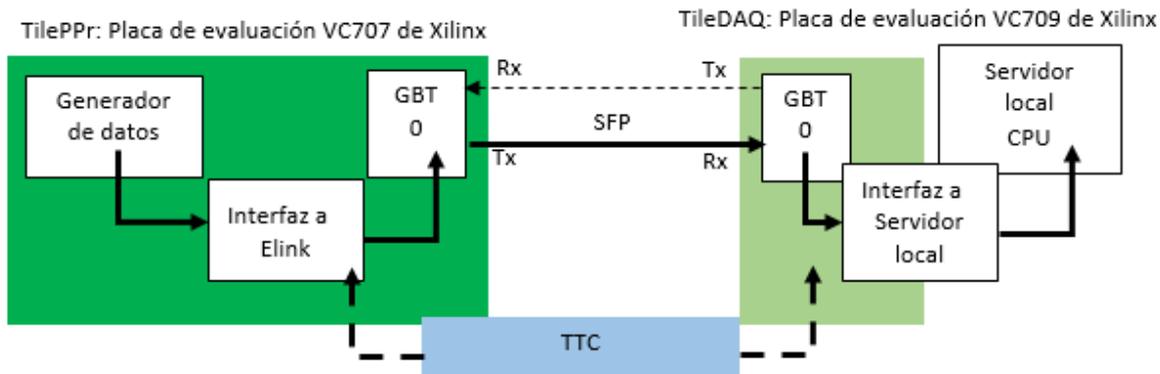


Fig. I.2 - Arquitectura del sistema de pruebas de FELIX en el laboratorio TileCal del IFIC-Valencia. La arquitectura real es más compleja, aquí se muestran los principales equipos para el desarrollo de este proyecto.

Las pruebas se llevan a cabo cuando se realizan cambios en código vhdl del TilePPr: modificaciones en la trama de datos a enviar, cambio de señales de reloj que controlan cada proceso, habilitación de interfaces para pruebas controladas manualmente, monitoreo de señales por ChipScope, etc. Modificaciones y debugging en la programación hardware del TilePPr tienen lugar después de comparar los resultados experimentales con las referencias documentales, comparar las señales obtenidas del Chipscope o ILA respecto a lo estimado, contraste entre el código original y las nuevas versiones, todo esto asegurando compatibilidad de las nuevas funciones con las anteriores que desempeñaba el TilePPr.

Respecto al lado del servidor local, el análisis por cada bloque de datos puede hacerse para muestras con duración configurable y resolución de un segundo, excelente para un análisis fino pero aún está lejos de las condiciones de operación normal del FELIX. Los *bugs*¹ que puedan encontrarse durante tiempos de muestreo mayores y con tramas de datos que son generadas más de cien mil veces por segundo, solamente pueden ser detectados con la ayuda de software.

Al comienzo de este trabajo, los datos enviados desde la FPGA XC7VX485T del TilePPr y los almacenados por el servidor local presentaban diferencias y algunos errores que podían ser identificados solamente estudiando la trama de datos. Se necesitaba realizar pruebas que fueran descartando cada una de las posibles causas de los errores y así subsanar los defectos del sistema.

Una vez que la interfaz del TilePPr con el FELIX se ha configurado correctamente se ha creado un punto de referencia para agregar la nueva programación hardware. Se programa en la FPGA un módulo que al final podrá ser instanciado fácilmente por nuevos diseñadores y que su interior, aunque puede ser modificado, no requiera demasiadas actualizaciones para las distintas fases del proyecto TileCal. Este módulo hardware puede ser conectado con señales con los demás ya existentes que procesan la información recibida de las interfaces QSFP y FMC para su reenvío por interfaz SFP a la TileDAQ.

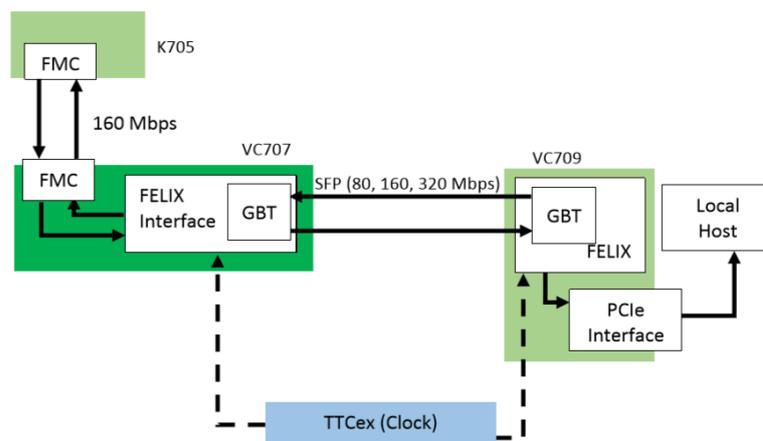


Fig. 1.3 - Arquitectura del sistema de pruebas de FE-I4 propuesto. La incorporación del TTCex sería en otra etapa del proyecto.

El contenido de este documento está dividido en capítulos que tratan aspectos específicos sobre el desarrollo de este trabajo:

II. Especificaciones de diseño: se mencionan las especificaciones técnicas y funcionales de los distintos componentes y equipos que fueron utilizados en este trabajo, así como de los nuevos productos que surgen del mismo.

III. Introducción: describe brevemente la manera en que este trabajo encaja, como si fuera el engranaje de una compleja maquinaria, en uno de los proyectos

¹ Un *bug*, cuya traducción literal al español es bicho, se utiliza comúnmente para nombrar los errores que se producen en un programa.

de ingeniería que se realiza en el laboratorio TileCal del IFIC-Valencia, parte colaboradora con los experimentos que tienen lugar en el LHC del CERN.

IV. Soluciones existentes: ubica con un enfoque aún más técnico en dónde tiene lugar este trabajo dentro de la fase de actualización del TilePPr al principio de este trabajo.

V. Estudio previo de las soluciones posibles: establece, considerando las soluciones existentes, los pasos a seguir para alcanzar los objetivos planteados en el capítulo “I. Objetivos y resumen del proyecto” respetando lo planteado en “II. Especificaciones de diseño”.

VI. Solución propuesta y especificaciones finales: expone brevemente las soluciones de programación hardware desarrolladas para alcanzar los objetivos planteados en capítulos anteriores.

VII. Normativa: brinda una vista superficial de los aspectos que más relevancia se tuvieron de la norma ANSI/VITA 57.1-2008 para el desarrollo de la placa electrónica que servirá para los experimentos y pruebas de la interfaz del FE-I4.

VIII. Simulaciones: presenta los resultados que se obtuvieron de simulaciones en las que se llevan a cabo las soluciones propuestas, esto con ayuda de software especializado en programación hardware de FPGA y en integridad de la comunicación.

IX. Elección de componentes y motivación: describe de forma particular el por qué y cómo se aprovecha el hardware adicional para llevar a cabo las soluciones propuestas, desde los conectores seriales hasta el hardware integrado que se instancia en la programación de la FPGA.

X. Interconexión y software: lista de manera concisa las interfaces que ya están operativas para la comunicación de las tarjetas electrónicas de prueba entre ellas mismas y con los demás equipos como computadoras, servidores, TTC, etc. También lista el software que aprovecha y hace posible esta comunicación, el software usado para el diseño y simulaciones y el software adicional creado para este trabajo.

XI. Diseño de la placa de pruebas: basado en “VII. Normativa” y ubicando su posición como un nuevo componente de “X. Interconexión y software”, sintetiza las reglas más importantes y las consideraciones de diseño que llevaron a la aceptación por el fabricante aprobado por el laboratorio IFIC-Valencia para la elaboración de la placa electrónica necesaria para la interfaz del FE-I4 en la VC707 y KC705.

XII. Pruebas y resultados experimentales: muestra el comportamiento real en distintas configuraciones de las soluciones propuestas: placa electrónica de prueba, programación hardware y software. También explica algunos comportamientos indeseados y los mecanismos tomados para su eliminación.

XIII. Pruebas de validación y certificación: pruebas realizadas con el equipo real del proyecto TileCal como la placa electrónica principal y los DAQ del integrador. Presenta evidencia del comportamiento real que tendrán las soluciones propuestas con su correcta utilización.

XIV. Manual de usuario: habla sobre el manual de usuario del que dispone la parte troncal de la interfaz del FELIX en la TilePPr.

XV. Costes de fabricación: cuantificación en moneda de la inversión que conlleva el desarrollo de este trabajo.

XVI. Conclusiones y trabajos futuros: presenta afirmaciones que surgen del desarrollo y culminación de este trabajo; también a las labores a realizar para dar continuidad a lo logrado con este proyecto.

XVII. Referencias bibliográficas: recursos documentales citados y sobre los que se apoyan numerosos conceptos presentados.

XVIII. Planos y Anexos: esquema de la placa electrónica de pruebas e información que complementa a los temas tratados.

XIX. Lista Acrónimos y Siglas: acrónimos y siglas de uso común en este documento.

II. ESPECIFICACIONES DE DISEÑO

Toda la programación hardware y placas electrónicas que se utilicen con la TilePPr y la TileDAQ deben ser compatibles y como mínimo satisfacer las especificaciones de las placas de evaluación VC707 y VC709 de Xilinx, respectivamente. La placa electrónica del TilePPr es diseñada en el IFIC-Valencia, TileDAQ es diseñada por otros grupos de investigación del CERN, así como las especificaciones del sistema FELIX y de las propuestas para el avance a la segunda fase, por lo que los requisitos pueden variar según las recomendaciones de estos grupos de investigación.

II. 1. Equipo disponible para pruebas del sistema FELIX en el IFIC

El equipamiento en el laboratorio TileCal del IFIC emula el área de trabajo del TileCal en el CERN, particularmente lo destinado para el TilePPr y el sistema FELIX. A continuación se muestran algunas de las especificaciones que debe reunir el equipo:

- a) Instalaciones y condiciones ambientales:
 - Alimentación eléctrica compatible con 110-240 Vac, 50-60 Hz con ± 1 Hz cada uno.
 - Máxima corriente de alimentación 16 A.
 - Clavija de enchufe tipo F (adaptadores que cumplan con los requisitos de arriba).
 - Condiciones ambientales de operación (alcance máximo esperado):
 - Temperatura ambiente: 22 ± 8 °C.
 - Humedad relativa: 20 % - 80 % HR (sin condensación).
 - Presión atmosférica: 1013 ± 10 kPa.
- b) Computadora de escritorio para conectarse con el servidor local:
 - CPU:
 - Procesador 3.6 GHz Intel Xeon.
 - Memoria RAM 8 GB (por separado).

- Almacenamiento en disco 250 GB.
 - Puertos USB 2.0.
 - Teclado estándar (español) QUERTY.
 - Ratón óptico USB con rueda de desplazamiento.
 - Monitor LED 20".
- c) Computadora portátil para programación hardware del VC707:
- Tamaño: 14,1".
 - Procesador 2.4 GHz x64 Intel Core i7.
 - Memoria RAM 8 GB (por separado).
 - Almacenamiento en disco 250 GB.
 - Puertos USB 2.0.
 - Teclado estándar (español) QUERTY.
- d) Placas electrónicas de evaluación VC707 y VC709 de Xilinx [1] [2]:
Son las mismas especificaciones de las tarjetas VC707 y VC709 de Xilinx, pero cabe mencionar las características que tomaron mayor importancia para el desarrollo de este trabajo:
- FPGA del TilePPr:
 - Virtex-7 XC7VX485T-2FFG1761C para el TilePPr.
 - Bloques BRAM Max (1kb): 37080.
 - Bloques BRAM 36 Kb: 1030.
 - Máximo de IO configurables por el usuario: 700.
 - Celdas lógicas: 485760.
 - Slices CLBs: 75900.
 - Máximo RAM distribuida: 8175.
 - Slices DSP: 2800.
 - GTX: 56.
 - GTH: 0.
 - FPGA del TileDAQ:
 - Virtex-7 XC7VX690T-2FFG1761C para la TileDAQ.
 - Bloques BRAM Max (1kb): 52920.
 - Máximo RAM distribuida: 108300.
 - Máximo de IO configurables por el usuario: 900.
 - Celdas lógicas: 693120.
 - Slices CLBs: 108300.
 - Slices DSP: 3600.
 - Bloques BRAM 36 Kb: 1470.
 - GTX: 0.
 - GTH: 80.
 - PCIe.
 - Oscilador diferencial integrado.
 - Conectores SMA (diferencial).
 - Conectores SMA para reloj por transceiver GT.
 - Memoria 1GB DDR3 SODIM, hasta 800MHz / 1600Mbps.
 - PCI Express de 8 líneas compatible con PCI Express 3.0 (TileDAQ).
 - Plataforma de conexión para alto ancho de banda y aplicaciones de alto desempeño de 40Gb/s.
 - Conectores SFP+.

- Compatible con desarrollo de aplicaciones Ethernet 10-100-1000 Mbps y sus respectivos puertos Ethernet RJ-45.
- Conectores con especificación FMC HPC VITA 57.1.
- Conector mini-B USB.
- GPIO: pushbuttons, dip switches, conectores SMA GPIO.
- Switch de encendido y apagado.

e) Placa electrónica de evaluación KC705 [3].

Esta placa toma relevancia para el apartado del FE-14:

- FPGA:
 - Kintex 7 XC7K325T-2FFG900C.
 - Máximo de IO configurables por el usuario: 500.
- Conectores con especificación FMC HPC VITA 57.1
- Conector mini-B USB.
- GPIO: pushbuttons, dip switches.
- Switch de encendido y apagado.

f) Generador de reloj con limpiador de jitter [4]:

- Máxima frecuencia de reloj aplicada a la referencia en modo fan-out: 1500 MHz.
- Máxima Frecuencia de reloj aplicada al divisor de referencia 250 MHz.
- Ciclos de trabajos en la referencia: 40 % - 60 %.

g) Osciloscopio digital de señal combinada [5]:

- Precisión de base de tiempo ± 5 ppm sobre cualquier intervalo superior a 1 ms.
- Especificaciones de los canales del sistema analógico vertical.
 - Entrada de cuatro canales.
 - Ancho de banda analógico (-3 dB) en el rango de 5 mV/div – 1V/div: 500 MHz.
 - Acoplamiento de entrada AC y DC.
 - Impedancia de entrada: $1M\Omega \pm 1\%$ (AC), $50\Omega \pm 1\%$ (DC).
 - Sensibilidad de entrada $1M\Omega$: 1 mV/div to 10 V/div.
 - Sensibilidad de entrada 50Ω : 1 mV/div to 1 V/div.
 - Resolución vertical: 8 bits.

II. 2. Especificaciones de la programación hardware del TilePPr para el sistema FELIX

- Soportar trabajo intensivo en modo GBT mientras mantiene la configuración en la ejecución de anchuras específicas de e-link.
- Configuraciones e-link estáticas (definidas ya en la síntesis).
- Configuración estática para obtener el máximo número posible de canales.
- Compatibilidad completa con configuraciones e-link.
- Trabajar con los dominios de reloj de 80 MHz, 160 MHz y 320 MHz obtenidos de la señal de 40 MHz proveída por el TTCex.
- Para las tramas provenientes del QSFP1, uso de e-links de 8 bits.

II. 2. 1 *E-link*

- Codificación (TilePPr) y decodificación (TileDAQ) 8b/10b.
- Codificación y decodificación HDLC para cualquier e-link (servidor local).
- TTC al límite frontal (cualquier e-link puede ser un TTC e-link).
- Paquete de información de L1A hacia los puntos finales de la red.
- Disponibilidad para paquetes fijos o de longitud variable (no requerido para la fase I).

II. 2. 2 *Especificaciones del módulo hardware del integrador*

- Transferencia de datos al TileDAQ simultánea con versiones anteriores y experimentales del módulo HDLC receptor del integrador.
- Independencia del L1A y transferencia ininterrumpida de datos.
- Trama de datos que contenga como mínimo la siguiente información: lado del integrador de donde provienen, contador de trama, índice de datos del receptor del integrador y datos adquiridos.
- Tramas enviadas a través del FELIX por e-links de 2 bits.

II. 2. 3 *Especificaciones del módulo hardware del FE-I4*

- Respetar las configuraciones requeridas de e-link (trama de datos de 8 bits, síncrono con señal de reloj de 40 MHz).
- Independencia del L1A y de los otros módulos de datos.
- Trama de datos que pueda adaptarse fácilmente a otros protocolos de comunicación y que sea compatible con el FELIX.
- Compatible con versiones recientes de software para programación hardware.
- Recibir a través del conector FMC HPC VITA 57.1 datos serializados con una frecuencia de reloj de 160 MHz, desde la tarjeta de evaluación KC705. Este conector está acoplado directamente a los pines I/O de las FPGA, por lo que su uso está limitado por la velocidad máxima de las interfaces, relojes globales y aquellos generados por las FPGA de la serie 7 de Xilinx.

III. INTRODUCCIÓN

III. 1. CERN e IFIC-Valencia

El Consejo Europeo de Investigación Nuclear CERN (Conseil Européen por la Recherche Nucléaire) fue establecido como una organización de clase mundial dedicada a la investigación de física fundamental. Su Gran Colisionador de Hadrones LHC (Large Hadron Collider), permite realizar experimentos que llevan a los protones e iones hacia colisiones con energía mayor a la de cualquier otra máquina de su tipo, aportando resultados a la línea de investigación de la física de altas energías en aceleradores.

En estos experimentos, es fundamental la adquisición de datos que brinda cada evento considerado estadísticamente como único. De esto se encargan las unidades detectoras dentro del LHC, cada una con objetivos diferentes, como el Aparato Toroidal del LHC ATLAS (A Toroidal LHC Apparatus) siendo éste uno de los cuatro experimentos más grandes en el LHC. Con la ayuda de ATLAS se busca mejorar nuestra comprensión de la constitución fundamental de la materia, respondiendo a preguntas asociadas con el modelo estándar, las partículas fundamentales y las fuerzas fundamentales que las rigen [6].

III. 1. 1 ATLAS y el modelo estándar

Aunque el modelo estándar es la mejor descripción del mundo subatómico que hay hasta ahora, hay preguntas que quedan por responder y teorías que poner a prueba. Un ejemplo son los experimentos del ATLAS que concluyeron haber observado una nueva partícula en la región de masa alrededor de 126 GeV (Giga electronvoltio), consistente con el bosón de Higgs, la manifestación más simple del mecanismo Brout-Englert-Higgs. El esfuerzo que exigen los experimentos en el ATLAS se traduce en la participación de miles de científicos de 182 instituciones ubicadas alrededor del mundo, entre ellas el Instituto de Física Corpuscular IFIC de Valencia en España [6].

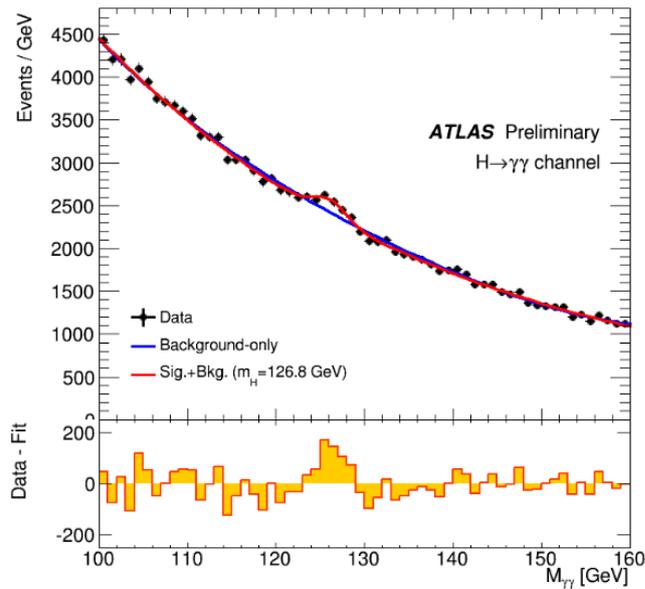


Fig. III.1 - Lecturas obtenidas del ATLAS que llevaron al anuncio de la observación de una nueva partícula en la región de masa de 126 GeV.

III. 1. 2 La interacción entre el CERN y el IFIC

Distintas unidades de investigación y desarrollo del IFIC participan en los experimentos llevados a cabo por el CERN en la línea de la física en altas energías en aceleradores. Esta participación está centrada en dos grandes proyectos: el LHC y el Compact Linear Collider (Colisionador Compacto Lineal CLIC). Respecto al LHC, el IFIC ha contribuido con la construcción de diversos sistemas del detector ATLAS, sistemas de computación y manejo de los datos proporcionados por el detector.



Fig. III.2 - Participación del IFIC-Valencia en el proyecto ATLAS.

III. 2. Instrumentos del experimento ATLAS

III. 2.1 TileCal: Hadronic Tile Calorimeter

El TileCal es el Calorímetro de Baldosas Hadrónico que cubre la región central del experimento ATLAS. Está compuesto por soportes de acero tipo placa y aproximadamente 460000 baldosas de escintilador² o centellador plástico, material que emite luz al recibir el impacto de una radiación ionizante o fotón. El TileCal está dividido longitudinalmente en cuatro secciones cilíndricas: los segmentos LBA, LBC que están ubicados al centro, EBA y EBC ubicados a los extremos. Cada uno de los segmentos está compuesto de 64 módulos de placas metálicas que portan a los escintiladores en células individuales. El grupo TileCal IFIC-Valencia ha construido 50 % de los módulos electrónicos del Calorímetro Hadrónico para uno de los cuatro segmentos o particiones del detector, también es responsable del diseño, ensamblaje y ensayo de las placas electrónicas de los drivers de lectura ROD (Read Out Driver) instalados en el TileCal de ATLAS [7].

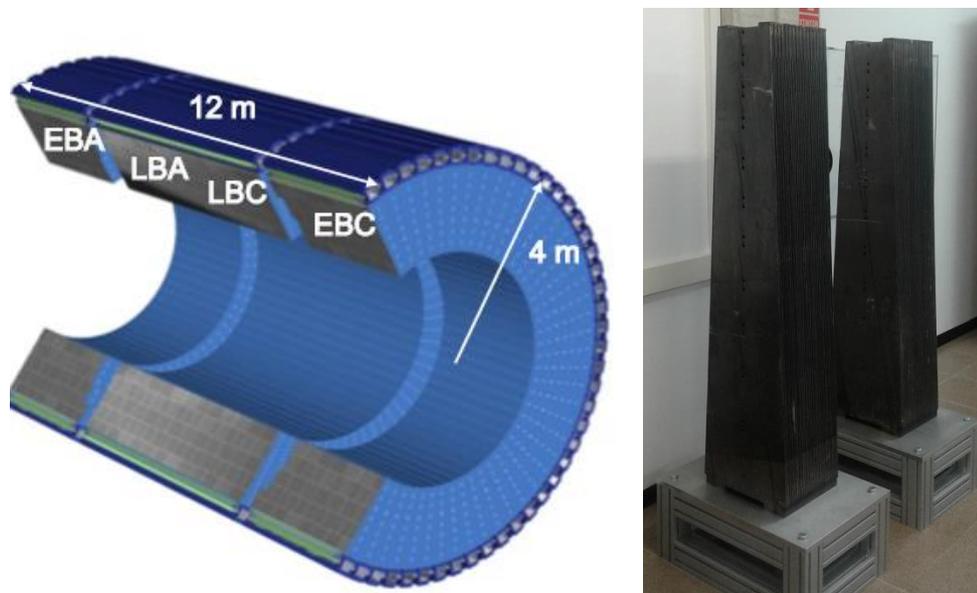


Fig. III.3 - Soportes de acero que conforman las secciones del TileCal. A la izquierda, disposición de las cuatro secciones dentro del calorímetro de hadrones. A la derecha, módulo de placas metálicas en el IFIC-Valencia que compone a cada una de las secciones.

En los experimentos, las partículas son detenidas gradualmente por las placas metálicas y se producen fotones solamente cuando las partículas que están cargadas atraviesan el material escintilador. La luz del lado de cada célula es colectada por fibras con longitudes de ondas cambiantes que son leídas por los tubos fotomultiplicadores PMT localizados en *super drawers*, *drawers* o cajones de la parte trasera de cada módulo, junto con la electrónica de lectura y de alta tensión. Son 9852 PMT necesarios para la lectura completa del TileCal [6].

² Tomado del diccionario usual de la Real Academia Española. La palabra original en inglés es *scintillator*. Llamarlo centellador es lo más habitual.

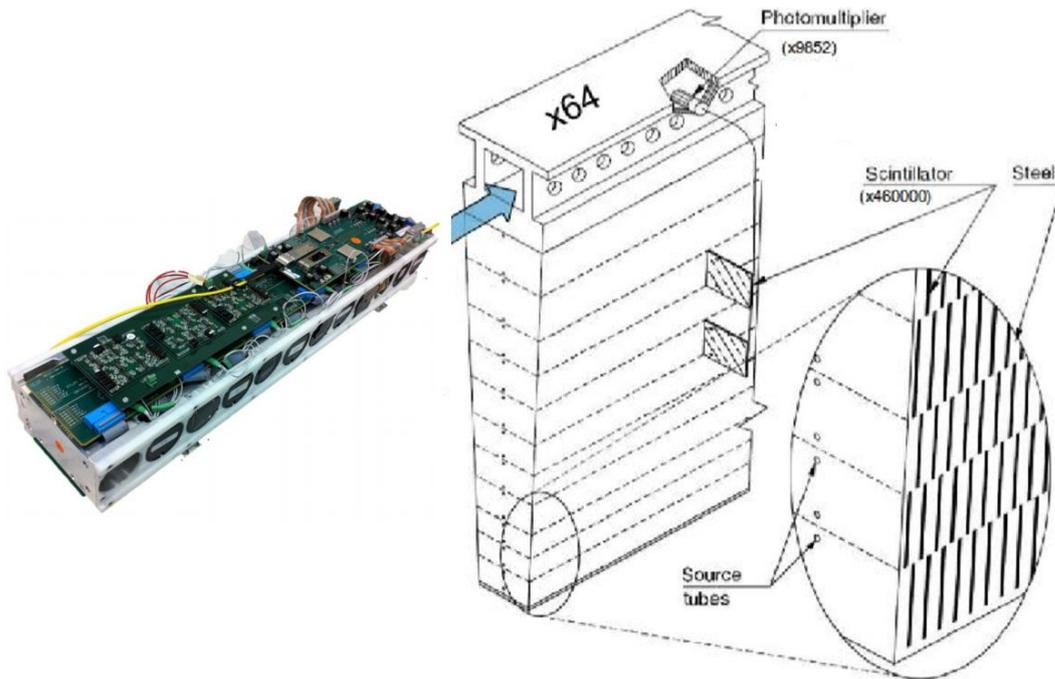


Fig. III.4 - Ubicación de la electrónica en los soportes de acero. A la izquierda, electrónica del TileCal a la que va conectada los PMT. A la derecha, detalle del módulo de placas metálicas.

La electrónica ubicada en estos cajones se encarga de adquirir, analizar, discriminar, procesar y enviar los datos que se obtienen de los PMT. A la fecha de elaboración de este documento, la electrónica se encuentra en un estado de transición hacia la “Fase 2 de actualización del Calorímetro de Baldosas de ATLAS” que incrementa por un factor de cinco la luminosidad instantánea, por lo que también se le llama actualización de Alta Luminosidad HL (High Luminosity). Esta actualización incluye múltiples niveles de redundancia, accesibilidad por el usuario mejorada, tolerancia a la radiación mejorada y mayor confiabilidad. La electrónica en los cajones consta de cinco partes principales:

- Tarjetas electrónicas del terminal frontal FE (Front End): conectadas directamente a los tubos de los fotomultiplicadores PMT.
- Placas electrónicas principales (Mainboards): digitalizan los datos provenientes de las tarjetas del terminal frontal.
- Placas electrónicas hijas (Daughterboards): sirven de interface entre la electrónica del terminal frontal y el terminal trasero BE (Back End), reciben las tramas de datos y contienen los enlaces de comunicación óptica de alta velocidad para la escritura de datos en el sistema de adquisición de datos.
- Sistema de control programable de alto voltaje HV.
- Fuente de potencia de bajo voltaje LVPS.

El control de tiempo y disparo TTC (time and trigger controller) provee a cada módulo con señales de reloj, señales de *trigger*³ L1A, control y capacidades de configuración a través de enlaces con fibra óptica.

³ Esta señal desencadena los eventos de mayor interés para este trabajo. Como su nombre indica, es un “desencadenador” de procesos en la programación hardware de la FPGA.

III. 2. 2 El TilePPr

El TilePreprocesador, TilePPr o simplemente PPr es una placa compleja con tres diferentes FPGA y es el principal de los componentes electrónicos en el BE, reemplazando al *Trigger Preprocessor* de la fase anterior. Puede controlar y leer cuatro mini-cajones, es decir, un completo súper cajón. Su funcionalidad incluye recibir y procesar datos digitales de las tarjetas electrónicas hijas, decodificando y distribuyendo señales del TTC a la electrónica en el terminal frontal para la configuración y sincronización con el reloj del LHC. Provee control y monitoreo de las fuentes de alto voltaje a través de un sistema de control de detector DCS. El sistema de HV y LVPS son controlados y monitorizados por el DCS [7].

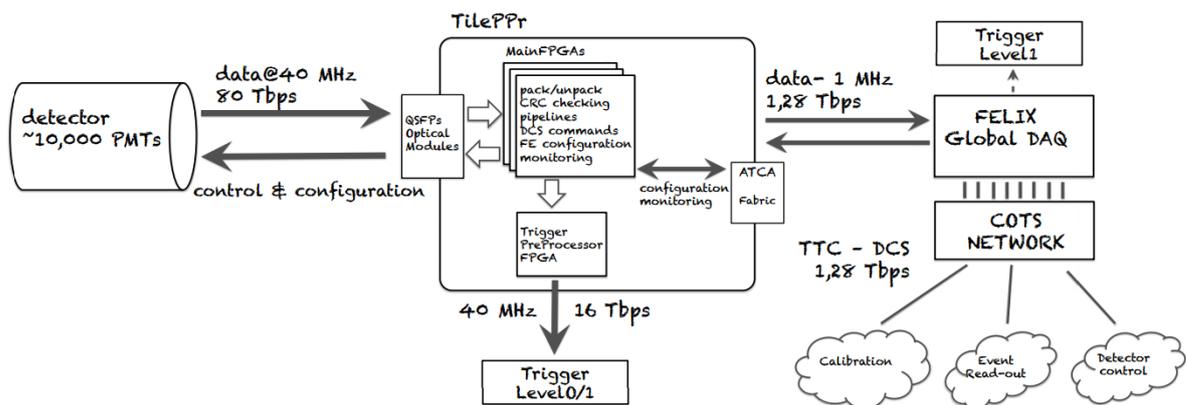


Fig. III.5 - Diagrama de la arquitectura de lectura de TileCal para la segunda fase de actualización de ATLAS.

En el sistema actual, las señales provenientes de los PMT son digitalizadas con un reloj de 40 MHz que es síncrono con el cruce de las partículas. Las señales análogas obtenidas de los PMT son simplificadas y adaptadas en tarjetas del FE, luego son enviadas a placas donde los pulsos son digitalizados y almacenados en *buffers pipeline* para esperar lectura cuando una señal activadora ocurre. Esta señal es el *trigger* de nivel uno L1A cuya latencia tiene una duración de 2,5 μ s. Simultáneamente las señales analógicas del PMT son agrupadas y transmitidas al sistema de nivel uno del calorímetro.

Con el FELIX, las lecturas electrónicas del TileCal introducirán una nueva estrategia de lectura donde la información digitalizada de cada cruce será leída por el TilePPr, que proveerá información al nuevo *trigger* y a los sistemas DAQ. La arquitectura propuesta y sus componentes prototipos son evaluados, como los módulos detectores demostradores (Demonstrators) que han sido expuestos a diferentes emisiones de partículas (electrones, muones y hadrones) y a diferentes energías para evaluar su desempeño.

III. 2. 3 Los demostradores

El desarrollo de los prototipos del nuevo sistema de lectura se lleva a cabo gracias al Proyecto Demostrador (Demonstrator Project) que facilita el desarrollo del nuevo sistema mientras está conectado con la infraestructura existente. La información de los eventos que ha sido seleccionada por el *trigger* de nivel uno es

leída mediante enlaces ópticos por placas de interface hacia los drivers de lectura ROD [8].

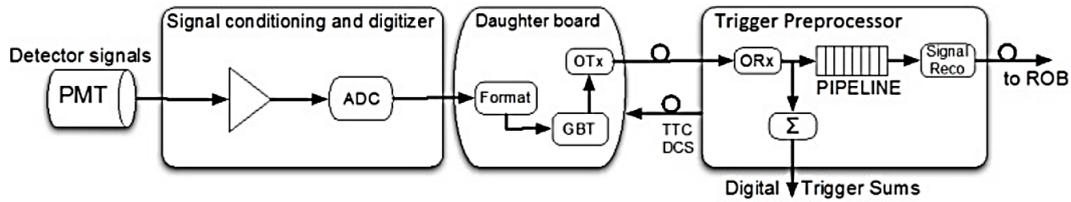


Fig. III.6 - Diagrama de bloques que muestra el flujo de la señal del sistema "Demonstrator".

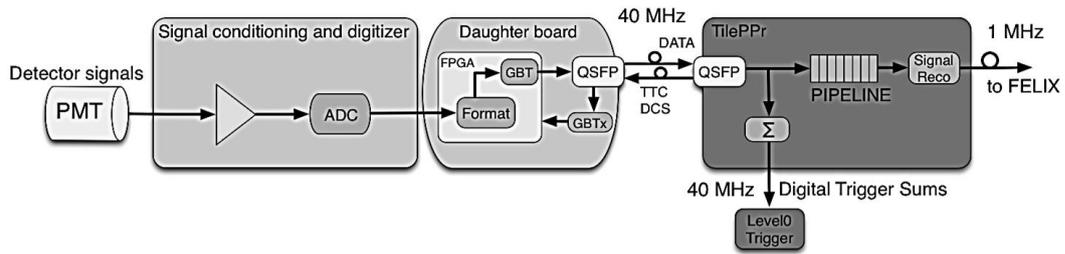


Fig. III.7 - Diagrama de bloques que muestra el flujo de la señal del sistema con la actualización de alta luminosidad.

III. 2. 4 Sistema de lectura FELIX

Durante la "Fase 2 de actualización del Calorímetro de Baldosas de ATLAS" se reemplaza el sistema actual por uno más modularizado. Toda la información de las células será leída continuamente, significando en la transición de los 256 enlaces ópticos (165 Gbps) a 8192 enlaces ópticos (80 Tbps) [8]. Los datos correspondientes a los eventos seleccionados por el trigger serán extraídos de las pipelines y transmitidos al FELIX. El software Athena es usado para la reconstrucción de datos que se realiza en línea durante la adquisición y los resultados son presentados para su verificación.

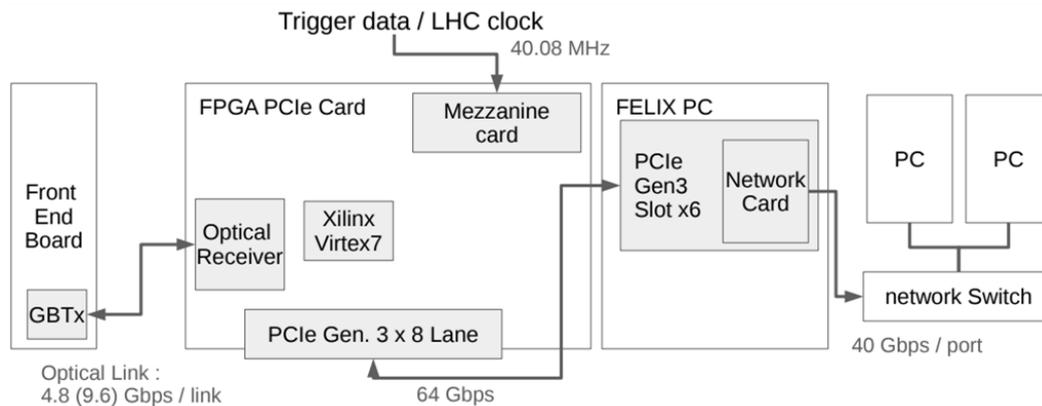


Fig. III.8 - Arquitectura del FELIX.

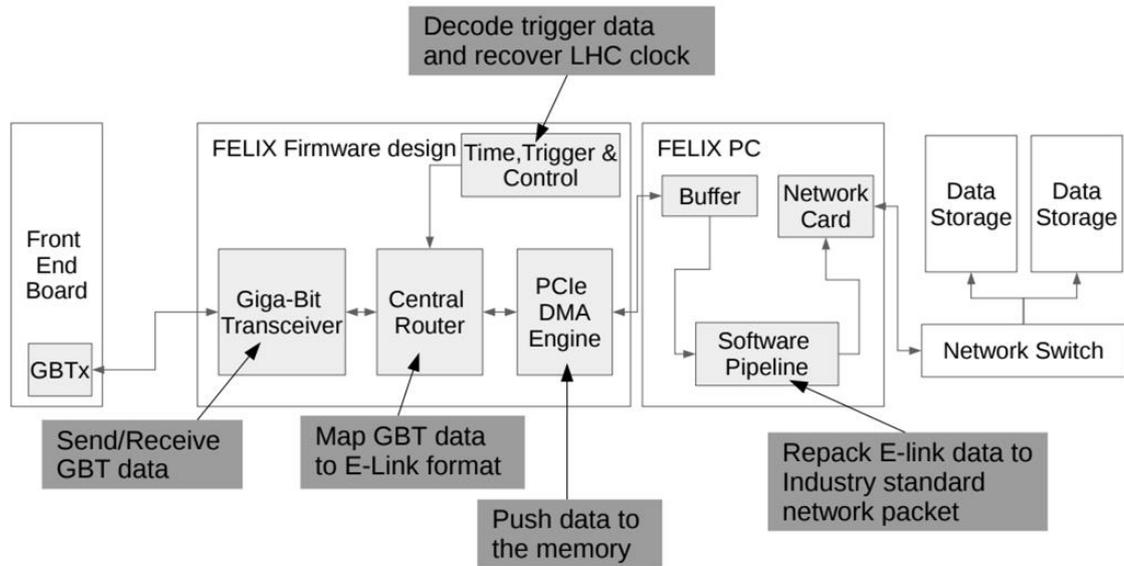


Fig. III.9 - Vista funcional del FELIX.

III. 2. 5 Sistema de lectura del integrador

La lectura del integrador es el núcleo del sistema del sesgo mínimo MB (minimun bias) que graba corrientes lentas para el calorímetro. Durante la adquisición de datos del LHC, una lectura basada en el integrador provee de señales que vienen de colisiones protón-protón inelásticas en transferencias de momento relativamente bajas y permite el monitoreo instantáneo de la luminosidad en ATLAS, así como la respuesta en las células del calorímetro. Cada vez que se realiza una prueba de colisión, esta es dividida en lumi-bloques LB, los cuales son intervalos con una duración fija donde la luminosidad instantánea es casi constante. Las corrientes del integrador son grabadas por cada unidad LB y son promediadas alrededor de 20-25 medidas de integrador por LB [9]. Las corrientes del integrador son grabadas por todos los canales del TileCal y han sido usadas para detectar y cuantificar el efecto de las radiaciones del escintilador con los datos tomados en 2012 y 2015.

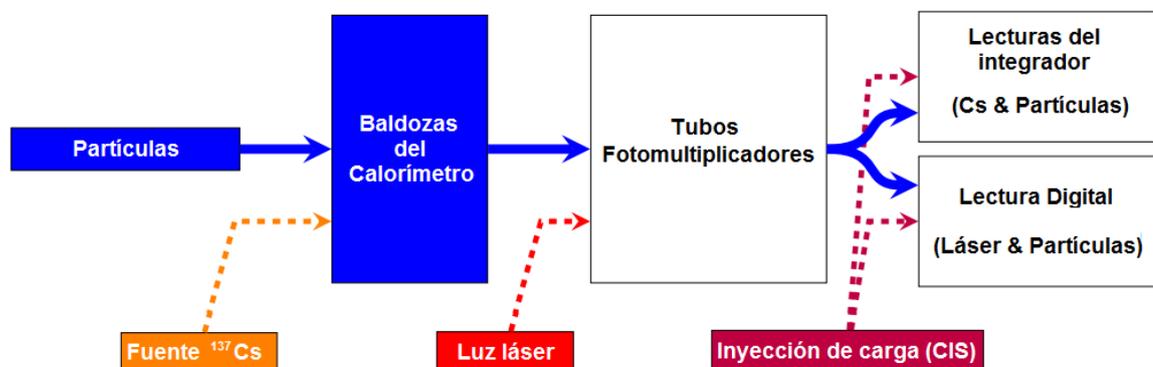


Fig. III.10 - Ruta de la señal del TileCal con sus diferentes sistemas de monitoreo.

Existe dependencia lineal entre las corrientes del integrador medida durante las colisiones y la luminosidad instantánea, de la cual surge el coeficiente de luminosidad (corriente/luminosidad instantánea). El coeficiente es constante a lo largo del tiempo y puede ser verificado dividiendo la corriente promedio sobre el promedio de la luminosidad instantánea.

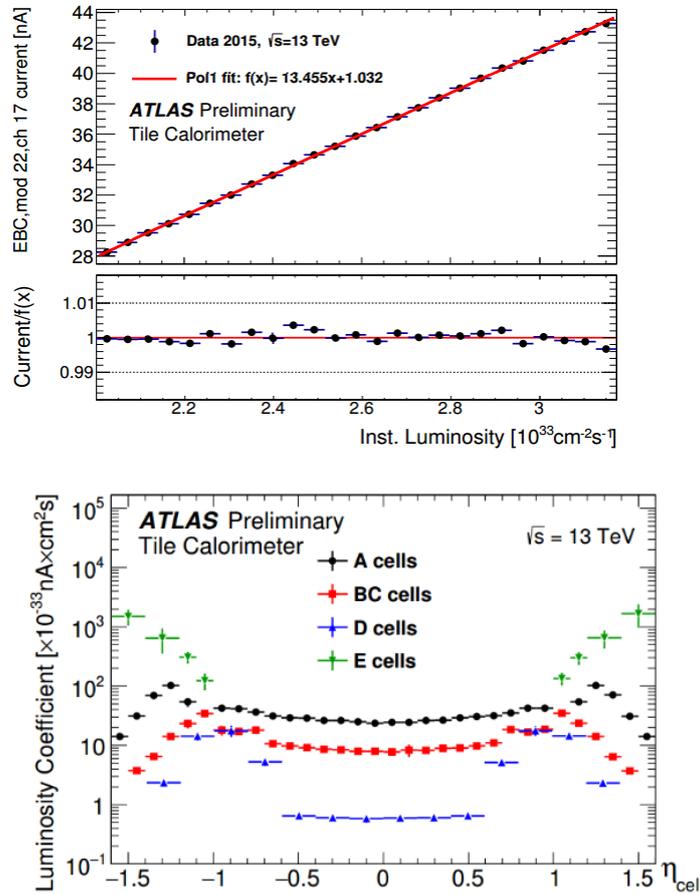


Fig. III.11 - Adquisición de datos 27262 en el año 2012. Arriba, corriente contra luminosidad instantánea, recta obtenida de la célula D5. Abajo, coeficiente de luminosidad η para todas las células TileCal.

Con la ayuda del sistema MB es posible identificar los efectos de la radiación en los escintiladores del TileCal. Un ejemplo es la cuantificación del efecto por irradiación que es a lo mucho de 2 % (es decir, una pérdida de 2 % en el rendimiento lumínico) en la célula más expuesta, según observaciones de los años 2012 a 2015.

III. 2. 6 Sistema de lectura del FE-I4

El FE-I4 es un circuito integrado diseñado para cubrir los requisitos de los experimentos de ATLAS. Es el circuito integrado más grande para las aplicaciones de la física de partículas, con un mejor rendimiento que la actual generación de los detectores de píxeles del LHC [10]. La utilidad de la arquitectura de este circuito integrado se ha evaluado con colisiones simuladas a una tasa de 40 MHz y con una variante cantidad de interacciones por colisión, abarcando las actualizaciones de HL del LHC.

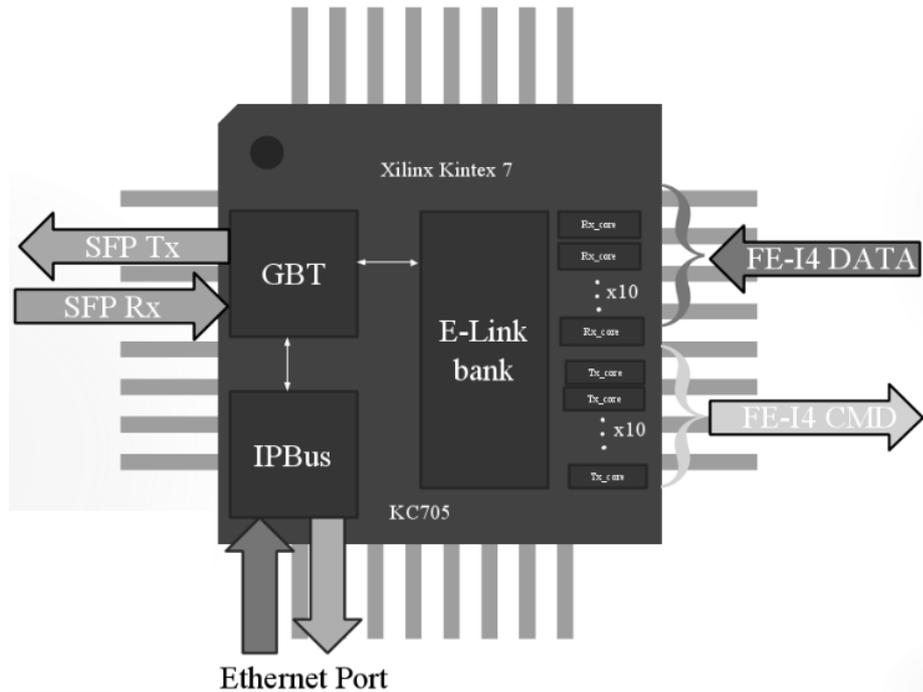


Fig. III.12 - Enlaces e interfaces programados en la FPGA de la placa de evaluación KC705 para el FE-I4.

Cada pixel tiene una etapa de amplificación independiente y de forma ajustable, seguida de un discriminador con un límite de disparo también ajustable. La información es sacada de forma serializada, codificada en 8b/10b. Los datos que recibe la KC705 proveniente de distintos FE-I4, cada uno por su canal individual, que al final serían retransmitidos por el TilePPr al FELIX.

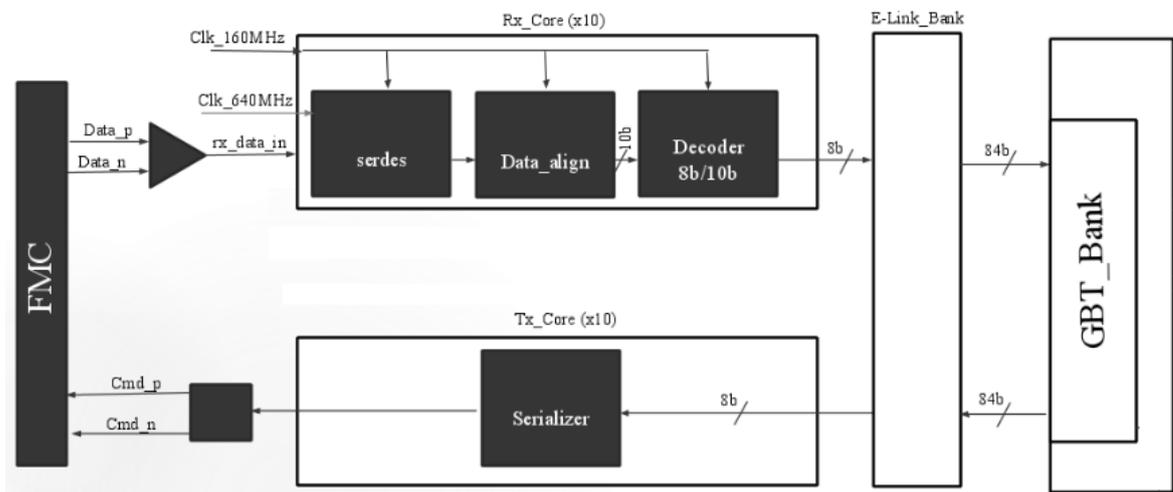


Fig. III.13 - Arquitectura que une el FE-I4 con el sistema FELIX a través del GBT programado en el TilePPr.

IV. SOLUCIONES EXISTENTES

El enlace de intercambio en el límite frontal FELIX es la interface entre el TilePPr y el sistema global de adquisición de datos de ATLAS. Es el núcleo de la nueva arquitectura de trigger y adquisición de datos de ATLAS: funciona como un enrutador, implementado por servidores PC brindando interfaces de red accesibles, placas electrónicas diseñadas para FPGA con interfaces y muchos transceptores de alta velocidad para usarse con fibra óptica.

Separa la información destinada al transporte de la destinada a la manipulación, así que puede ser procesada cómodamente por servidores conectados a la red, reemplazando los enlaces punto a punto entre los componentes en el límite frontal y el sistema DAQ con una red conmutable.

Up Link only	Present	Upgrade
Total Bandwidth	200 Gbps	80 Tbps
Number of fibers	256	8192
Fiber bandwidth	800 Mbps	9,6 Gbps
Number of modules	32	32
Number of crates	4 (VME)	4 (ATCA)
Input bandwidth per board	6,4 Gbps	2,5 Tbps
Out bandwidth to DAQ per module	3,2 Gbps	40 Gbps
Out bandwidth to trigger per module	Analog front-end	500 Gbps

Fig. IV.1 - Comparación entre el sistema de lectura de TileCal y la segunda fase de actualización.

IV. 1. FELIX en el TileDAQ: El enrutador central

El enrutador central del FELIX en el TileDAQ es encargado de enrutar y dar formato a la información de los e-links entre la interface del Transceptor Gigabit GBT y el motor de Acceso Directo a Memoria DMA conectándose a través del PCIe, interfaz que permite el acceso al servidor local.

Por el lado de la GBT se observa que el tránsito de información puede tomar dos direcciones y un proceso diferente para cada una: hacia el servidor local (to Host) y desde el servidor local (from Host). Fuera de los protocolos y señales que aseguran el enlace entre el TTC, servidor local, TileDAQ y TilePPr, la ruta desde el servidor local no ha sufrido mayores modificaciones durante el desarrollo de este trabajo.

La administración de datos consiste en módulos e-group, donde cada uno maneja 16 bits de una sola entrada. Por el lado del motor PCIe hay una FIFO con un puerto de 256 bits. Son cuatro las anchuras de los datos e-link que pueden tomar dentro de un e-group: 2, 4, 8 y 16 bits. Cada posible e-link es administrado por un proceso electrónico e-proc que escribe los datos recibidos a su FIFO dedicado de 2 KB. Es la configuración del GBT la que define cuáles e-links existen y por tanto cuáles e-procs están activos. Los e-procs cogen los datos por cada ciclo de reloj de 40 MHz.

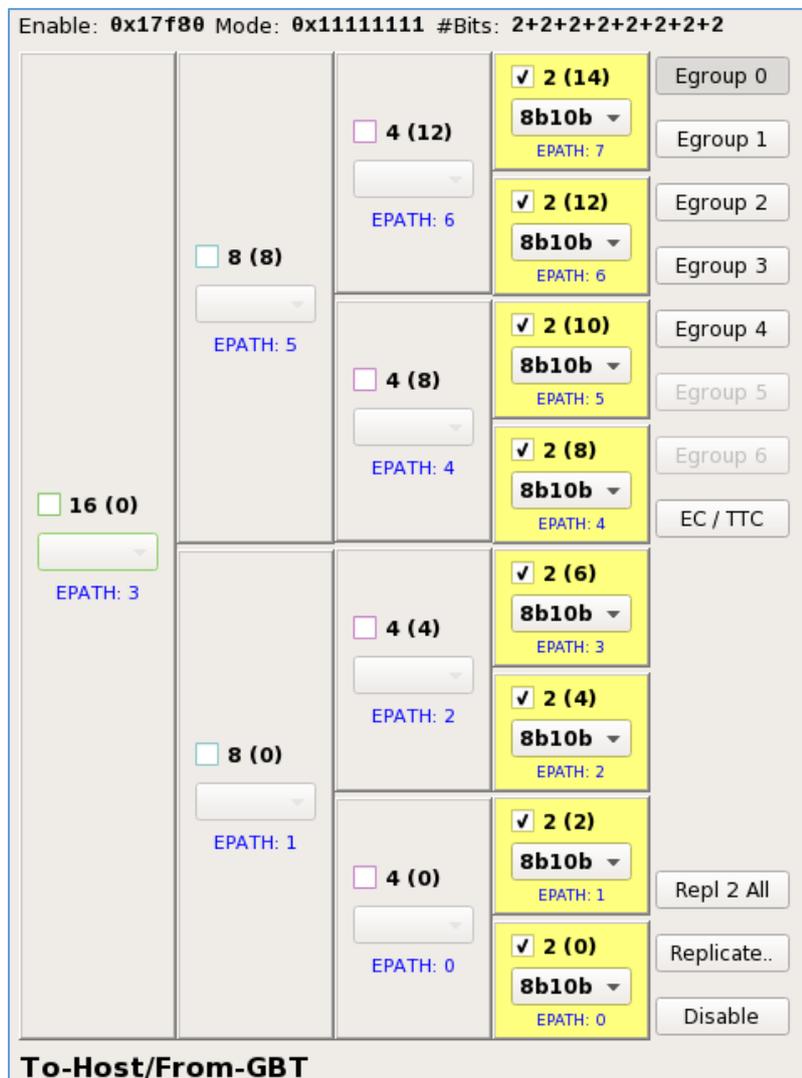


Fig. IV.2 - Captura tomada de la interfaz gráfica "FELIX E-LINK Configurator". Apartado de configuración de la ruta hacia el servidor en modo normal con 8 e-links de 2 bits en codificación 8b10b del e-group 0.

El administrador de datos consiste en módulos de agrupamiento o e-groups que manejan 16 bits de una entrada. La cantidad de e-groups del administrador de datos depende del modo de operación del GBT: en modo normal son 5 e-groups, en modo *wide* son 7 e-groups. Cada e-group puede procesar datos de entrada de hasta 8 e-links (si son de 2 bits).

Según el manual de referencia del FELIX para el manejo del TileDAQ, los datos de cada e-link se almacenan en 16 bits por la FIFO con anchura de 2 KB, acumulados en bloques de datos de 1 KB. Estos bloques de 1 KB de un solo canal GBT son escritos a un FIFO con anchura de 16 KB, 16 bits de entrada, 256 bits de salida. De esta forma, las FIFO de 16 KB, una por cada canal GBT, sirven para igualar la anchura de los datos y que puedan ser leídos por un simple algoritmo Round-Robin, escribiendo los datos a la FIFO de 4 KB de anchura que está a la salida del enrutador central y la cual es interfaz con el PCIe.

El archivo .bit proveído por el CERN para programar la FPGA del TileDAQ no se modifica, pero la configuración e-link puede ser manipulada para realizar las pruebas con el TilePPr. La función *elinkconfig* dentro de la aplicación *setup.sh* que es ejecutada desde el terminal de LINUX, hace posible lo anterior.

IV. 2. FELIX y el TilePPr: Señales de reloj

Las señales de reloj con las que funciona FELIX son todas provenientes del TTC⁴. Por los diferentes factores que puedan aumentar el jitter a lo largo del enlace óptico, la señal de 160 MHz es sometida a limpieza recibándose por la tarjeta TTC FMC-EDA 02319-V3, conectada al puerto FMC HPC1 de la VC707.



Fig. IV.3 - Fibras de conexión entre el TTCex y las placas electrónicas de prueba: puerto 1 hacia la TileDAQ y el puerto 4 hacia el TilePPr.

⁴ Eventualmente llamado TTCex para su identificación en el laboratorio TileCal del IFIC-Valencia.

Se identifica en la programación hardware de esta FPGA XC7VX485T:

- a) Archivo .ucf donde se declaran los constraints básicos y los puertos:
 - TTC_FMC_CLK160_IN_P (LOC = U39)
 - TTC_FMC_CLK160_IN_N (LOC = T39)
- b) La lógica programable de la FPGA conecta la señal de 160 MHz recibida con la señal CLK_TTC_FMC_160MHz dirigida hacia los puertos:
 - USR_SMA_P_OUT (LOC = AJ32)
 - USR_SMA_N_OUT (LOC = AK32)
- c) Ambos puertos SMA se conectan al generador de reloj y limpiador de jitter CDCE62005 de Texas Instruments, llevando de una señal de referencia de 160 MHz a una señal de 120 MHz.
- d) El puerto de salida del CDCE62005 se conecta a los puertos MGT CLK P (J25) y MGT CLK N (J26) de la tarjeta de evaluación VIRTEX VC707.
- e) Se declara en los constraints FELIX del firmware los puertos que reciben la señal de reloj 120 MHz:
 - SMA_SFP_CLK_P (LOC = AK8)
 - SMA_SFP_CLK_N (LOC = AK7)
- f) La señal de reloj de 120 MHz es llevada de los constraints a los bloques de mayor nivel y generadores de otras señales de reloj definidos en el firmware a través de las señales:
 - GTX_SMA_REF_CLOCK_120MHZ_P
 - GTX_SMA_REF_CLOCK_120MHZ_N

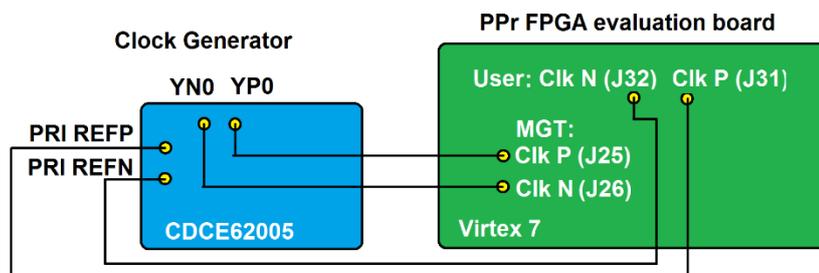


Fig. IV.4 - Diagrama de conexión entre el TilePPr y el generador de reloj en el sistema de pruebas de IFIC-Valencia.

IV. 3. FELIX y el TilePPr: La interfaz electrónica E-link

El e-link es una interfaz eléctrica concebida para la transmisión de datos sobre PCB o cables eléctricos dentro de una distancia de pocos metros. Los e-links pueden operar a cualquier velocidad hasta 320 Mbps, por lo que el módulo donde está programado incorpora transmisores y receptores para las siguientes velocidades y codificaciones:

- 80 Mbps para 10b8b, codificación HDLC o sin codificación.
- 160 Mbps para 10b8b o sin codificación.
- 320 Mbps para 10b8b o sin codificación.

Los módulos FIFO2Elink y Elink2FIFO programados en la FPGA del TilePPr se utilizan como interface para e-link separados físicamente o directamente a un marco GBT. Ambos módulos FIFO pueden servir de interface serial de e-link a 80, 160 o 320 Mbps y a marcos GBT de 2, 4 y 8 bits, este último a 40 MHz.

IV. 4. FELIX y el TilePPr: Vista superficial de su programación VHDL

La programación de cada FPGA se realiza en vhd y por la cantidad de módulos e IP cores utilizados esta es extensa. Aún en el (ya simplificado) esquema RTL del archivo *top* de la jerarquía vhd existen decenas de señales que comunican distintos módulos⁵ de programación vhd. Su tamaño y tiempo de sintetizado justifica el trabajar en la placa electrónica de evaluación en lugar de trabajar directamente en la TilePPr: los componentes absolutamente necesarios y solamente uno de los QSFP son programados en la placa electrónica VC707.

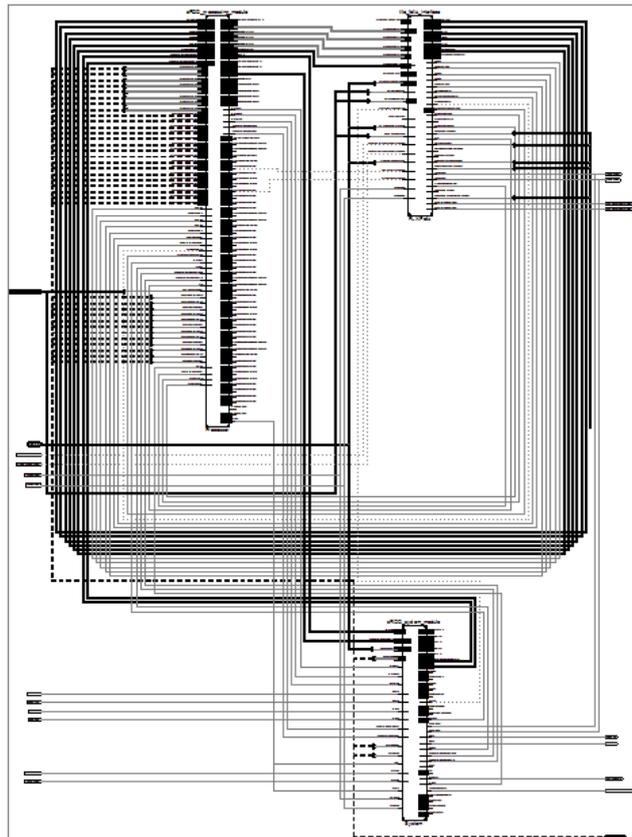


Fig. IV.5 - Esquema RTL: Vista de los tres módulos principales que componen a la interfaz FELIX en la FPGA del VC707. Solamente las señales y componentes más relevantes con el trabajo serán tratados en este documento.

⁵ El término módulo de programación se utiliza frecuentemente en este documento para referirse a algún bloque RTL o conjuntos de bloques RTL envueltos en un solo *top* que cumplan con una función específica.

Por el gran tamaño y cantidad de archivos de códigos que conforman la interfaz FELIX, la mejor forma de representar y comprender rápidamente la programación en la FPGA de la VC707 es con los bloques principales que surgen del esquema del “RTL View” (esquema a nivel de compuerta) o del “Technology View” (esquema de arquitectura específica) que se obtiene después del sintetizado.

IV. 4. 1 Processor (sROD_processing_module.vhd)

Debajo del *top*, este es el primer módulo que recibe las palabras de 16 bits provenientes de los QSFP.

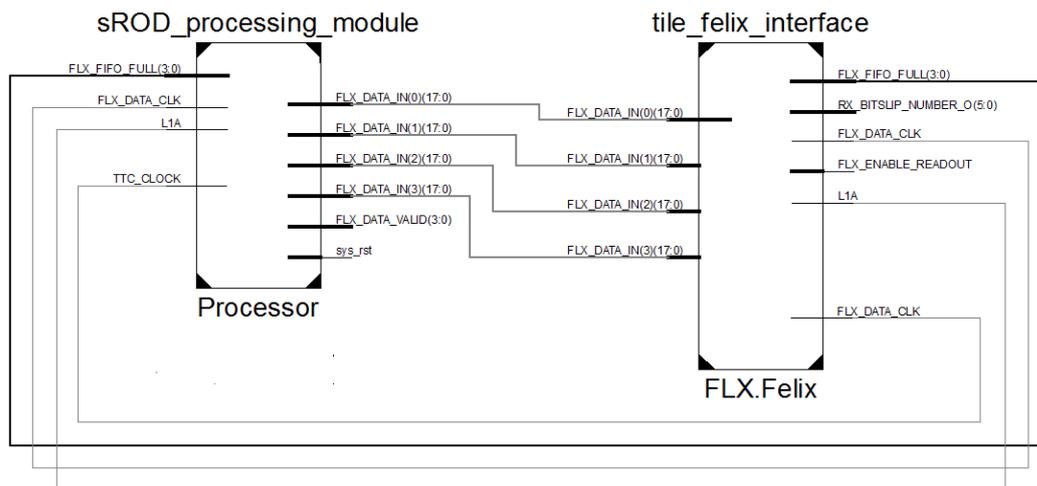


Fig. IV.6 - Esquema RTL: Relación de algunas señales entre los módulos Processor y FLX.Felix. Se han omitido puertos y señales para mayor claridad.

Los módulos y archivos más relevantes dentro del procesado del ROD para los fines de este trabajo son:

- *ttc_fmc_wrapper_xilinx.vhd*, que recibe el reloj de 40 MHz proveniente del TTC, el cual transporta también la señal L1A.
- *slaves.vhd*, que recibe los estados de conexión de los “mini drawers” (mini cajones) y las lecturas del integrador. De este módulo sale la configuración de los registros, algunas señales de control de los registros y algunos comandos hacia la placa electrónica hija.
- *readout_top_module*, que recibe los registros pipeline, FLX_DATA_CLK, TTC_CLOCK, el FLX_FIFO_FULL (3:0), la configuración de la lectura, el BCID (11:0). La señal de salida de mayor importancia es el FLX_DATA_IN, registro compuesto de 4 vectores con 18 bits de longitud cada uno.
- *integrator_receiver.vhd*, es instanciado uno para cada uno de los mini cajones. La información que recibe este módulo procede del módulo decodificador del GBT o *MD1_GBT_decoder (GBT_tile_decoder.vhd)*. Ya decodificada la información sale del *MD1_GBT_decoder* por las

señales *QSFP1_In_Integrator_dataA*, *QSFP1_In_Integrator_dataB*, datos provenientes de un solo mini cajón. Las señales luego son procesadas por el *integrator_receiver.vhd* y reenviadas al módulo *slaves* (*slaves.vhd*), ahora se identifican como las señales *QSFP1_IntegratorDatatoIPbusA* y *QSFP1_IntegratorDatatoIPbusB*.

IV. 4. 2 FLX.Felix (*tile_felix_interface.vhd*)

El módulo más importante para los fines de este trabajo: procesa la mayoría de las señales entre el *SROD_processing_module* y el *sROD_system_module*, utiliza los IP core *FIFO2Elink* y el *xlx_k7v7_gbt_example_design.vhd*. Entre sus señales de salida llevan al conector SFP Tx y SFP Rx que le permite comunicarse con la FPGA del TileDAQ. En el *FLX.interface* se instancian los nuevos módulos elaborados en este trabajo. Dentro de la interfaz FELIX del TilePPr está:

- *mmcmfelixclocks* (*clk_felix_MMCM.vhd*), genera hacia su salida las señales de reloj de 80 MHz, 160 MHz y 320 MHz con una entrada de 40 MHz en la frecuencia de la señal de reloj proveniente del *gbtExmpIDsgn*.
- *gbtExmpIDsgn* (*xlx_k7v7_gbt_example_design.vhd*), módulo donde se instancia el Transceptor Gigabit y todos los demás parámetros para su funcionamiento según el protocolo acordado para FELIX. Cuenta con su respectivo control *bitslip*, ajustes por desfases, etc. Programado por otros equipos de trabajo del CERN, solamente se instancian de él y hacia él las señales necesarias para su funcionamiento. Se cuentan con distintas versiones para poder ser utilizado en ISE o en Vivado.
- *elink_tx* (*FIFO2Elink.vhd*), instancia al empaquetado mediante los e-procs, a los e-procs en sí y al IP core de FIFO dedicado de 2 KB como lo hace el TileDAQ.

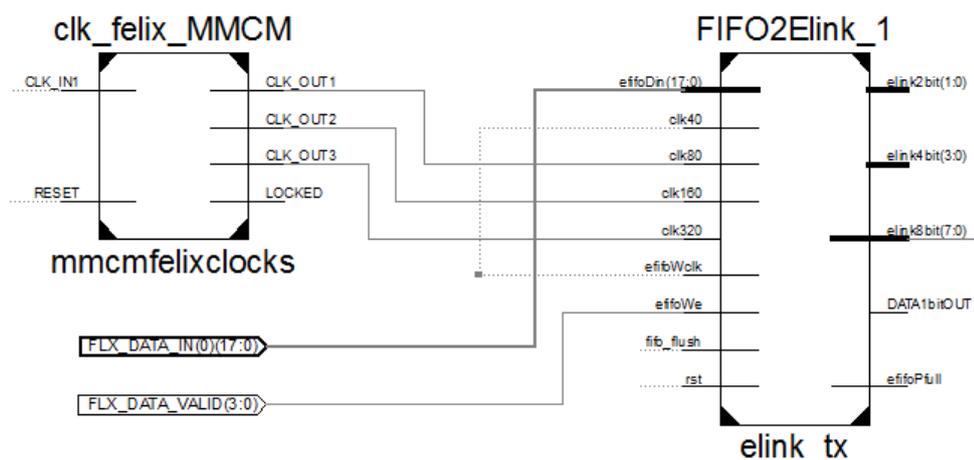


Fig. IV.7 - Esquema RTL: Relación entre el módulo *elink_tx* y el *mmcmfelixclocks* dentro del módulo *FLX.Felix*. Se han omitido puertos y señales para mayor claridad.

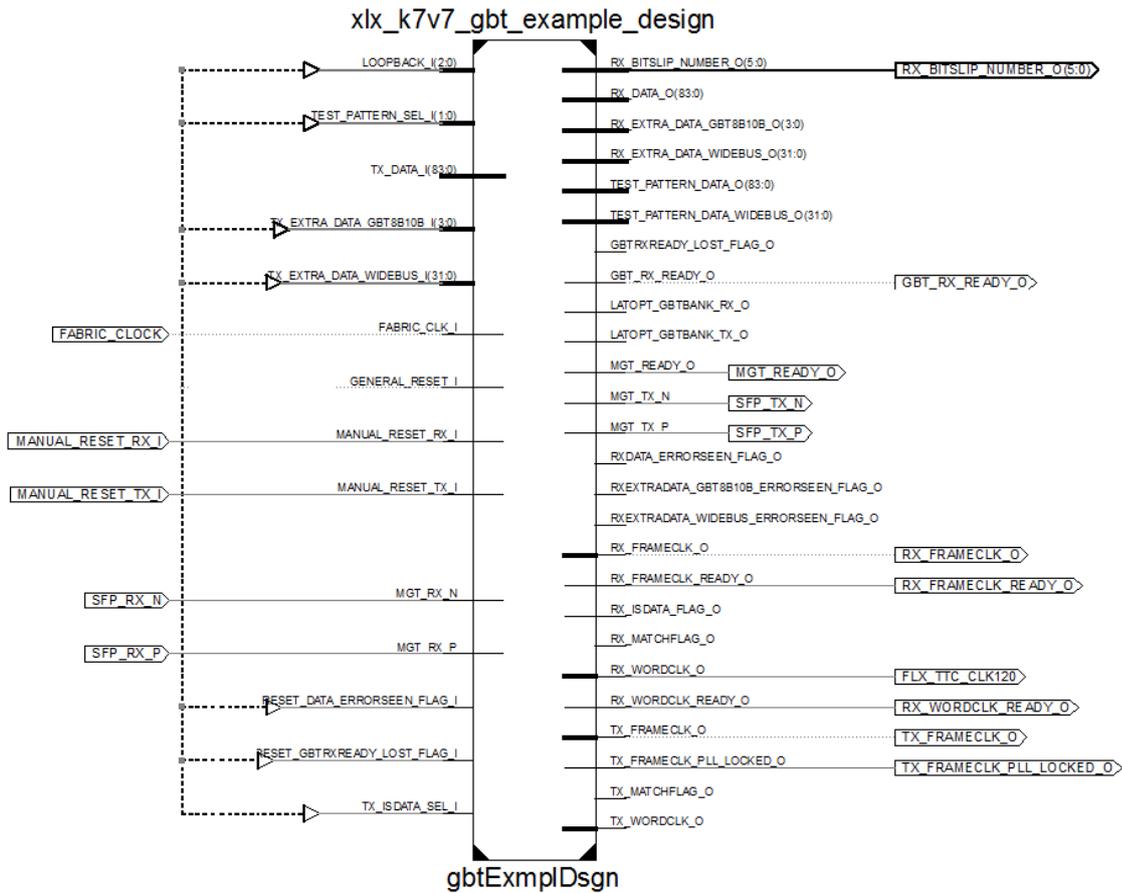


Fig. IV.8 - Esquema RTL: Vista top del gbtExmplDsgn instanciado dentro del módulo FLX.Felix. Se han omitido puertos y señales para mayor claridad.

IV. 4. 3 System (sROD_system_module.vhd)

El módulo está destinado para usarlo con el sistema IPbus, Ethernet y otros generadores de señales de reloj. Para los fines de este trabajo su aporte es mínimo.

IV. 5. El módulo integrator_receiver en el TilePPr

El módulo *integrator_receiver* (*integrator_receiver.vhd*) del FELIX es instanciado dentro del módulo *Processor* (*sROD_processing_module.vhd*) para cada uno de los mini cajones. Por cada mini cajón se instancia un módulo *integrator_receiver* cuyos puertos son mapeados con las señales del código:

```
-- Signals mapped inside the Processor module for the integrator.vhd:

--in          reset                => sys_rsti,
--in          GBT_rx_clk            => QSFP1_GBT_rx_clkA,  --(80MHz)
--in          IntegratorData1      => QSFP1_In_Integrator_dataA,
--in          IntegratorData2      => QSFP1_In_Integrator_dataB,
--out         IntegratorOutDataA   => QSFP1_IntegratorDatatoIPbusA,
--out         Integ_weA            => QSFP1_Integ_weA,
--out         Integ_indexA        => QSFP1_Integ_indexA,
--out         IntegratorOutDataB   => QSFP1_IntegratorDatatoIPbusB,
--out         Integ_weB            => QSFP1_Integ_weB,
--out         Integ_indexB        => QSFP1_Integ_indexB,
--out         Integ_clk            => QSFP1_Integ_clk
```

Las señales de entrada son vectores de 5 bits de anchura, provenientes de los lados A y B de cada placa electrónica principal. El bit más significativo sirve para indicar la validez de los datos precedentes, cuando su valor es de '1' el *integrator_receiver* pone en marcha su máquina de estados llenando el registro de salida con los cuatro bits menos significativos.

TABLA IV.1 Puerto de entrada del *integrator_receiver.vhd* que contiene información del lado A

Posición en el puerto IntegratorData1:	IntegratorData1 (4)	IntegratorData1 (3 down to 0)
Uso:	Bit de validación	Datos enviados

En cada ciclo de 80 MHz, los tres bits menos significativos proporcionan los datos obtenidos de la lectura de cada canal y el índice de lectura del integrador *IndexA* (número de muestra del canal según el *integrator_receiver*). Este índice se compara con el índice de lectura del TilePPr, el registro de seis vectores con anchura de cuatro bits *ReadoutIndexA* (número de muestra del canal), para verificar que entre los procesos a 80 MHz del *integrator_receiver* y los procesos a 40 MHz del *FLX.felix* no existe pérdida de información.

La salida del módulo del integrador para cada lado de la placa electrónica principal es un registro de seis vectores con 32 bits de anchura. Cada vez que un vector se llena, su respectiva señal de salida *Integ_weA* o *Integ_weB* cambia de '0' a '1'. Cuando el *Integ_weA* o *Integ_weB* del sexto vector toma un valor de '1', el registro se considera lleno y se pueden extraer los datos de él.

TABLA IV.2 Puertos de salida del integrator_receiver.vhd para el lado A de la placa electrónica principal

Registros a la salida del integrator_receiver.vhd									
IndexA	Integ_weA	IntegratorOutDataA							
(3 downto 0)	(0)	(31 downto 16)	(15 downto 12)	(11 downto 8)	(7 downto 4)	(3 downto 0)			
IntegratorData1 (3 downto 0)	'X'	ReadoutIndexA	IntegratorData1 (3 downto 0)						
IntegratorData1 (3 downto 0)	'X'	ReadoutIndexA	IntegratorData1 (3 downto 0)						
IntegratorData1 (3 downto 0)	'X'	ReadoutIndexA	IntegratorData1 (3 downto 0)						
IntegratorData1 (3 downto 0)	'X'	ReadoutIndexA	IntegratorData1 (3 downto 0)						
IntegratorData1 (3 downto 0)	'X'	ReadoutIndexA	IntegratorData1 (3 downto 0)						
IntegratorData1 (3 downto 0)	'X'	ReadoutIndexA	IntegratorData1 (3 downto 0)						
IntegratorData1 (3 downto 0)	'X'	ReadoutIndexA	IntegratorData1 (3 downto 0)						
Ejemplo de los valores que se pueden obtener de los registros a la salida del integrator_receiver.vhd									
IndexA	Integ_weA	IntegratorOutDataA							
(3 downto 0)	(0)	(31 downto 16)	(15 downto 12)	(11 downto 8)	(7 downto 4)	(3 downto 0)			
"0000"	'1'	x"0064"	x"7"	x"0"	x"0"	x"1"	x"1"		
"0010"	'1'	x"0064"	x"6"	x"f"	x"5"	x"4"	x"4"		
"0011"	'1'	x"0064"	x"7"	x"0"	x"0"	x"0"	x"0"		
"0100"	'1'	x"0064"	x"7"	x"0"	x"0"	x"0"	x"0"		
"0101"	'1'	x"0064"	x"6"	x"f"	x"5"	x"4"	x"4"		
"0111"	'1'	x"0064"	x"7"	x"0"	x"0"	x"0"	x"0"		
"1000"	'1'	x"0064"	x"7"	x"0"	x"0"	x"0"	x"0"		

IV. 6. Propuesta del FE-I4 y el sistema FELIX

El nuevo sistema Felix permite incluir una mayor cantidad de señales y transmisión de datos como los del FE-I4. Una de las propuestas consiste en enviar a través de distintos canales seriales información al TileDAQ con ayuda del conector FMC.

La propuesta ya existe, así como la arquitectura que se pretende emplear. Lo que hace falta es evaluar experimentalmente la opción de utilizar cableado de fácil acceso en el comercio (SCSI, HDMI, DisplayPort). Aunque no se utilice la normativa aplicada con las señales de alta velocidad de estos estándares como el TMDS33 para el HDMI, estos cables servirían únicamente como línea de transmisión.

El emulador del FE-I4 en VIVADO es parte de la transición a las nuevas herramientas en la programación FPGA, así como una gran cantidad de módulos que se han programado en el CERN y son utilizados en el proyecto TileCal. Siendo esto solo una propuesta, es prácticamente un sistema que se desarrollará desde cero dentro de la TilePPr.

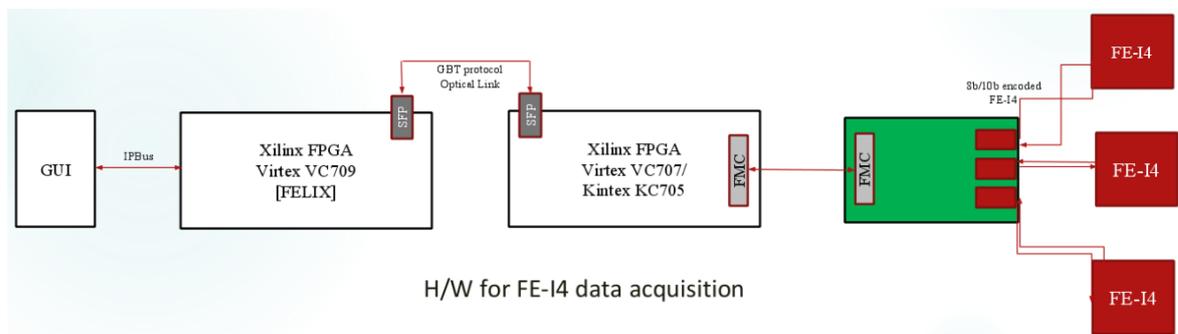


Fig. IV.9 - Propuesta para la adquisición de datos del FE-I4.

V. ESTUDIO PREVIO DE LAS SOLUCIONES POSIBLES

Las soluciones se han dividido para alcanzar los objetivos presentados en las primeras páginas de este documento.

V. 1. Puesta en marcha del FLX.Felix

Se puede establecer comunicación entre las dos FPGA del TilePPr y del TileDAQ, aunque se presentan algunos *bugs*. Antes de agregar cualquier otro módulo, el sistema FELIX debe estar completamente funcional, con la ayuda de un plan de verificación y ajustes en la FPGA del TilePPr:

- a) Programación hardware adicional para el control manual y automático de la señal trigger L1A.
- b) Verificación del funcionamiento correcto en el lado del TilePPr con el Chipscope.
- c) Determinar la configuración correcta de e-link en el lado del servidor local con el programa *setup.sh*; guardar del archivo binario que servirá de referencia.
- d) Documentación del procedimiento específico para la puesta en marcha del sistema FELIX en el laboratorio de TileCal en IFIC.
- e) Verificación en la adquisición de datos (al menos a nivel de bloque) con diferentes programaciones hardware, distintas frecuencias de disparo de L1A y distintas configuraciones e-link.

El software *setup.sh* proporcionado por el CERN se usará para verificar la comunicación exitosa, así se deja un archivo *.bit* funcional y verificado antes de agregar la funcionalidad del *integrator_receiver.vhd* compatible con FELIX.

V. 2. Software para la detección de errores más específicos

El CERN proporciona un software para verificar fallos a nivel de bloques que genera el GBT hacia el PCIe. Este software puede inicializarse a través de terminal y ejecutarse a través de comandos. La verificación a nivel de bloque es insuficiente para asegurar la integridad de la comunicación. Para mejorar esto:

- a) Crear un programa basado en python verificando que entregue los mismos resultados que *setup.sh* utilizando el archivo de referencia.
- b) Agregar funcionalidades de análisis de bloques, fragmentos y canales al programa basado en Python.
- c) Agregar una interfaz gráfica que permita hacer fácil e incluso intuitivo realizar los análisis.
- d) Verificar el programa de análisis con otros archivos binarios que sirvan de referencia.

V. 3. Programación hardware del módulo que comunique el *integrator_receiver.vhd* al servidor local

Con la interfaz FELIX operativa en el TilePPr y con un software de verificación hecho en el laboratorio, se procede de la siguiente manera:

- a) Revisión documental y del código de programación hardware para caracterizar el funcionamiento del *integrator_receiver.vhd*.
- b) Identificar las señales, lo más cercanas posible a los puertos del *top*, que sirven de entrada al módulo *integrator_receiver.vhd* y las señales que se necesitarán de salida.
- c) Propuesta de la trama de datos que servirá en los resultados experimentales, basada en el funcionamiento del actual archivo *integrator_receiver.vhd*.
- d) Importación de una copia del módulo *integrator_receiver.vhd* a un nuevo proyecto ISE.
- e) Programación de los archivos *.vhd* e instanciaciones.
- f) Programación de un archivo *vhd* que “envuelva” a todo lo programado previamente.
- g) Importación del módulo envuelto al proyecto ISE del FELIX. Ajuste de las señales en la programación hardware, ajustes en el *debugger .cdc* de ChipScope y verificación de la ausencia de errores desde la síntesis hasta la obtención del archivo *.bit*.

- h) Programación del TilePPr, programación de la FPGA que se conecta al servidor local, configuración de los e-links y adquisición de datos.
- i) Verificación del TilePPr en el Chipscope. Verificar la obtención correcta de datos en el servidor local con el programa *setup.sh* y el programa basado en Python.
- j) Ajustes para utilizar la que será el formato final de la trama de datos.
- k) Documentación de lo programado y del procedimiento de verificación.

V. 4. Programación hardware del sistema de comunicación entre el FE-I4 y el servidor local

En principio, todas las tarjetas electrónicas están sincronizadas con la señal de reloj de 40 MHz, aunque para hacer frente a la posibilidad de cualquier desfase o diferencia se contempla que esta parte del proyecto pueda responder a dominios de reloj diferentes. Para llevarla a cabo se contempla:

- a) Asignar los pines de la FPGA que operarán con la interfaz FMC de las placas de evaluación VC707 y KC705.
- b) Diseñar la PCB que se conectará entre la interfaz FMC y la interfaz serial (HDMI, SCSI o DisplayPort), cumpliendo las normas pertinentes.
- c) Estimación de costos y solicitud de componentes.
- d) Soldadura y ensamblaje de los conectores a la PCB.
- e) Programación hardware de ambas FPGA, que incluya un módulo generador de tramas de ejemplo que simulen menos de doce canales independientes.
- f) Monitorero mediante ILA de la transferencia de las tramas de datos generados con un reloj de 20 MHz en la KC705 (independiente de los relojes de la VC707) y recuperados con una señal de reloj de 40 MHz en la XC7VX485T.

VI. SOLUCIÓN PROPUESTA Y ESPECIFICACIONES FINALES

VI. 1. Puesta en marcha del FLX.Felix

VI. 1. 1 Programación hardware adicional y verificación con Chipscope

Se identifica dentro del código vhd la configuración e-link con la que se programa la FPGA del TilePPr, esto se hace en la configuración de las señales de entrada del *elink_tx* (*FIFO2Elink.vhd*) y del *gbtExmplDsgn* (*xlx_k7v7_gbt_example_design.vhd*) que son mapeados dentro del *FLX.Felix* (*tile_felix_interface.vhd*).

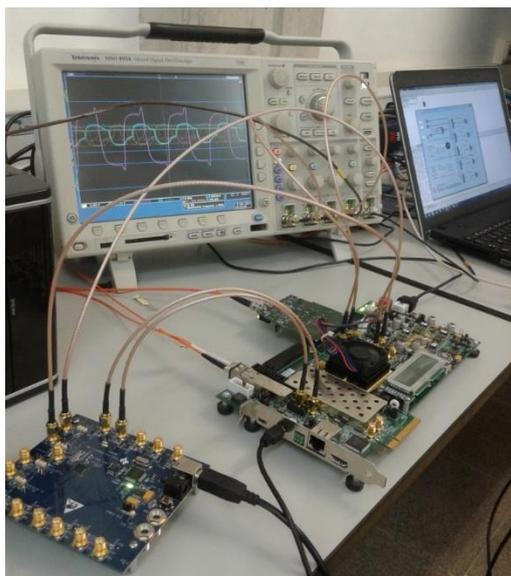


Fig. VI.1 - Montaje para el monitoreo por osciloscopio de las señales de reloj.

TABLA VI.1 Configuración del e-link en la programación de la FPGA del VC707

Parámetro configurado	Código vhdl en tile_felix_interface.vhd para la configuración del parámetro (deben coincidir con la FPGA del TileDAQ)
Codificación: 8b10b	<pre>-- ... -- elink_tx: entity work.FIF02Elink generic map (-- ... -- elinkEncoding => "01" -- ... --</pre>
e-path: eprocs de 8 bits provenientes del mini cajón 1	<pre>-- ... -- elink_tx: entity work.FIF02Elink generic map (OutputDataRate => 320, -- ... -- -- ... -- DATA1bitOUT => open, elink2bit => open, elink4bit => open, elink8bit => eproc_8in_md1 -- ... --</pre>
e-path: Habilitados 32 bits, concatenación de 4 eprocs de 8 bits.	<pre>-- ... -- if rising_edge(clk40) then if(FELIX_DATA_SELECTOR ='0') then TX_GBT_DATA_84B_I(83 downto 32) <=(others => '0'); TX_GBT_DATA_84B_I(31 downto 0) <= eproc_8in_md1 & eproc_8in_md1 & eproc_8in_md1 & eproc_8in_md1; else TX_GBT_DATA_84B_I <= rxData_from_gbtExmplDsgn; end if; -- ... --</pre>
Configuración de reloj: TTC	<pre>-- ... -- smaMGTRefCLKfroFLX: ibufds_gte2 port map (O=>mgtRefClk_from_smaMgtRefClkIbufdsGtxe2, ODIV2=> open, CEB=> '0', I=>GTX_SMA_REF_CLOCK_120MHZ_P, IB=>GTX_SMA_REF_CLOCK_120MHZ_N); -- ... -- gbtExmplDsgn: entity gbt_felix.xlx_k7v7_gbt_example_design port map (-- ... -- -- Clocks scheme: FABRIC_CLK_I=>fabricClk_from_userClockIbufgds, MGT_REFCLK_I=>mgtRefClk_from_smaMgtRefClkIbufdsGtxe2, -- ... --</pre>

El TilePPr del FELIX, antes de operar con el integrador, debe ser capaz de:

- a) Responder a un pulso aislado de trigger de L1A con duración de 50 ns

Para realizar esta prueba se deshabilita el bit de la señal *rxData_from_gbtExmplDsgn* (83 downto 0) que vincula al Canal A proveniente del *gbtExmplDsgn* con el L1A declarado en el *FLX.Felix*.

Una máquina de estado que se actualiza con la señal de reloj de 40 MHz (*clk40*) prolonga la duración de L1A activado con el botón de presión durante dos ciclos (50 ns). El trigger L1A puede ser disparado manualmente mediante el botón de pulso una vez y solamente se podrá disparar nuevamente hasta que la máquina de estado sea reiniciada presionando una vez el botón de reinicio para el *trigger* (no confundir con el botón de reinicio general). La modificación se hace en el *FLX.Felix*.

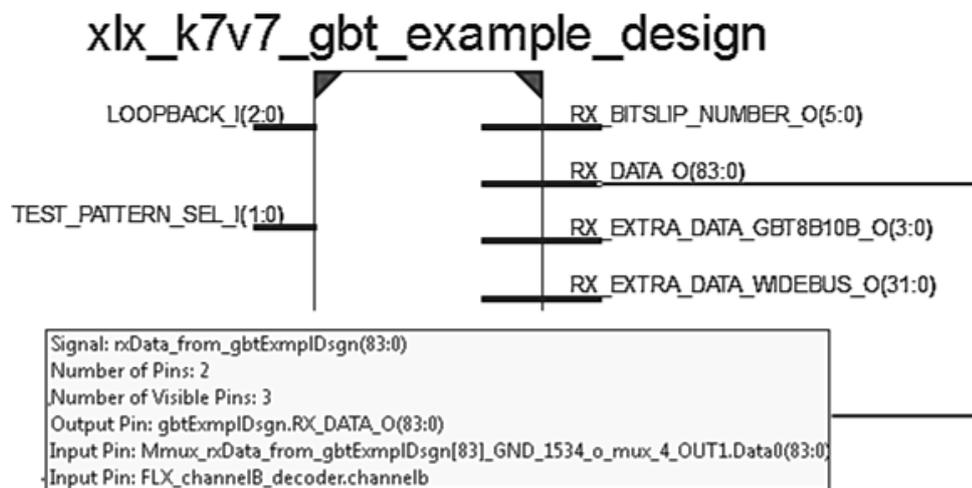


Fig. VI.2 - Señal *rxData_from_gbtExmplDsg* mostrada en esquema RTL. Esta es la señal que contiene la información y el bit del trigger L1A.

Los botones a utilizar son aquellos que no tengan un uso predeterminado. Se hace la declaración de los mismos en los constraints y se agregan las respectivas señales a la programación.

TABLA VI.2 Nueva configuración de los botones de pulso para la tarjeta de evaluación VC707

Botones de presión en la placa VC707			Uso	SW [4 downto 0]
SW3	SWN	AR40	Ocupado para rst_ipb	3
SW4	SWE	AU38	Libre para usarse: button_reset	0
SW5	SWS	AP40	Ocupado para button_in	4
SW6	SWC	AV39	Libre para usarse: button_trigger	1
SW7	SWW	AW40	Libre	2

Para enviar tramas predecibles, la configuración del patrón de datos se ajusta en el archivo *tile_config_pkg.vhd*:

- constant nSampleMaxGlinkReadout : positive := 16;
- constant NChannels_MD_FELIX : positive := 3;
- constant nSampleMaxFELIXReadout : positive := 8;

La trama de datos que se obtiene al final dependerá del número de canales y número de muestras, según el código que aparece en el archivo *tile_felix_event_packer.vhd*:

```
when TX_SAMPLES_HG_1 =>
    stateID<= x"8";
    flx_data_valid <= '1';
    flx_data_in(17 downto 16) <= code_data;
    flx_data_in(7 downto 0)<= std_logic_vector(to_unsigned(CHANNEL_cnt,8));
    flx_data_in(15 downto 8)<= std_logic_vector(to_unsigned(SAMPLES_cnt,8));
    state <= TX_SAMPLES_HG_2;
when TX_SAMPLES_HG_2 =>
    flx_data_valid <= '1';
    flx_data_in(17 downto 16) <= code_data;
    flx_data_in(7 downto 0)<= std_logic_vector(to_unsigned(CHANNEL_cnt,8));
    flx_data_in(15 downto 8)<= std_logic_vector(to_unsigned(SAMPLES_cnt+1,8));

if( SAMPLES_cnt = nSampleMaxFELIXReadout-2) then
    if(CHANNEL_cnt = NChannels_MD_FELIX-1) then
        SAMPLES_cnt<= 0;
        CHANNEL_cnt<= 0;
        state <= TX_SAMPLES_LG_1;
    else
        SAMPLES_cnt<= 0;
        CHANNEL_cnt <= CHANNEL_cnt + 1;
        state <= TX_SAMPLES_HG_1;
    end if;
else
    SAMPLES_cnt<= SAMPLES_cnt + 2;
    state<= TX_SAMPLES_HG_1;
end if;
```

El valor *nSampleMaxFELIXReadout* corresponde a la mitad del número de muestras que se obtendrán de cada canal por fragmento. La cantidad total de muestras es el doble pues se hacen dos recuentos: para LG y HG. Estos recuentos son el resultado de las señales de los PMT amplificadas en dos ganancias distintas: alta y baja.

Hay un componente de la cabecera que es importante y es el vector correspondiente a los *RunParameters*, en la versión original del VC707 sólo hace llamado a uno de los vectores.

TABLA VI.3 Trama de datos enviada por el TilePPr a través del SFP Tx sin modificación de los *RunParameters*⁶

Componente del fragmento	Posiciones dentro del fragmento	Tamaño del componente	Valores que toma el componente
SOP	--	16 b / 2 B	0000
Header 1	927 downto 912	16 b / 2 B	5678
Header 2	911 downto 896	16 b / 2 B	1234
Header 3 (RunParameters(0))	895 downto 880	16 b / 2 B	0000
Header 4 (RunParameters(0))	879 downto 864	16 b / 2 B	0000
Header 7 (MD_ID = 3)	863 downto 848	16 b / 2 B	0003
Header 8 (x0 + BCID_buffer)	847 downto 832	16 b / 2 B	0000 a 0FFF
Header 9 (L1ID_buffer)	831 downto 816	16 b / 2 B	0000 a FFFF
Header 10 (L1ID_buffer)	815 downto 800	16 b / 2 B	0000 a FFFF
Muestras y canales (HG)	799 downto 672	128 b / 16 B	0000 a 0700
	671 downto 544	128 b / 16 B	0001 a 0701
	543 downto 416	128 b / 16 B	0002 a 0702
Muestras y canales (LG)	415 downto 288	128 b / 16 B	0000 a 0700
	287 downto 160	128 b / 16 B	0001 a 0701
	159 downto 32	128 b / 16 B	0002 a 0702
Trailer 1	31 downto 16	16 b / 2 B	4321
Trailer 2	15 downto 0	16 b / 2 B	8765
EOP 1	--	16 b / 2 B	0000
EOP 2	--	16 b / 2 B	0000
EOP 3	--	16 b / 2 B	0000
EOP 4	--	16 b / 2 B	0000
Tamaño del fragmento:	927 bits / 116 Bytes		

Aunque para las aplicaciones que se requieren en este trabajo se ha dado un alcance para los valores que toman el *L1ID_buffer* y del *BCID_buffer*, estos dependen de la configuración del usuario. El tamaño del *RunParameters* a proyectar tiene por defecto el valor de 1 en la versión original del FELIX en el VC707, mostrando dos vectores de 32 bits cada uno: *RunParameters* (0) y *RunParameters* (1). Para hacer que este valor sea ajustable desde un registro externo, se modifica en el archivo *slaves.vhd*:

- `RunParameters(0) <=x"00000000";`
- `RunParameters(1) <=x"00000000";`

```
RunParamsSize <= to_integer (unsigned (FLX_RunParams (0) (15 downto 8)))+1;
```

Solo con las modificaciones previas, debería obtenerse la siguiente trama de datos:

⁶ Trama de datos elegida para la prueba experimental T_01_01 y T_01_03.

TABLA VI.4 Trama de datos enviada por el TilePPr a través del SFP Tx con modificación de los RunParameters⁷

Componente del fragmento	Posiciones dentro del fragmento	Tamaño del componente	Valores que toma el componente
SOP	--	16 b / 2 B	0000
Header 1	959 downto 944	16 b / 2 B	5678
Header 2	943 downto 928	16 b / 2 B	1234
Header 3 (RunParameters(0))	927 downto 912	16 b / 2 B	0000
Header 4 (RunParameters(0))	911 downto 896	16 b / 2 B	0000
Header 5 (RunParameters(1))	895 downto 880	16 b / 2 B	0000
Header 6 (RunParameters(1))	879 downto 864	16 b / 2 B	0000
Header 7 (MD_ID = 3)	863 downto 848	16 b / 2 B	0003
Header 8 (x0 + BCID_buffer)	847 downto 832	16 b / 2 B	0000 a 0FFF
Header 9 (L1ID_buffer)	831 downto 816	16 b / 2 B	0000 a FFFF
Header 10 (L1ID_buffer)	815 downto 800	16 b / 2 B	0000 a FFFF
Muestras y canales (HG)	799 downto 672	128 b / 16 B	0000 a 0700
	671 downto 544	128 b / 16 B	0001 a 0701
	543 downto 416	128 b / 16 B	0002 a 0702
Muestras y canales (LG)	415 downto 288	128 b / 16 B	0000 a 0700
	287 downto 160	128 b / 16 B	0001 a 0701
	159 downto 32	128 b / 16 B	0002 a 0702
Trailer 1	31 downto 16	16 b / 2 B	4321
Trailer 2	15 downto 0	16 b / 2 B	8765
EOP 1	--	16 b / 2 B	0000
EOP 2	--	16 b / 2 B	0000
EOP 3	--	16 b / 2 B	0000
EOP 4	--	16 b / 2 B	0000
Tamaño en bits del fragmento:	960 bits / 120 Bytes		

⁷ Trama de datos elegida para la prueba experimental T_01_02.

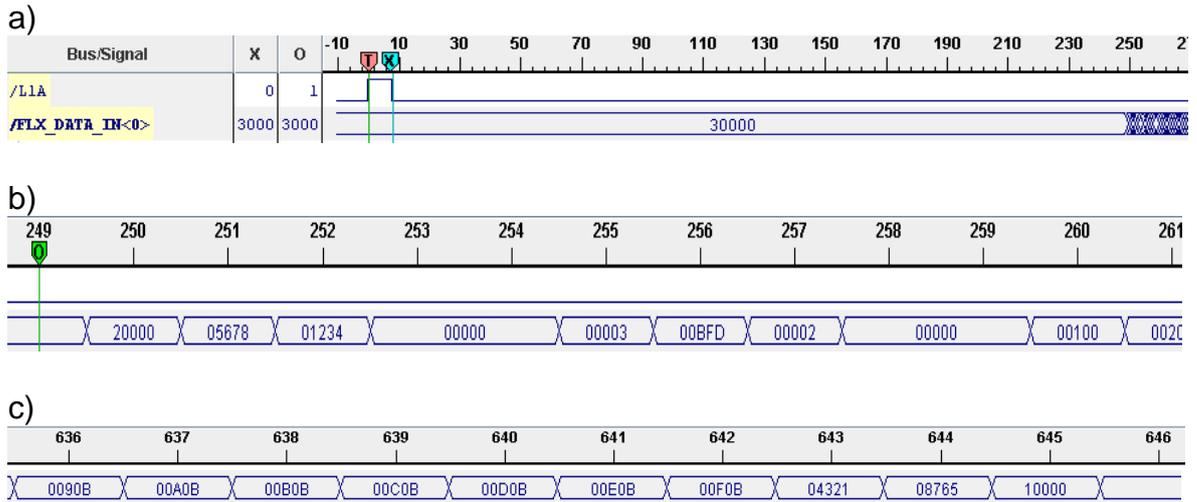


Fig. VI.3 - Formas de onda obtenidas por Chipscope con frecuencia de muestreo de 160 MHz. a) Señal de L1A con duración de 8 muestras (50 ns) y latencia de 250 muestras (1562,5 ns) para el FLX_DATA_IN, b) Cabecera de la trama de datos de FLX_DATA_IN a partir de la muestra 251, c) Trailer o cola de la trama de datos de FLX_DATA_IN que termina en la muestra 644.

- b) Responder a un número finito de pulsos de trigger L1A con duración de 50 ns, separadas entre sí por intervalos de 100 ns o a una señal proveniente del TTC.

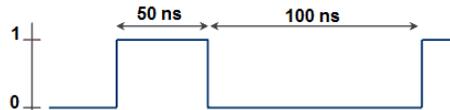


Fig. VI.4 - Duración de la señal trigger L1A y separación mínima antes del próximo trigger L1A.

Para esto se agrega a la máquina de estados una señal de 4 bits de anchura cuyo valor depende de la combinación de cuatro de los interruptores en el *dip switch* de la tarjeta de prueba. Así, se puede alternar la cantidad de *triggers* disparados cada vez que se pulsa el botón de presión, desde 1 pulso hasta 16 pulsos separados por 100 ns, como si el L1A trabajara por un breve instante a una frecuencia de 6,7 MHz. Se mantiene el mecanismo de bloqueo del botón de reinicio y el proceso sigue actualizándose con una señal de reloj de 40 MHz (*clk40*).

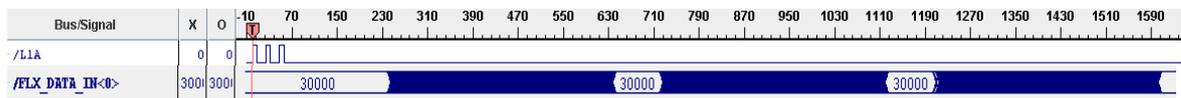


Fig. VI.5 - Señal de FLX_DATA_IN resultante de tres pulsos consecutivos L1A.

La configuración del patrón de datos se ajusta en el *tile_config_pkg.vhd*:

- constant nSampleMaxGlinkReadout : positive := 16;
- constant NChannels_MD_FELIX: positive := 12;
- constant nSampleMaxFELIXReadout : positive := 16;

TABLA VI.5 Trama de datos enviada por el TilePPr a través del SFP Tx sin modificación de los RunParameters y con los doce canales⁸

Componente del fragmento	Posiciones dentro del fragmento	Tamaño del componente	Valores que toma el componente
SOP	--	16 b / 2 B	0000
Header 1	6303 downto 6288	16 b / 2 B	5678
Header 2	6287 downto 6272	16 b / 2 B	1234
Header 3 (RunParameters(0)(15 downto 0))	6271 downto 6256	16 b / 2 B	0000
Header 4 (RunParameters(0)(31 downto 16))	6255 downto 6240	16 b / 2 B	0000
Header 5 (MD_ID = 3)	6239 downto 6224	16 b / 2 B	0003
Header 6 (x0 + BCID_buffer)	6223 downto 6208	16 b / 2 B	0000 a 0FFF
Header 7 (L1ID_buffer)	6207 downto 6192	16 b / 2 B	0000 a FFFF
Header 8 (L1ID_buffer)	6191 downto 6176	16 b / 2 B	0000 a FFFF
Muestras y canales (HG)	6175 downto 5920	256 b / 32 B	0000 a 0f00
	5919 downto 5664	256 b / 32 B	0001 a 0f01
	5663 downto 5408	256 b / 32 B	0002 a 0f02
	5407 downto 5152	256 b / 32 B	0003 a 0f03
	5151 downto 4896	256 b / 32 B	0004 a 0f04
	4895 downto 4640	256 b / 32 B	0005 a 0f05
	4639 downto 4384	256 b / 32 B	0006 a 0f06
	4383 downto 4128	256 b / 32 B	0007 a 0f07
	4127 downto 3872	256 b / 32 B	0008 a 0f08
	3871 downto 3616	256 b / 32 B	0009 a 0f09
	3615 downto 3360	256 b / 32 B	000a a 0f0a
	3359 downto 3104	256 b / 32 B	000b a 0f0b
Muestras y canales (LG)	3103 downto 2848	256 b / 32 B	0000 a 0f00
	2847 downto 2592	256 b / 32 B	0001 a 0f01
	2591 downto 2336	256 b / 32 B	0002 a 0f02
	2335 downto 2080	256 b / 32 B	0003 a 0f03
	2079 downto 1824	256 b / 32 B	0004 a 0f04
	1823 downto 1568	256 b / 32 B	0005 a 0f05
	1567 downto 1312	256 b / 32 B	0006 a 0f06
	1311 downto 1056	256 b / 32 B	0007 a 0f07
	1055 downto 800	256 b / 32 B	0008 a 0f08
	799 downto 544	256 b / 32 B	0009 a 0f09
	543 downto 288	256 b / 32 B	000a a 0f0a
	287 downto 32	256 b / 32 B	000b a 0f0b
Trailer 1	31 downto 16	16 b / 2 B	4321
Trailer 2	15 downto 0	16 b / 2 B	8765
EOP 1	--	16 b / 2 B	0000
EOP 2	--	16 b / 2 B	0000
EOP 3	--	16 b / 2 B	0000
EOP 4	--	16 b / 2 B	0000
Tamaño del fragmento:	6304 bits / 788 Bytes		

⁸ Formato de fragmento para las pruebas de validación T_03_01, T_03_02, T_03_03, T_03_04.

El tamaño del RunParameters de la cabecera se modifica en el archivo *slaves.vhd*:

TABLA VI.6 Trama de datos enviada por el TilePPr a través del SFP Tx con modificación de los RunParameters y con los doce canales

Componente del fragmento	Posiciones dentro del fragmento	Tamaño del componente	Valores que toma el componente
SOP	--	16 b / 2 B	0000
Header 1	6399 downto 6384	16 b / 2 B	5678
Header 2	6383 downto 6368	16 b / 2 B	1234
Header 3 (RunParameters(0))(15 downto 0)	6367 downto 6352	16 b / 2 B	03c0
Header 4 (RunParameters(0))(31 downto 16)	6351 downto 6336	16 b / 2 B	0000
Header 5 (RunParameters(1))(15 downto 0)	6335 downto 6320	16 b / 2 B	0000
Header 6 (RunParameters(1))(31 downto 16)	6319 downto 6304	16 b / 2 B	0000
Header 7 (RunParameters(2))(15 downto 0)	6303 downto 6288	16 b / 2 B	0000
Header 8 (RunParameters(2))(31 downto 16)	6287 downto 6272	16 b / 2 B	0000
Header 9 (RunParameters(3))(15 downto 0)	6271 downto 6256	16 b / 2 B	0000
Header 10 (RunParameters(3))(31 downto 16)	6255 downto 6240	16 b / 2 B	0000
Header 11 (MD_ID = 3)	6239 downto 6224	16 b / 2 B	0003
Header 12 (x0 + BCID_buffer)	6223 downto 6208	16 b / 2 B	0000 a 0FFF
Header 13 (L1ID_buffer)	6207 downto 6192	16 b / 2 B	0000 a FFFF
Header 14 (L1ID_buffer)	6191 downto 6176	16 b / 2 B	0000 a FFFF
Muestras y canales (HG)	6175 downto 5920	256 b / 32 B	0000 a 0f00
	5919 downto 5664	256 b / 32 B	0001 a 0f01
	5663 downto 5408	256 b / 32 B	0002 a 0f02
	5407 downto 5152	256 b / 32 B	0003 a 0f03
	5151 downto 4896	256 b / 32 B	0004 a 0f04
	4895 downto 4640	256 b / 32 B	0005 a 0f05
	4639 downto 4384	256 b / 32 B	0006 a 0f06
	4383 downto 4128	256 b / 32 B	0007 a 0f07
	4127 downto 3872	256 b / 32 B	0008 a 0f08
	3871 downto 3616	256 b / 32 B	0009 a 0f09
	3615 downto 3360	256 b / 32 B	000a a 0f0a
	3359 downto 3104	256 b / 32 B	000b a 0f0b
Muestras y canales (LG)	3103 downto 2848	256 b / 32 B	0000 a 0f00
	2847 downto 2592	256 b / 32 B	0001 a 0f01
	2591 downto 2336	256 b / 32 B	0002 a 0f02
	2335 downto 2080	256 b / 32 B	0003 a 0f03
	2079 downto 1824	256 b / 32 B	0004 a 0f04
	1823 downto 1568	256 b / 32 B	0005 a 0f05
	1567 downto 1312	256 b / 32 B	0006 a 0f06
	1311 downto 1056	256 b / 32 B	0007 a 0f07
	1055 downto 800	256 b / 32 B	0008 a 0f08
	799 downto 544	256 b / 32 B	0009 a 0f09
	543 downto 288	256 b / 32 B	000a a 0f0a
	287 downto 32	256 b / 32 B	000b a 0f0b
Trailer 1	31 downto 16	16 b / 2 B	4321
Trailer 2	15 downto 0	16 b / 2 B	8765
EOP 1	--	16 b / 2 B	0000
EOP 2	--	16 b / 2 B	0000
EOP 3	--	16 b / 2 B	0000
EOP 4	--	16 b / 2 B	0000
Tamaño en bits del fragmento:			6400 bits / 800 Bytes

- RunParameters(0) <=x"000003c0";
- RunParameters(1) <=x"00000000";
- RunParameters(2) <=x"00000000";
- RunParameters(3) <=x"00000000";

Ya en el *tile_felix_event_packer.vhd* se establece la dependencia respecto al *RunParameters*:

```
RunParamsSize <= to_integer (unsigned (FLX_RunParams (0) (15 downto 8)));
```

De acuerdo al valor que posean estos ocho bits, se muestran en la cabecera los parámetros que el usuario desea proyectar en la trama.

Partiendo de la máquina de estados cuya cantidad de pulsos se obtienen del *dip switch* de la tarjeta de prueba, se deja uno de los *dip switch* para alternar entre la fuente de triggers de L1A: entre los pulsos manuales o la señal proveniente de *gbtExmplDsgn*. Reasignar el último *dip switch* para escoger la fuente de la señal de L1A, limita la cantidad de disparos manuales L1A hasta 8. La trama de datos es la misma que la usada para responder a un número finito de L1A, aunque ahora se puede alternar en cualquier momento los L1A provenientes del trigger manual con los provenientes del TTC.

VI. 1. 2 Configuración del e-link en el lado del servidor

La ejecución del comando source *setup.sh* pone en marcha el programa de configuración proporcionado por el CERN. Con el programa y la instrucción *elinkconfig*, se ajusta de acuerdo a la programación de la FPGA del VC707:

TABLA VI.7 Configuración del E-link en el firmware de la FPGA del servidor local

Parámetros configurados que deben coincidir con el TilePPr	
Codificación:	8b10b
Epath:	Ruta 1 (To-Host/From-GBT), datos de eproc a leer 8 bits (7 downto 0) de los 32 enviados por el TilePPr.
Configuración de reloj:	TTC
Configuración adicional (afectan solamente la interacción GBT-servidor local)	
Link GBT:	0
Emulador GBT To-Host	None (ninguno)
Emulador GBT From-Host	None (ninguno)
Time-out Datablocks	Time-out enabled (necesario sólo para la prueba de un pulso): 64 x 25 ns
	Time-out disabled (para las demás pruebas)
Max Chunk	3584 bits for 2, 4, 8, 16 bit.

VI. 1. 3 Conexión estable entre TilePPr y TileDAQ

Para asegurar la conexión estable entre las dos FPGA, se programa en el TilePPr una señal que lleve a un terminal SMA la señal de reloj del FELIX_DATA_CLK, la misma debe estar en sincronía con el TTC. La forma para verificar esta sincronía es con la ayuda de un osciloscopio en la que se coloquen tres canales: uno para la señal de TTC, otro para el reloj de Tx del VC707 y otro para el reloj Rx del VC707. Con el *trigger* del osciloscopio asignado a la señal TTC, las señales de los tres canales deben en el osciloscopio.

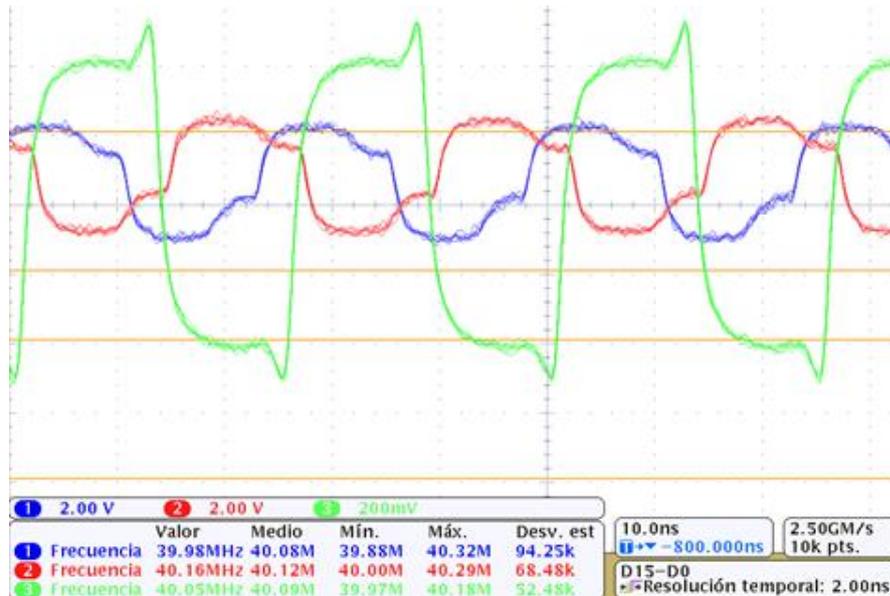


Fig. VI.6 - Señales de reloj de la VC707 obtenidas por osciloscopio. Verde: TTC. Rojo: señal de reloj Tx. Azul: señal de reloj Rx.

El programa proveído por el CERN realiza la adquisición de datos y hace una evaluación de las tramas de datos a nivel de bloque y superficial a nivel de fragmento.

VI. 2. Software para la detección de errores más específicos

La programación de ambas FPGA debe ser capaz de transmitir y almacenar miles de bloques de información donde cada fragmento es una trama de datos enviada por el VC707 o el TilePPr, por lo que para verificar es mejor utilizar un programa de uso libre. Como parte de este trabajo se desarrolla en Python un software que facilite esta tarea. Las funciones que se programan en la versión terminal del software *unboxing_v0_4.py* son:

a) Generales:

- Conversión de archivo binario .dat a archivo hexadecimal en .txt.
- Cantidad de caracteres en formato hexadecimal.
- Número de bloques calculado.

b) A nivel de bloques:

- Creación de archivo .txt en formato de bloques: cada línea corresponde a un bloque.
 - Proyección en el terminal de un bloque en particular.
 - Determinar la secuencia de cabecera (abcd ó cdab) en los bloques.
 - Verificar las cabeceras de los bloques con la secuencia especificada; contar los bloques con cabeceras correctas e incorrectas.
- c) A nivel de fragmentos:
- Creación de archivo .txt en formato de fragmentos: cada línea corresponde a un fragmento.
 - Determinar la secuencia de cabecera (5678 por defecto) y la secuencia del *trailer* o cola (4321 por defecto) en los fragmentos.
 - Verificar las cabeceras y colas con la secuencia especificada; contar la cantidad de cabeceras y colas correctas encontradas en los fragmentos.
- d) A nivel de canal:
- Creación de archivo .txt en formato de canales: idéntico al del formato en fragmento, pero eliminando el *header* y *trailer*.
 - Recuento de la cantidad de muestras encontradas para todos y cada uno de los canales (desde el 00 al 0f).
 - Creación de un archivo .txt con las muestras correspondientes a un solo canal.
- e) A nivel de datos:
- Comparación entre las muestras de todos los canales con el rango de valores de 00 a 0f. No verifica si la secuencia es correcta, solamente si el valor de la muestra está dentro de un intervalo en específico.

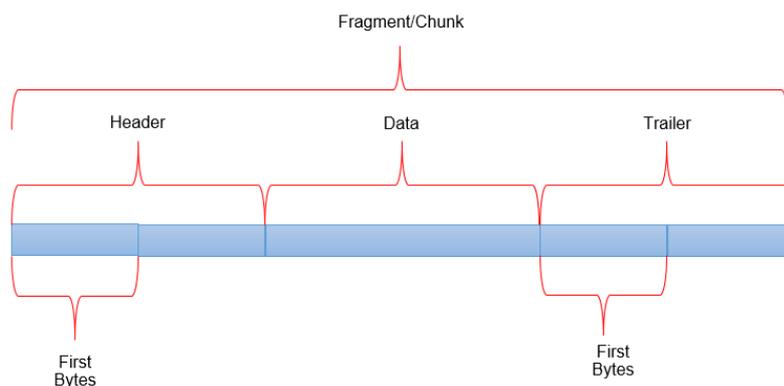


Fig. VI.7 - Estructura del fragmento y componentes considerados para el análisis por software.

Las funciones, aunque limitadas, permiten verificar automáticamente una mayor cantidad de datos de los cuales se espera que el tamaño de los bloques, fragmentos, cabeceras, colas y canales se mantengan durante toda la adquisición de datos. Esta versión es pensada para ser utilizada por personal experimentado en Python, para ser modificada y optimizada fácilmente.

La versión mejorada con interfaz gráfica *unboxing_v01_1.py* es más versátil, rápida y con algunas funciones adicionales:

- a) Generales:
 - Interfaz gráfica, configuración mediante cajas de texto y comandos ejecutados con botones.
 - Secuencia de comandos única: los menús solamente se activan cuando los archivos y análisis previos son ejecutados.
- b) A nivel de bloque:
 - Ventana individual que muestra el bloque especificado.
- c) A nivel de fragmento:
 - Tamaño de cabeceras, cola y *data* ajustables. Pueden escogerse valores por defecto o incluso un cálculo automático para FELIX.
 - Cálculo del número esperado de fragmentos.
 - Cálculo del porcentaje de error entre el número esperado de fragmentos y el número de fragmentos con cabeceras correctas.
 - Conteo de la cantidad de lecturas erróneas de cabeceras durante su búsqueda.
- d) A nivel de canal:
 - Conteo más rápido de muestras por canal y proyección por tabla.
 - Cálculo del número esperado de muestras totales.
 - Cálculo del porcentaje de error entre el número esperado de muestras totales y las que se encontraron.
 - Mayor facilidad para la creación del archivo de datos para un solo canal, organizando las muestras de un canal por fragmento en cada línea.

Para esta verificación la cantidad de fragmentos y bloques es mínima, así que se puede comparar los resultados obtenidos del programa basado en python con los teóricos visualmente.

```
For just one L1A trigger:
Channel 01
TFM_02_un_solo_puls...lected_chan_only.txt
00 01 02 03 04 05 06 07 00 01 02 03 04 05 06 07

For three consecutive L1A trigger:
Channel 00
TFM_01_tren_tres_pu...lected_chan_only.txt
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f

Channel 0a
TFM_01_tren_tres_pu...lected_chan_only.txt
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f |
```

Fig. VI.8 - Datos extraídos de cada canal con ayuda del software *unboxing_v01_1.py*.

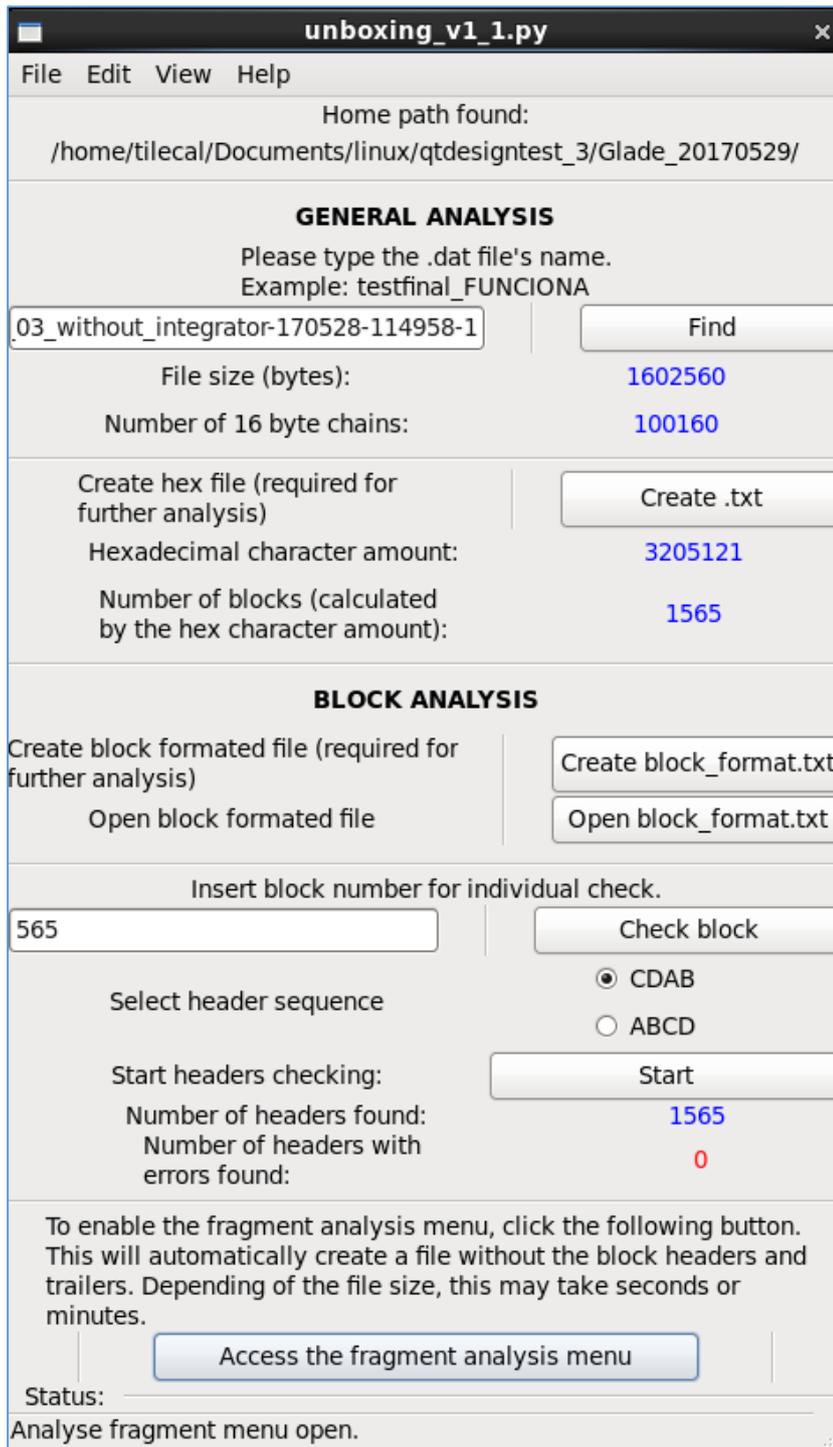


Fig. VI.9 - Imagen del software *Unboxing v.1.1*, la primer ventana brinda opciones para el análisis general del archivo binario y sus bloques.

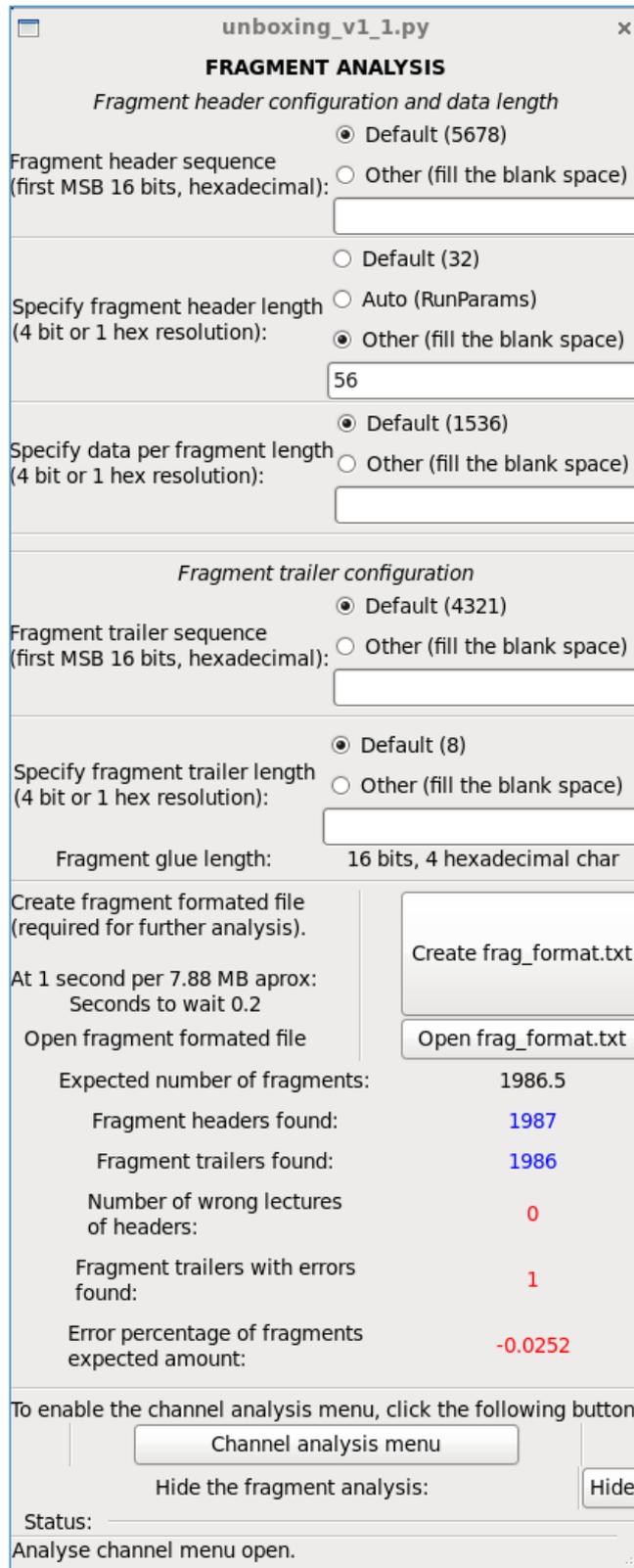


Fig. VI.10 - Imagen del software Unboxing v.1.1, la segunda ventana brinda opciones para el análisis de los fragmentos.

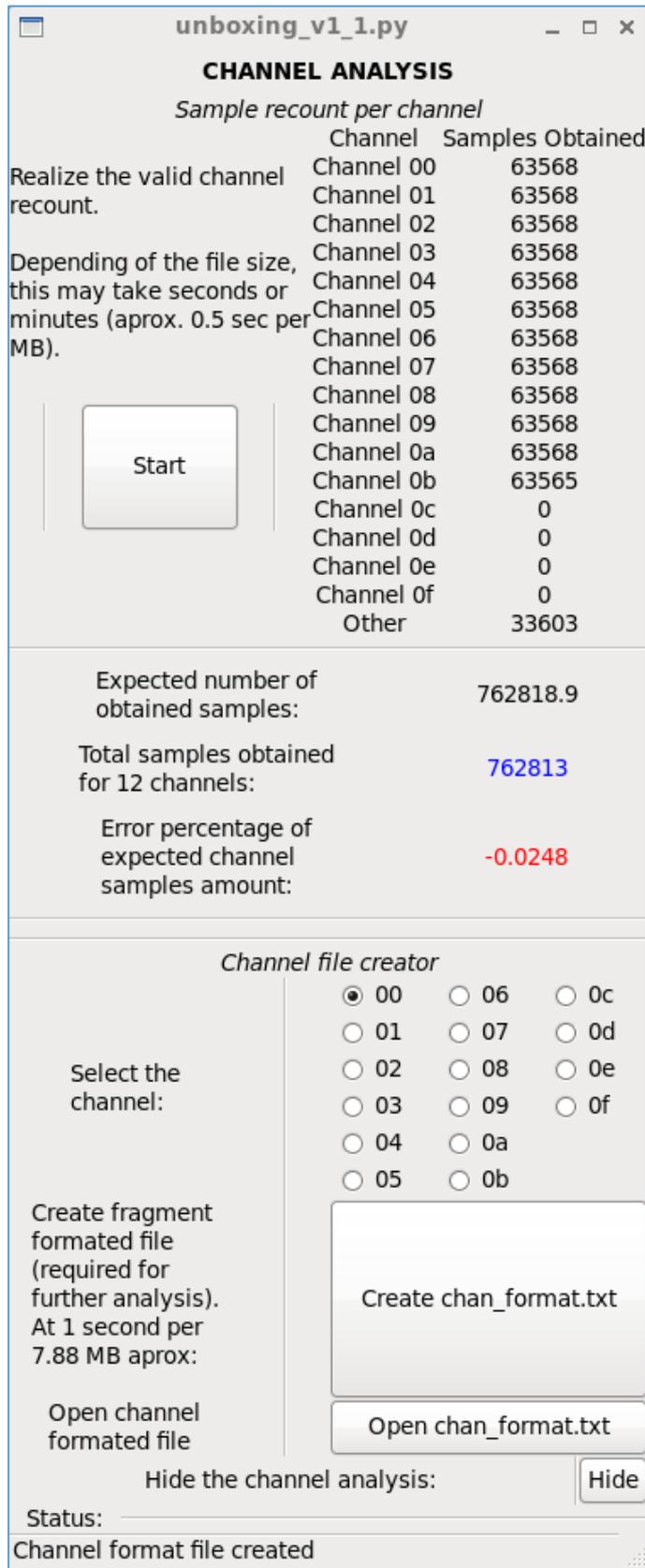


Fig. VI.11 - Imagen del software Unboxing v.1.1, la tercera y última ventana brinda opciones para el análisis de los canales y crea el archivo de los datos obtenidos.

VI. 3. Programación hardware del módulo que comunique el integrador al servidor local

VI. 3. 1 Propuesta de la trama de datos para el integrador

Se manejará un protocolo diferente a los datos que maneja FELIX. La trama de información elaborada para enviar datos en un solo e-proc es:

TABLA VI.8 Trama de datos para ser enviada por el TilePPr a través del SFP Tx: Único e-proc de 2 bits⁹

Componente del fragmento	Posiciones dentro del fragmento	Tamaño del componente	Valores que toma el componente
SOP	--	16 b / 2 B	FFFF
Header 1 (secuencia de ceros)	175 downto 160	16 b / 2 B	0000
Header 2 (lado)	159 downto 144	16 b / 2 B	000A o 000B
Header 3 (No. de fragmento)	143 downto 128	16 b / 2 B	0000 a 000F
Datos ¹⁰	127 downto 112	16 b / 2 B	0000
	111 downto 96	16 b / 2 B	1111
	95 downto 80	16 b / 2 B	2222
	79 downto 64	16 b / 2 B	3333
	63 downto 48	16 b / 2 B	4444
	47 downto 32	16 b / 2 B	5555
Trailer 1 (No. de fragmento esperado a continuación)	31 downto 16	16 b / 2 B	No. de fragmento +1 = 0000 a 000F
Trailer 2	15 downto 0	16 b / 2 B	No. de fragmento +2 = 0000 a 000F
EOP	--	16 b / 2 B	FFFF
Tamaño en bits del fragmento	176 bits / 22 bytes		

Debe ser independiente de los datos transferidos por el L1A, por lo que durante las pruebas se estará enviando ininterrumpidamente tramas de datos. Además, debe excluirse de la recepción del servidor local fácilmente quitando el e-path correspondiente.

⁹ Trama de datos elegida para las pruebas experimentales T_02_01, T_02_02, T_02_03, T_02_04, T_02_05, T_02_06, T_02_07.

¹⁰ Por cada lado de la placa electrónica principal se leen seis canales, haciendo un total de doce. En dos fragmentos se tendría lectura de ambos canales.

TABLA VI.9 Trama de datos para ser enviada por el TilePPr a través del SFP Tx: Dos e-procs de 2 bits¹¹

Componente del fragmento	Posiciones dentro del fragmento	Tamaño del componente	Valores que toma el componente
SOP	--	16 b / 2 B	DDDD
Header 1 (año de edición)	271 downto 256	16 b / 2 B	2017
Header 1 (No. de fragmento)	255 downto 240	16 b / 2 B	0000 a FFFF
Header 2 (lado)	239 downto 224	16 b / 2 B	000A o 000B
No. de registro del primer canal ¹²	223 downto 208	16 b / 2 B	0000 a FFFF
Datos del primer canal	207 downto 192	16 b / 2 B	0000 a FFFF
No. de registro del segundo canal	191 downto 176	16 b / 2 B	0000 a FFFF
Datos del segundo canal	175 downto 160	16 b / 2 B	0000 a FFFF
No. de registro del tercer canal	159 downto 144	16 b / 2 B	0000 a FFFF
Datos del tercer canal	143 downto 128	16 b / 2 B	0000 a FFFF
No. de registro del cuarto canal	127 downto 112	16 b / 2 B	0000 a FFFF
Datos del cuarto canal	111 downto 96	16 b / 2 B	0000 a FFFF
No. de registro del quinto canal	95 downto 80	16 b / 2 B	0000 a FFFF
Datos del quinto canal	79 downto 64	16 b / 2 B	0000 a FFFF
No. de registro del sexto canal	63 downto 48	16 b / 2 B	0000 a FFFF
Datos del sexto canal	47 downto 32	16 b / 2 B	0000 a FFFF
Trailer 1 (Valor preestablecido)	31 downto 16	16 b / 2 B	4446
Trailer 2 (Valor preestablecido)	15 downto 0	16 b / 2 B	4456
EOP	--	16 b / 2 B	FFFF
Tamaño en bits del fragmento	272 bits / 34 bytes		

¹¹ Formato de fragmento para las pruebas de validación T_03_04.

¹² Por cada lado del integrador se leen seis canales, haciendo un total de doce. Los fragmentos son enviados por lado, independientes uno de otro.

La diferencia entre las dos versiones de tramas de datos es simplemente que el primer diseño de trama no provee mucha información para verificar el correcto funcionamiento de los DAQs del integrador pero su tamaño reduce la latencia del envío de datos usando al final un solo e-proc de 2 bits con información de ambos lados del integrador.

Para las primeras pruebas, los datos que recibe el módulo *INT2HOST* se generarán con un patrón bien conocido desde la misma programación vhd. Para sus pruebas de validación, el laboratorio TileCal del IFIC-Valencia cuenta con una placa electrónica principal y varios DAQs que también se usan en el TileCal del CERN.

VI. 3. 2 Módulo *INT2HOST* (*integrator2host_top.vhd*)

Dentro del módulo se instanciarán como IP cores el *integrator_receiver.vhd*, el *FIFO2Elink* y en las primeras etapas un MMCM síncrono con la señal de reloj de 40 MHz. En la señal de salida se obtendrán los datos listos como e-proc de 2 bits. De esta forma, los datos del integrador serán enviados a través de la SFP aprovechando el mismo GBT instanciado dentro del *tile_felix_interface.vhd*. El módulo se instancia dentro del *FLX.Felix* y tiene por definición el código:

```
ENTITY integrator2host_top IS
PORT
(
  ---- GBT LINK RESETS ----
  GENERAL_RESET_I          : IN STD_LOGIC;

  ---- INPUT CLOCKS-----
  EXTERNAL_CLK40_FROM_FELIX : IN STD_LOGIC;
  EXTERNAL_CLK80_FROM_FELIX : IN STD_LOGIC;
  EXTERNAL_CLK160_FROM_FELIX : IN STD_LOGIC;
  EXTERNAL_CLK320_FROM_FELIX : IN STD_LOGIC;

  INT_FIFO_FULL          : OUT STD_LOGIC_VECTOR (3 downto 0);
  INT_ENABLE_READOUT    : OUT STD_LOGIC;

  -- INTEGRATOR 2 FIFO

  IntegratorData1_MD1    : IN STD_LOGIC_VECTOR (4 downto 0);
  IntegratorData2_MD1    : IN STD_LOGIC_VECTOR (4 downto 0);
  GBT_rx_clk             : IN STD_LOGIC;
  MGT_REFCLK_I_INT2HOST : IN STD_LOGIC; -- Erase for A and B separate
  TEST_OUT              : OUT STD_LOGIC_VECTOR (1 downto 0);
  ELINK2BIT_2_TXDATAGBT : OUT STD_LOGIC_VECTOR (83 downto 0);
  -- ELINK2BIT_2_TXDATAGBT
  separate              : OUT STD_LOGIC_VECTOR (3 downto 0); -- A and B
  MGT_REFCLK_I          : OUT STD_LOGIC -- Erase for A and B separate
);
end integrator2host_top;
```

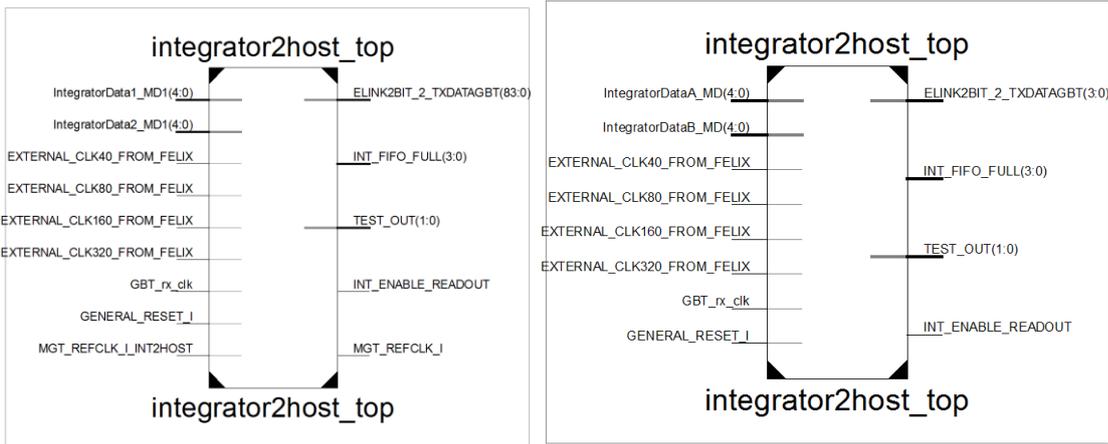


Fig. VI.12 - Esquema RTL Top del módulo INT2HOST. A la izquierda, para un solo e-proc de 2 bits. A la derecha, para dos e-proc de 2 bits.

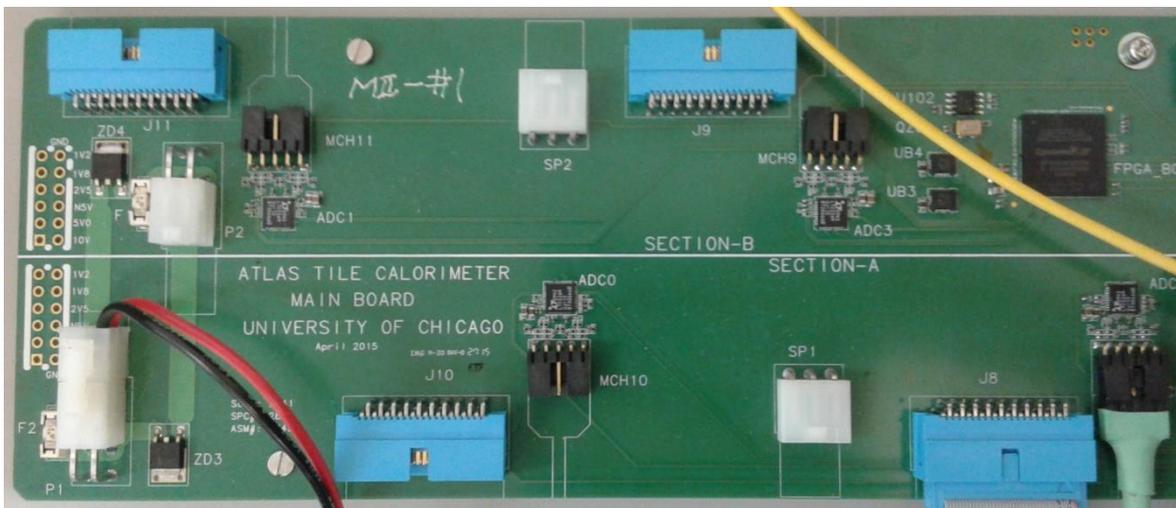


Fig. VI.13 - Vista del lado A (SECTION-A) y el lado B (SECTION-B) de la placa electrónica principal. A cada lado corresponde una señal en la programación de la VC707: para el lado A es la IntegratorData1_MD1 y para el lado B es IntegratorData2_MD1.

El módulo de comunicación está habilitado para operar con uno de los cuatro mini cajones, en este caso para el MD1 o mini cajón 1. Dentro del *integrator2host_top.vhd* se mapean los cuatro módulos esenciales con los que funciona:

VI. 3. 3 INT2FIFO_MD1 (*integrator2fifo_top.vhd*)

Se encarga de adaptar las señales de datos de 5 bits provenientes de los mini cajones para las especificaciones del *FIFO2Elink.vhd*. Toma de referencia la señal de reloj de 80 MHz generada por uno de los MMCM instanciado dentro del *felix_interface.vhd*, síncrona con la señal de 40 MHz. Contiene los generadores de señales para realizar las pruebas en la tarjeta electrónica VC707.

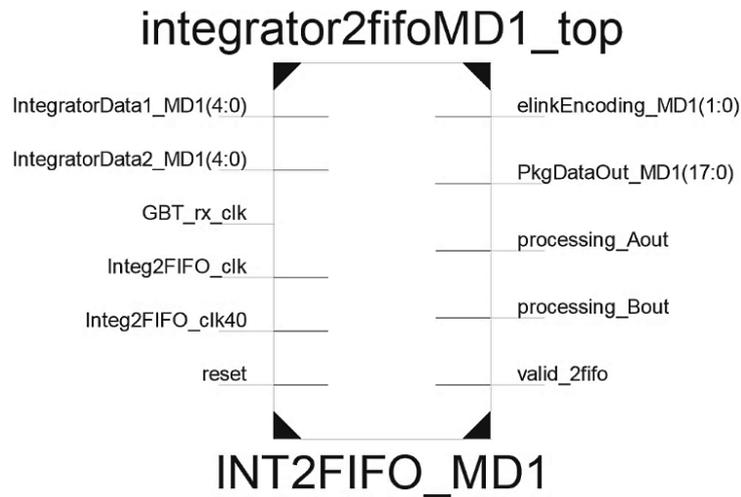


Fig. VI.14 - Esquema RTL: Top del módulo INT2FIFO_MD1. Se requiere uno por mini cajón.

VI. 3. 3. 1 EXDATA_S2F_MD1 (example_data_generator.vhd)

Simula las señales que se obtienen para uno de los mini cajones, tanto el lado A como el lado B. La frecuencia con la que funciona el generador de datos es estática, 20 MHz utilizando Flip-Flops síncronos. La secuencia de datos generada sirve para probar el correcto funcionamiento del sistema, tanto para permitir los datos válidos como para rechazar los no válidos pues entrega una trama para llenar un registro de 9 vectores.

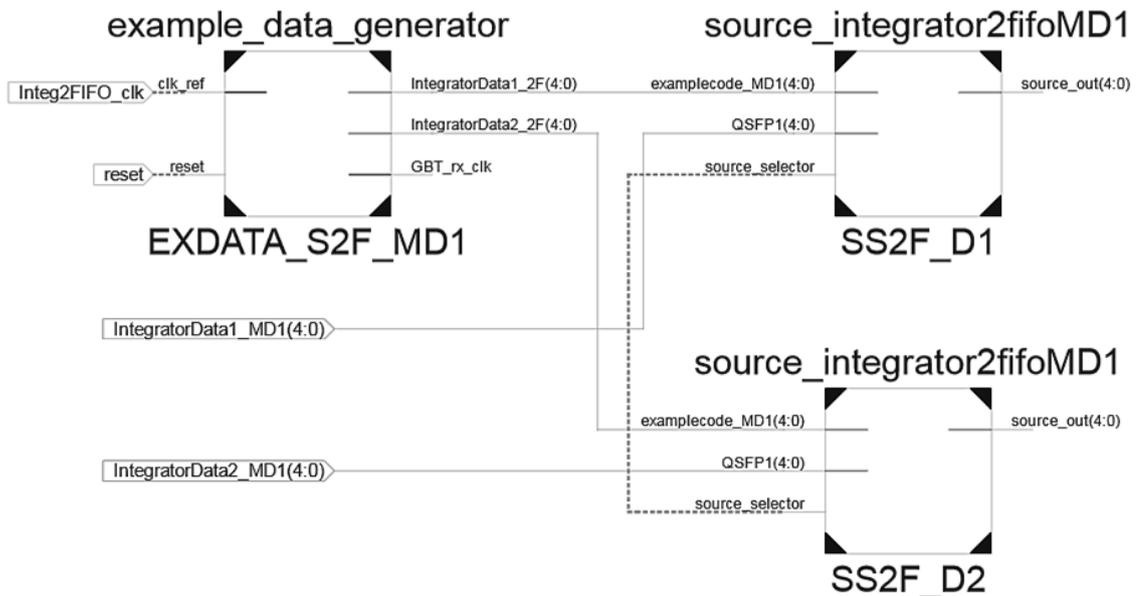


Fig. VI.15 - Esquema RTL: generador de datos para las pruebas y los módulos SS2F.

VI. 3. 3. 2 SS2F_D1 y SS2F_D2
(source_integrator2fifoMD1.vhd)

Multiplexor de dos entradas y una salida, todas con 5 bits de anchura. Permite alternar entre tomar como fuente de datos el EXDATA_S2F_MD1 (example_data_generator.vhd) o la señales integratorData1_MD1 e integratorData2_MD1, para enviarla al INT_MD1 (integrator_receiver.vhd) instanciado.

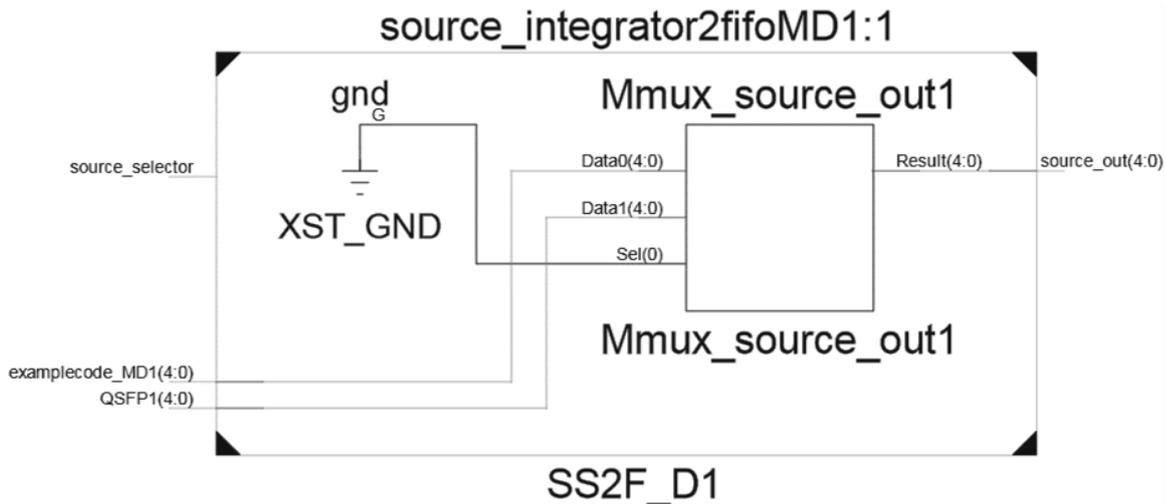


Fig. VI.16 - Esquema RTL: selector de fuente SS2F_D1. El multiplexor depende de una señal con valor fijado desde antes del sintetizado.

VI. 3. 3. 3 INT_MD1 (integrator_receiver.vhd)

Código previamente programado para funcionar dentro del sROD_processing_module y explicado en páginas anteriores. Este módulo toma los datos válidos que reciba a la salida del SS2F_D1 y del SS2F_D2 para guardarlos en los registros IntegratorOutDataA e IntegratorOutDataB de 6 vectores con 32 bits de anchura, los dos registros integ_weA e integ_weB de 6 vectores y de un bit de anchura. Sus procesos operan a una frecuencia de 80 MHz.

VI. 3. 3. 4 IWEAclk80toclk40 e IWEBclk80toclk40
(Integ_wAB_clk80toclk40.vhd)

Máquina de estados y flip flop con reset síncrono, prolonga la duración de la señal cuando toma el valor de un bit de integ_weA e integ_weB obtenida del proceso a 80 MHz (una muestra) para que sea detectada por un proceso de 40 MHz (tres muestras), es decir, para los procesos del P2F_MD2. Sus salidas se conectan a las señales Integ_weA_signal_clk40 y Integ_weB_signal_clk40. Sus procesos se actualizan con una frecuencia de 80 MHz (señal Integ2FIFO_clk).

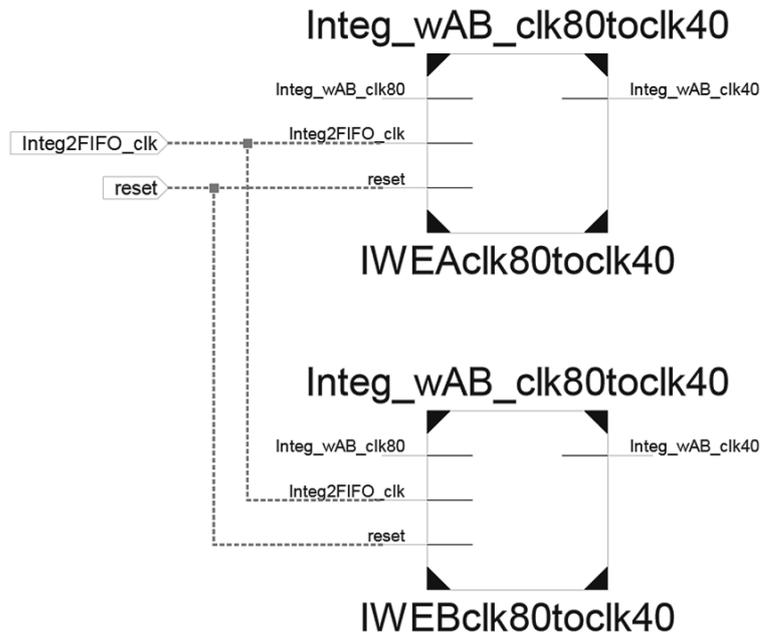


Fig. VI.17 - Esquema RTL: prolongadores de duración de las señales integ_weA e integ_weB.

VI. 3. 3. 5 P2F_MD1 (integrator_pack2fifo_16bits)

El módulo de mayor importancia dentro de *INT2FIFO_MD1*, procesa los datos que se reciben y los devuelve de acuerdo al diseño de la trama especificada. Sus procesos se actualizan con una frecuencia de 40 MHz (señal *Integ2FIFO_clk40*).

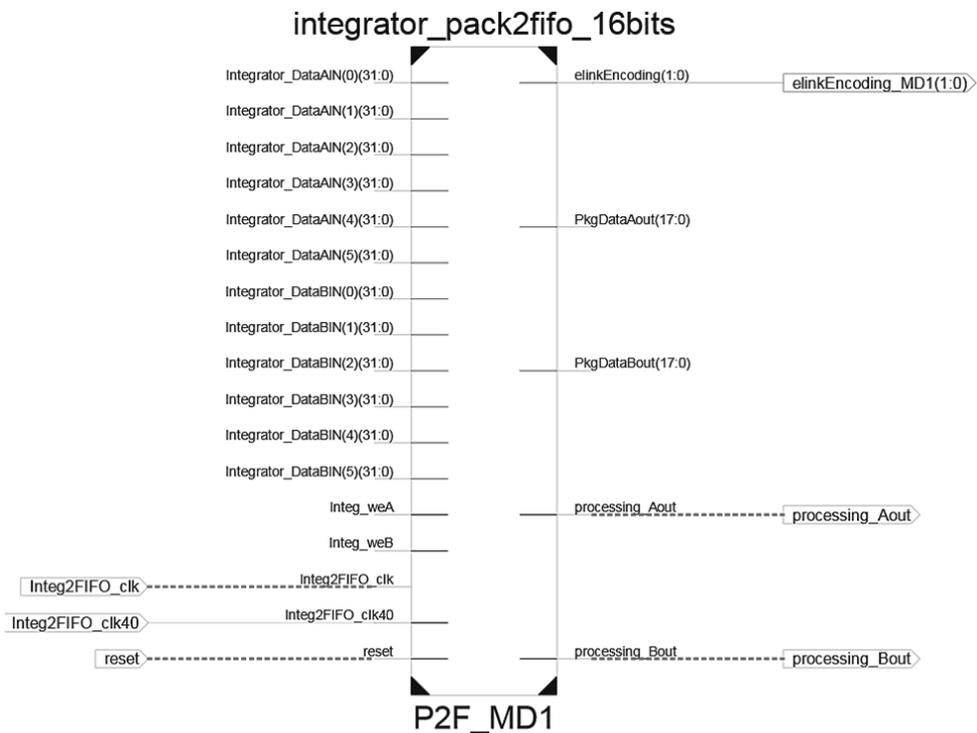


Fig. VI.18 - Esquema RTL: Top del P2F_MD1. Se requiere uno por mini cajón.

Los componentes más relevantes del módulo *P2F_MD1* son:

- *IntegratorDataA* e *IntegratorDataB*

Una vez llenos y enviados los registros por el *INT_MD1*, los datos dentro de los mismos (precisamente los últimos 16 bits más significativos de cada vector en el registro) se asignan a un solo vector de 96 bits de longitud, estos son las señales *IntegratorDataA* e *IntegratorDataB*.

- Máquina de Estado para el Lado A y Máquina de Estado para el Lado B

Descrita dentro de los procesos con reinicio asíncrono y con frecuencia de actualización 40 MHz (señal *Integ2FIFO_clk40*) *STATE_M_A* y *STATE_M_B*, se encargan de elaborar la trama de datos que será enviada a la salida del *INT2FIFO_MD1* y las señales que pondrán en marcha al *FIFO2Elink*. Cada máquina de estado es independiente de la otra y pueden operar simultáneamente. Para sacarla del estado *idle* se necesita de las señales *Integ_weA_signal_clk40* o *Integ_weB_signal_clk40*, que tendrán el valor de '1' cuando los registros del lado respectivo estén llenos.

Su salida consiste principalmente en señales de 18 bits llamadas *PkgData* (*PkgDataA* y *PkgDataB*) y dos señales que indican al *FIFO2Elink* cuándo se están enviando tramas de datos (*processing_A* y *processing_B*).

VI. 3. 3. 6 *S2F_MD1* (*integrator_side2fifo_16bits*)

Multiplexor, compuertas "or", "xor". Limita los datos de salida *PkgData* a un solo lado a la vez. Para pruebas y ensayos, está predeterminado seleccionar el lado "A". Su salida, el puerto *PkgDataOut* de 18 bits de anchura, se conecta directamente a la salida del *INT2FIFO_MD1* mientras que la señal que habilita al *FIFO2Elink* sale del puerto *valid_2fifo*.

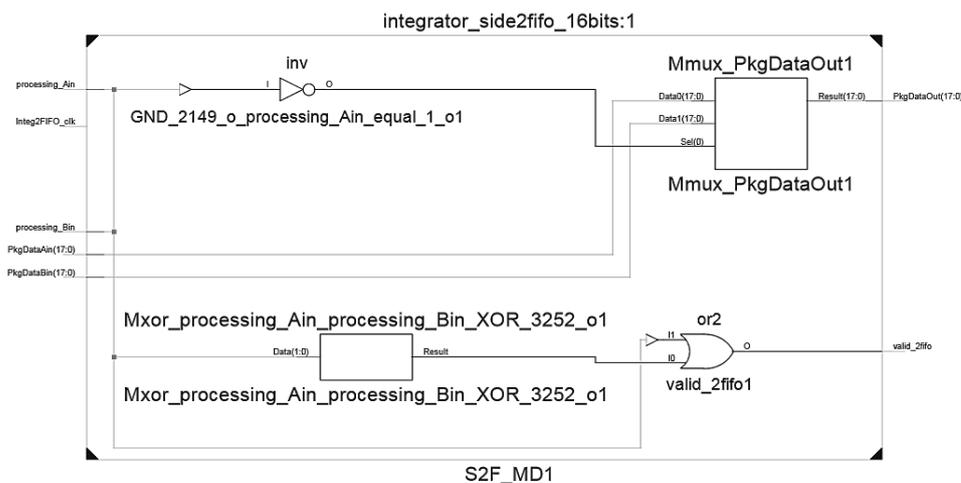


Fig. VI.19 - Esquema RTL: *S2F_MD1* para un solo E-Proc de 2 bits. Se requiere uno por mini cajón, para las pruebas opera de manera que solamente entregue la señal del lado A.

Este módulo es eliminado automáticamente durante la síntesis cuando se trabaja con dos e-procs de dos bits, una e-proc por lado.

VI. 3. 4 *elink_tx_INT2HOST* (FIFO2Elink.vhd)

Es configurado para que por su salida los datos lleguen al GBT como eprocs de 2 bits.

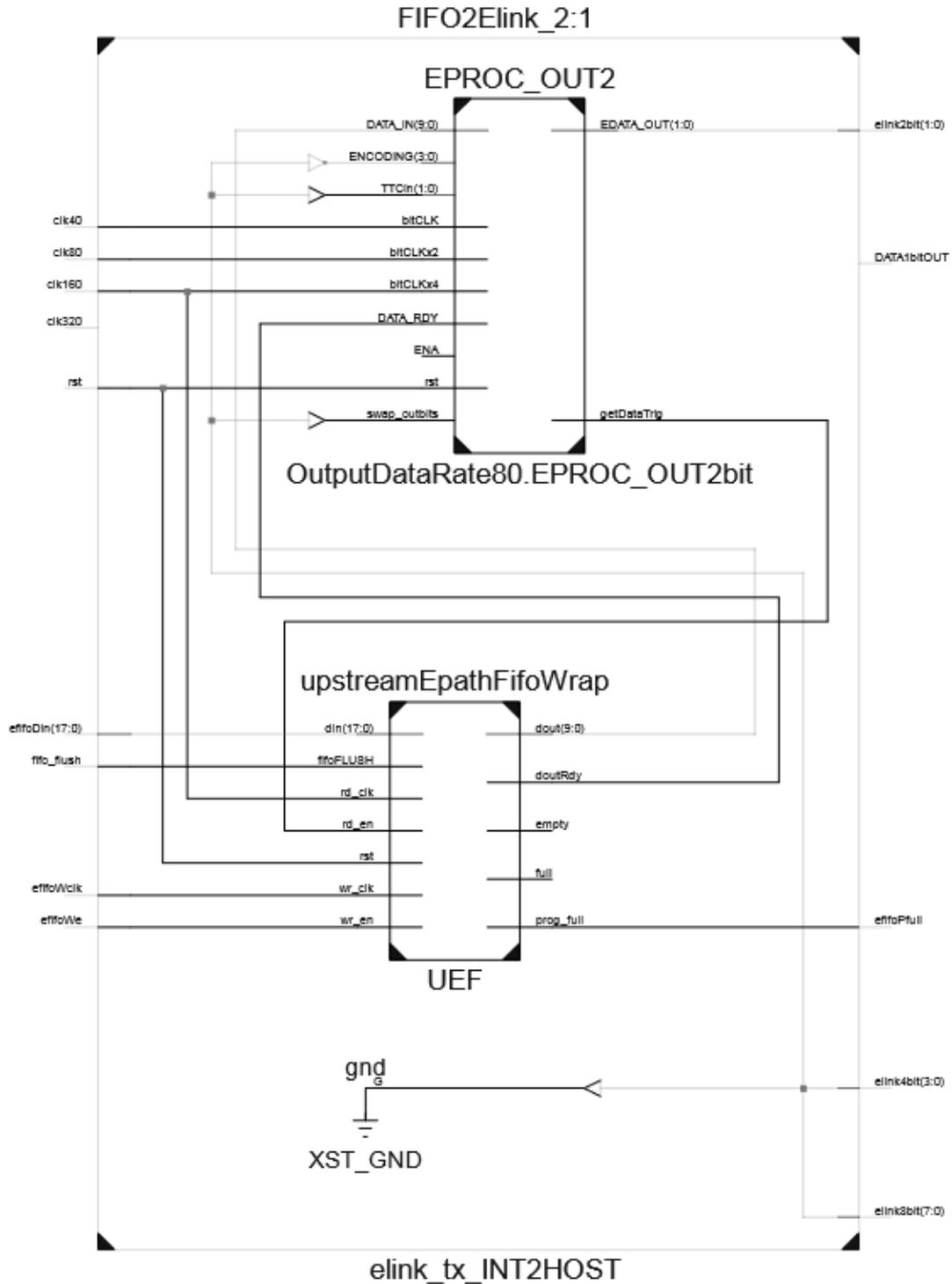


Fig. VI.20 - Esquema RTL: *elink_tx_INT2HOST*. Los bloques se obtienen al configurarlo para que su salida sean datos de 2 bits (*elink2bit*).

Se utiliza de forma independiente a los registros de *FLX_DATA_IN* destinados a ser enviados al GBT como eprocs de 8 bits. Si se utilizan dos E-procs de 2 bits, se necesita instanciar dos veces este recurso, uno por cada lado.

VI. 3. 5 FIFO2GBT (*Interface2_gbt_example_design.vhd*)

Su función es convertir la salida de 2 bits del *elink_tx_INT2HOST* (*FIFO2Elink.vhd*) en un vector de 84 bits de anchura. El tamaño de este vector depende de las pruebas que pretenda hacer el usuario.

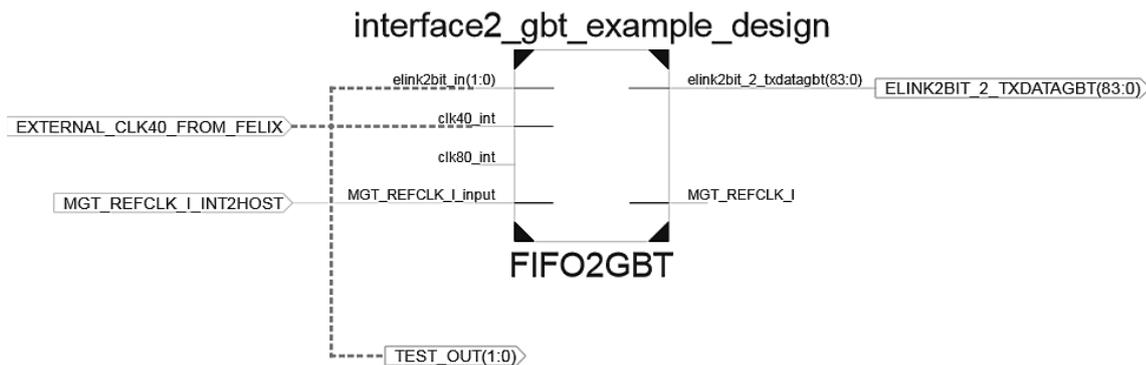


Fig. VI.21 - Esquema RTL: FIFO2GBT. La salida es un único vector de 84 bits de longitud constituido por concatenaciones de la señal de salida del *elink_tx_INT2HOST*; es independiente de las señales L1A.

VI. 4. Sistema de comunicación del FE-I4

VI. 4. 1 Propuesta de la trama de datos para el FE-I4

La trama de datos que se enviará entre ambas placas electrónicas contiene información suficiente como para distinguirla por cada canal en la que es enviada. La misma ya cuenta con su cabecera y cola, que incluyen palabras clave o *keywords* que sirven para identificar el comienzo o final de una trama en el receptor.

La trama se genera con señal de reloj de 20 MHz para serializarse directamente con la señal de reloj a 160 MHz y cuenta con el *Header 3*, para hacer la prueba de lazo donde se recupera en la KC705 la misma trama enviada a la VC707.

Cada componente de la trama de datos tiene el tamaño de un *byte*, así las palabras clave pueden tomar valores de x"00" hasta x"FF" y no deben estar presente en ninguna otra parte de la trama. Para identificar el inicio de un fragmento se elige la palabra x"CD" y para el final se utiliza x"EA". El control de la FIFO que recibe los datos descarta los primeros 2 *bytes* de la cabecera incluyendo la palabra clave, hace lo mismo para los primeros 2 *bytes* de la cola. En ningún momento debe almacenarse en la FIFO la palabra x"AA" que antecede a la palabra clave de la cabecera ni la palabra x"FF" que sigue al final de la cola.

TABLA VI.10 Trama de datos para ser enviada entre la VC707 y la KC705 a través del conector FMC

Componente del fragmento	Posiciones dentro del fragmento	Tamaño del componente	Valores que toma el componente
SOP	--	8 b / 1 B	AA
Header 1 (keyword)	127 downto 120	8 b / 1 B	CD
Header 2	119 downto 112	8 b / 1 B	AA
Header 3 (loop)	111 downto 104	8 b / 1 B	CD
Header 4	103 downto 96	8 b / 1 B	AE
Datos	95 downto 88	8 b / 1 B	No. de fragmento = 00 a 7F
	87 downto 80	8 b / 1 B	01
	79 downto 72	8 b / 1 B	02
	71 downto 64	8 b / 1 B	03
	63 downto 56	8 b / 1 B	04
	55 downto 48	8 b / 1 B	05
	47 downto 40	8 b / 1 B	06
	39 downto 32	8 b / 1 B	07
	31 downto 24	8 b / 1 B	AA
	23 downto 16	8 b / 1 B	No. de canal = 01 a 0C
Trailer 1 (keyword)	15 downto 8	8 b / 1 B	EA
Trailer 2	7 downto 0	8 b / 1 B	DC
EOP	--	8 b / 1 B	FF
Tamaño en bits del fragmento	128 bits / 16 bytes		

Los pines de las FPGA que pueden ser asignados para comunicación por el conector FMC lo hacen a través del *ISERDES2* y del *OSERDES2*, ambas herramientas que deben programarse en VHD o Verilog para la serie 7 de placas electrónicas de prueba, como la KC705 y la VC707.

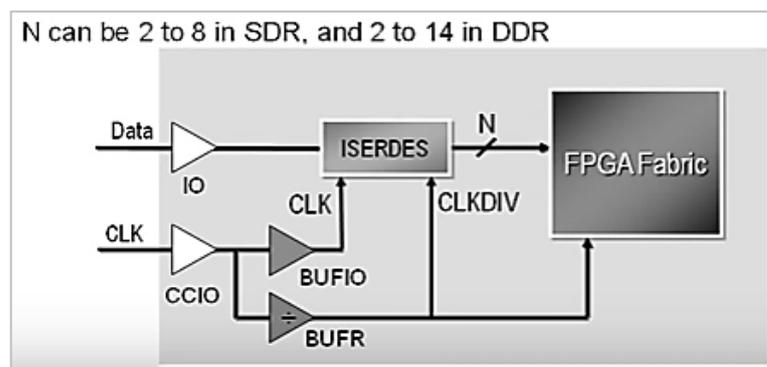


Fig. VI.22 - Esquema sobre el manejo del ISERDES según Xilinx.

VI. 4. 2 Tx y Rx del FE-I4 en la VC707 y la KC705

Como el desarrollo de este módulo hardware es en VIVADO, se aprovecha la herramienta *Add IP* para importar el bloque IP *SelectIO Interface Wizard*. Presuntamente, solo se debe configurar los parámetros que por defecto VIVADO permite modificar, sin embargo, se aprovecha el código generado automáticamente para crear un bloque IP configurable completamente con Verilog¹³ o VHD.

La necesidad de esto surge porque el diseño propuesto consistirá de múltiples canales que dependan de una sola señal de reloj, por lo que la señal de reloj de 160 MHz, pueda usarse para los *ISERDES2* de otros canales. Además, se desea dejar ya instanciadas y configurables a través de código estas herramientas para una mayor personalización en algún futuro.

El diseño hará uso de los diagramas de bloques, facilitando la modificación y comprensión para nuevos usuarios. Muchos de los bloques se crearán exclusivamente para esta aplicación, otros son proporcionados por la librería de VIVADO.

VI. 4. 3 Bloques IP de la librería de VIVADO

El módulo hardware, ya sea de la VC707 o de la KC705, se realiza prácticamente con el diagrama de bloques. Muchos de estos bloques IP están incluidos en la licencia del *Design Suite* y solo se parametrizan para cumplir las especificaciones finales. En las siguientes figuras se muestran cada uno de ellos y los parámetros más importantes que fueron ajustados.

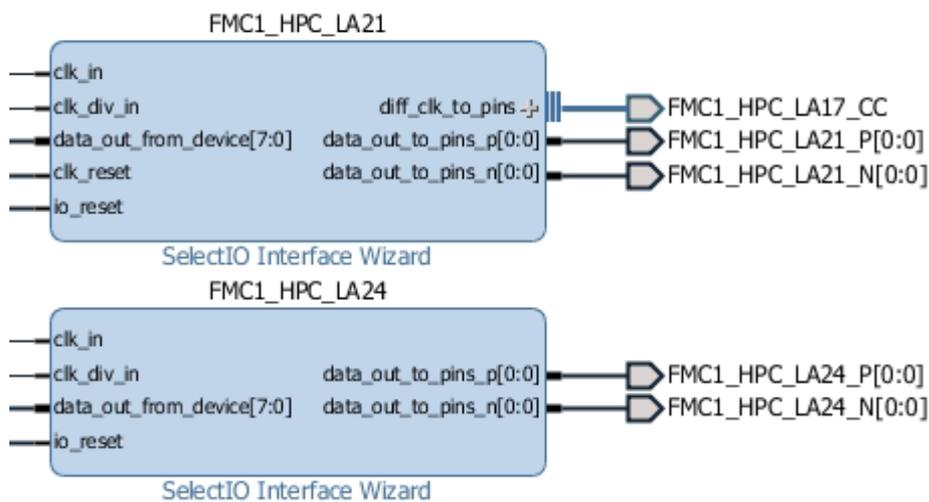


Fig. VI.23 - Bloques IP de VIVADO: *SelectIO Interface Wizard* para OSERDES2. Arriba, serializador que también reenvía la señal de reloj de 160 MHz. Abajo, serializador sin reenvío de señal de reloj. Ambos se configuran para utilizar el estándar LVDS y LVDS_25. Se usa tanto en la VC707 como en la KC705, con el cuidado de utilizar el estándar especificado para cada una.

¹³ Los archivos generados para esta interface por VIVADO están escritos en lenguaje Verilog.

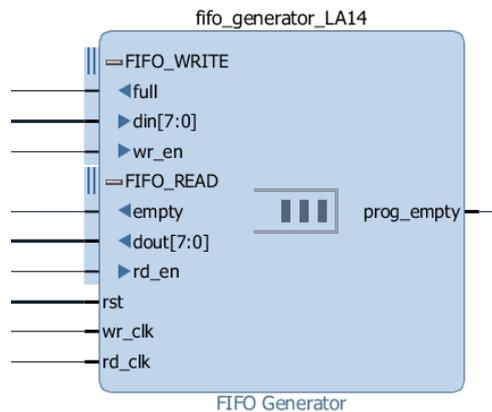


Fig. VI.24 - Bloque IP de VIVADO: FIFO Generator. Proporcionado por la librería de VIVADO, del tipo "Independent Clocks Built-in FIFO" para frecuencia nominal de escritura de 20 MHz y frecuencia de lectura de 40 MHz. La señal de vacío programable se ajusta a 14. Se usa tanto en la VC707 como en la KC705.

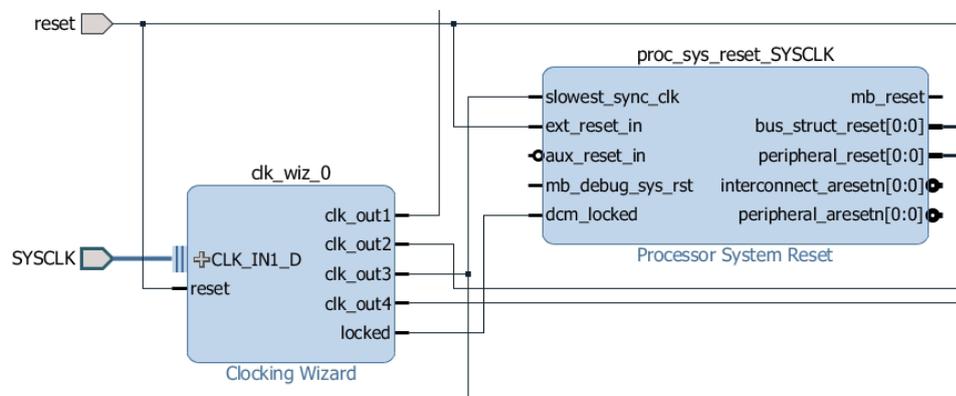


Fig. VI.25 - Bloques IP de VIVADO: Cloning Wizard y Processor System Reset. El bloque Cloning Wizard genera señales de reloj de 200 MHz, 160 MHz, 40 MHz y 20 MHz. El Processor System Reset evita los reinicios que afecten negativamente a las interfaces de entrada y salida, basándose en el reloj de frecuencia más baja de su dominio. Se usa tanto en la VC707 como en la KC705.

VI. 4. 4 Bloques IP creados en este trabajo

Estos módulos surgen sobre todo para cumplir necesidades de control, pruebas y disponibilidad de señales. Son modificables desde su código y pueden utilizarse en otros proyectos del TileCal.

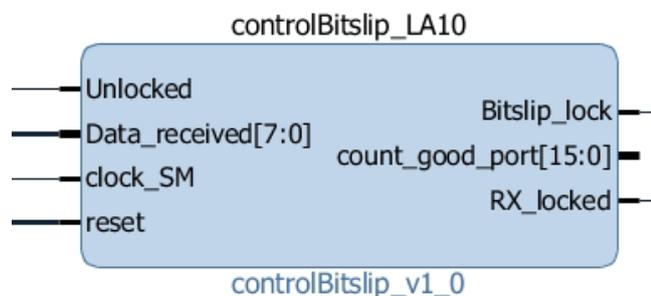


Fig. VI.26 - Bloque IP creado en este trabajo: controlBitslip_v1_0. Genera las señales bitslip para el ajuste del ISERDES2 y proporciona una señal de enganche una vez que ha alcanzado las suficientes palabras de control. Se usa tanto en la VC707 como en la KC705.

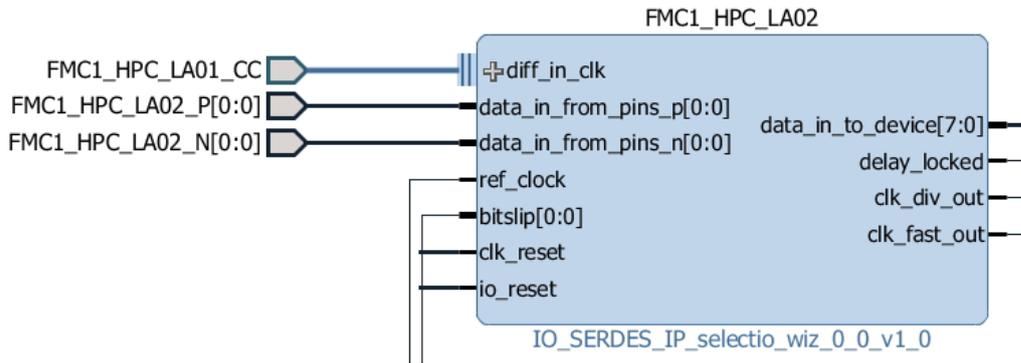


Fig. VI.27 - Bloque IP sacado del código verilog del SelectIO Interface Wizard: IO_SERDES_IP_selectio_wiz_0_v1_0. Deserializador que recibe la señal de reloj y devuelve la frecuencia dividida por el factor de serialización. La modificación consiste en sacar la señal de reloj de 160 MHz e instanciar los terminales diferenciales. Se usa tanto en la VC707 como en la KC705, con el cuidado de utilizar el estándar especificado para cada una.

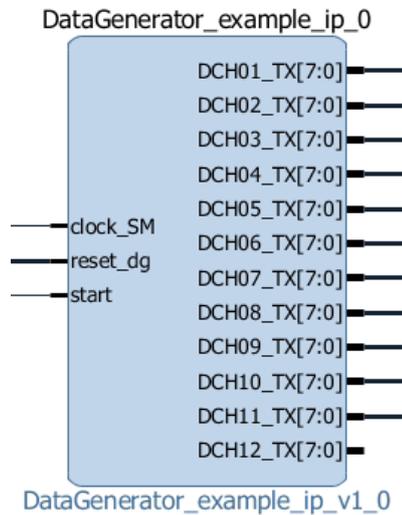


Fig. VI.28 - Bloque IP creado en este trabajo: DataGenerator_example_ip_V1_0. Depende de una señal de reloj a 20 MHz y simula doce canales de datos independientes y sincronizados a esa frecuencia. Se usa tanto en la VC707 como en la KC705.

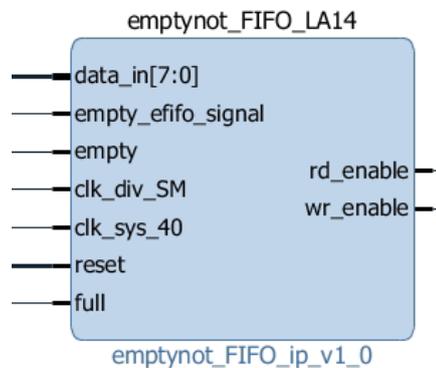


Fig. VI.29 - Bloque IP creado en este trabajo: emptynot_FIFO_ip_v1_0. Controla la escritura y lectura de las FIFO. Trabaja en dos dominios de reloj, el externo y el del sistema. Se usa tanto en la VC707 como en la KC705.

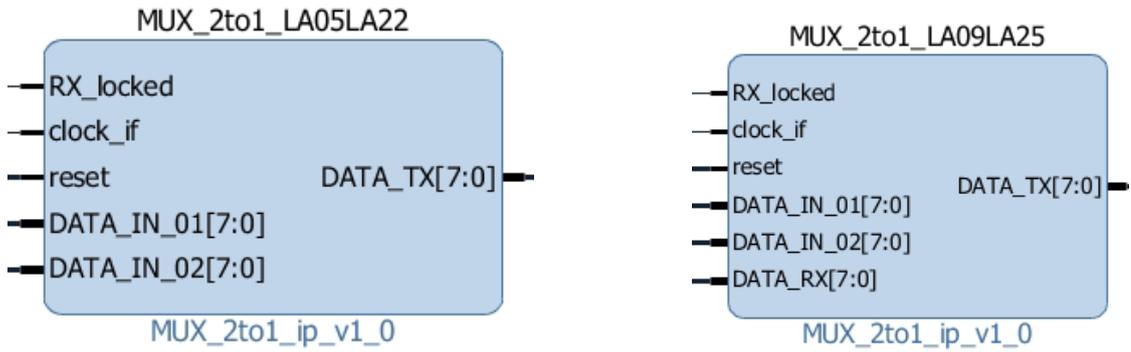


Fig. VI.30 - Bloque IP creado en este trabajo: MUX_2to1_ip_v1_0. Depende de las señales RX_locked del bloque controlBitslip_v1_0 y las palabras de control que detecte en el puerto del DATA_IN_01. Sirve para retransmitir automáticamente los datos recibidos. Su uso es para la VC707.

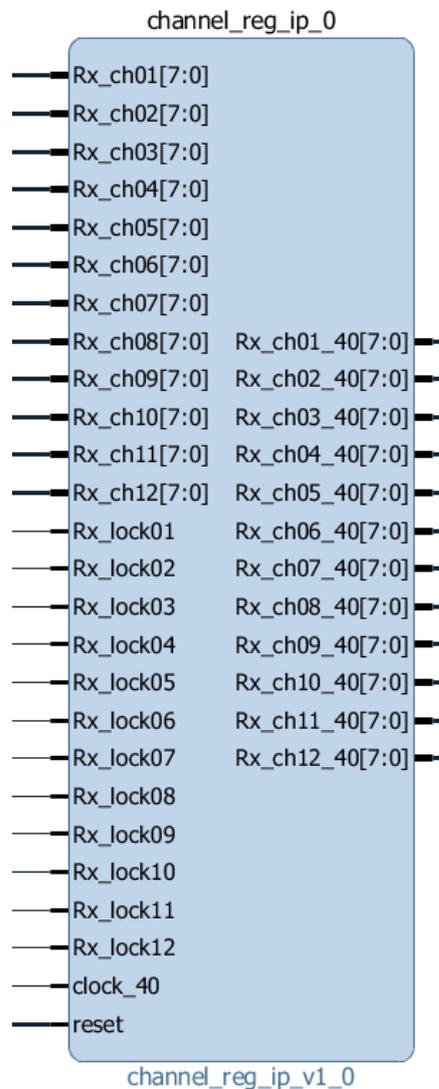


Fig. VI.31 - Bloque IP creados en este trabajo: cannel_reg_ip_v1_0. Discriminador que solo permite la salida de la señal solamente si detecta un '1' del RX_locked. Su actualización es de 40 MHz simulando las lecturas que verían los e-proc con el reloj del sistema. Se usa tanto en la VC707 como en la KC705.

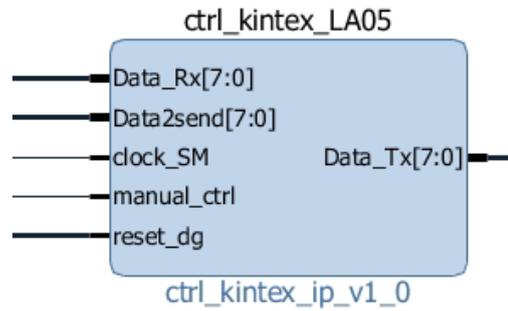


Fig. VI.32 - Bloque IP creado en este trabajo: *ctrl_kintex_ip_v1_0*. Depende de una señal de reloj a 20 MHz y es el primer paso de handshaking entre la KC705 y la VC707. Su uso es exclusivo para la KC705.

VI. 4. 5 Estandar LVDS y LVDS_25

Los bloques I/O de las dos placas electrónicas de evaluación operan con diferentes voltajes de alimentación V_{CCO} en los bloques I/O, importante cuando son terminaciones sencillas a 1.8 V o 2.5 V. Una forma de trabajar con ambos valores de tensión sin que pueda causar daño a la electrónica es con estándares de comunicación diferencial. La opción es utilizar LVDS y LVDS_25, cuyos valores de tensión son idénticos cuando se utilizan terminaciones diferenciales de 100 Ω .

TABLA VI.11 Especificaciones LVDS para los bloques HP de la placa electrónica VC707.

Symbol	DC Parameter	Conditions	Min	Typ	Max	Units
V_{CCO}	Supply voltage		1.710	1.800	1.890	V
V_{OH}	Output High voltage for Q and \bar{Q}	$R_T = 100 \Omega$ across Q and \bar{Q} signals	-	-	1.675	V
V_{OL}	Output Low voltage for Q and \bar{Q}	$R_T = 100 \Omega$ across Q and \bar{Q} signals	0.825	-	-	V
V_{ODIFF}	Differential output voltage (Q - \bar{Q}), Q = High (\bar{Q} - Q), \bar{Q} = High	$R_T = 100 \Omega$ across Q and \bar{Q} signals	247	350	600	mV
V_{OCM}	Output common-mode voltage	$R_T = 100 \Omega$ across Q and \bar{Q} signals	1.000	1.250	1.425	V
V_{IDIFF}	Differential input voltage (Q - \bar{Q}), Q = High (\bar{Q} - Q), \bar{Q} = High	Common-mode input voltage = 1.25V	100	350	600	mV
V_{ICM}	Input common-mode voltage	Differential input voltage = ± 350 mV	0.300	1.200	1.425	V

TABLA VI.12 Especificaciones LVDS_25 para los bloques HR de la placa electrónica KC705.

Symbol	DC Parameter	Conditions	Min	Typ	Max	Units
V_{CCO}	Supply voltage		2.375	2.500	2.625	V
V_{OH}	Output High voltage for Q and \bar{Q}	$R_T = 100 \Omega$ across Q and \bar{Q} signals	-	-	1.675	V
V_{OL}	Output Low voltage for Q and \bar{Q}	$R_T = 100 \Omega$ across Q and \bar{Q} signals	0.700	-	-	V
V_{ODIFF}	Differential output voltage (Q - \bar{Q}), Q = High (\bar{Q} - Q), \bar{Q} = High	$R_T = 100 \Omega$ across Q and \bar{Q} signals	247	350	600	mV
V_{OCM}	Output common-mode voltage	$R_T = 100 \Omega$ across Q and \bar{Q} signals	1.000	1.250	1.425	V
V_{IDIFF}	Differential input voltage (Q - \bar{Q}), Q = High (\bar{Q} - Q), \bar{Q} = High		100	350	600	mV
V_{ICM}	Input common-mode voltage		0.300	1.200	1.500	V

VII. NORMATIVA

La normativa aplicada en este trabajo se aplica de acuerdo a dos enfoques principales: los recursos hardware que pueden ser utilizados por las FPGA y la interfaz para poder conectar las FPGA a través de los pines I/O de sus FMC.

Algunas buenas prácticas y normas que regulan las condiciones de seguridad, impacto ambiental y otras aplicables pero alejadas de los objetivos de este trabajo no son mencionadas.

VII. 1. ANSI/VITA 57.1-2008

Esta es la principal norma que se toma en cuenta para el diseño de la PCB que permitirá conectar a la XC7K325T directamente con la XC7VX485T con ayuda de sus placas electrónicas de evaluación. Al seguir esta norma se asegura compatibilidad de la nueva placa electrónica de pruebas con las placas electrónicas de evaluación VC707 y KC705. La compatibilidad con el conector FMC del TilePPr debe revisarse antes de usar la nueva placa electrónica de pruebas.

La norma tiene por objetivo definir una familia de conectores de alta velocidad para entradas y salidas que soporten hasta 2 Gb/s en señales diferenciales, escalable para aplicaciones desde bajo hasta alto rendimiento, tamaño mínimo y disponibilidad de varias I/O [11]. Es aplicable hasta con 160 señales definidas por el usuario.

No se pretende elaborar una placa electrónica acreditable de conformidad con esta norma, aunque se puede dejar esa posibilidad para trabajos futuros. Con la baja potencia eléctrica que utilizará la PCB, muchas de las recomendaciones de la norma ANSI/VITA 57.1-2008, como aquellas que hablan sobre los mecanismos de enfriado, no son necesarias revisar.

VII. 2. Dimensiones de la PCB

La norma es muy explícita con las dimensiones a utilizar aunque muchas de las aplicaciones del TileCal han requerido un área mayor a la sugerida para las interfaces I/O, extendiendo la profundidad máxima de 31 milímetros a 42.5 milímetros. Las posiciones de los orificios y sus tamaños se respetan.

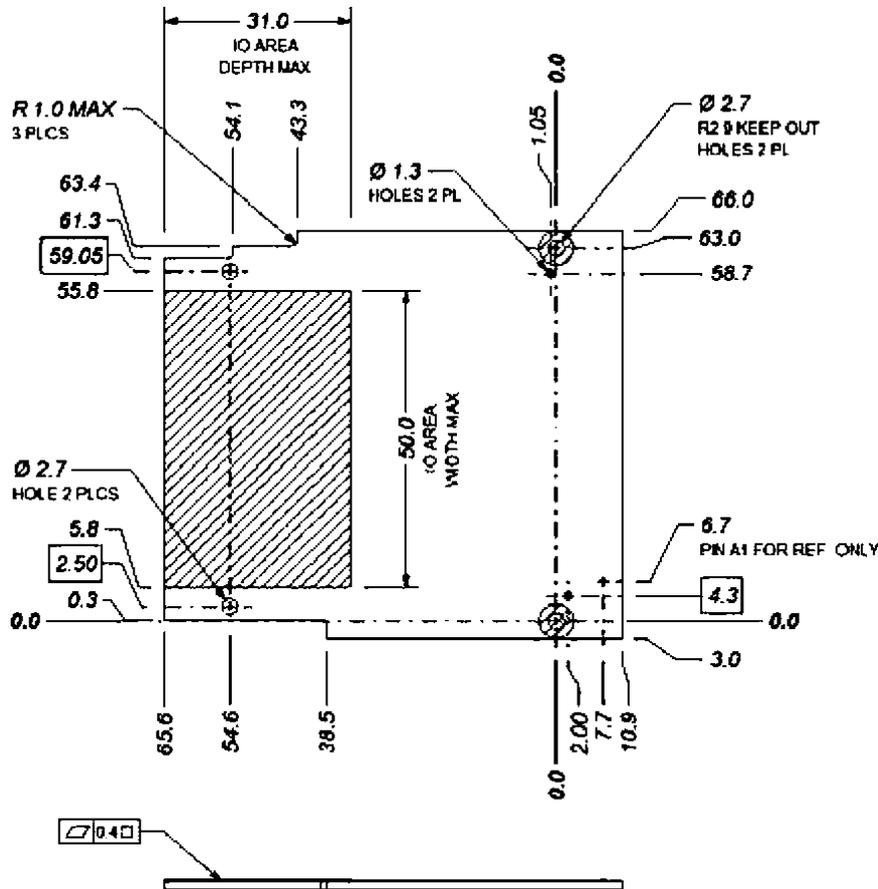


Fig. VII.1 - Mecánica del módulo FMC de anchura simple de grado comercial según ANSI/VITA 57.1-2008

La PCB también se puede dividir en tres regiones diferentes asignando al conector FMC en la región 2. De acuerdo con la norma se aceptan tres combinaciones de las regiones: regiones 1 y 2, regiones 2 y 3, regiones 1, 2 y 3. Para el uso que tendrá este trabajo, solo se necesita la combinación de las regiones 1 y 2, extendiendo la región 1 y descartando la región 3.

El módulo FMC de anchura simple es el elegido para estudio básicamente por dos razones: el TilePPr actualmente cuenta con solo un conector FMC y el otro es que la placa electrónica de evaluación VC707 necesita de uno para recibir la señal del TTC.

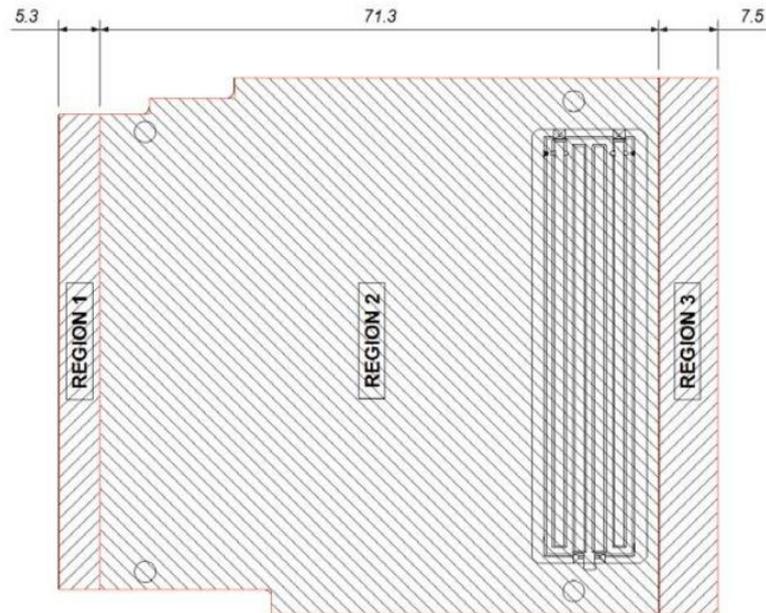


Fig. VII.2 - Regiones de la PCB para módulo FMC según VITA 57.1.

VII. 3. Interfaz FMC

Las posiciones del conector FMC se han respetado, correspondientes a lo estipulado en las reglas de la sección 3.4.4. de esta norma.

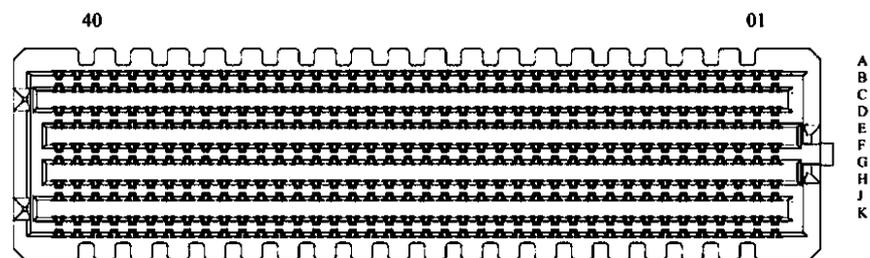


Fig. VII.3 - Etiquetado del conector del módulo FMC (vista del lado del componente).

Los conectores compatibles son listados en la sección 3.4 de la ANSI/VITA 57.1-2008, haciendo hincapié que existen dos versiones de los mismos: los *high-pin count* con 400 contactos y *low-pin count* con 160 contactos. Solamente la VC707 tiene capacidad para usar el estándar de la versión *high-pin count*. Las huellas para utilizar este conector están disponibles en internet y son compatibles con software de uso libre como KiCad.

VII. 4. Pines y señales

Respecto a la asignación de pines, se ha respetado el nombre que le ha dado la norma en el capítulo 5, en la “Tabla 2. Definición de señales para el conector de mayor conteo de pines” a cada una de ellos, tanto para el diseño de la PCB como en los archivos .xdc, programación hardware y bloques de diseño del proyecto en VIVADO.

En el caso particular de la KC705, esta no dispone de conexiones para los bloques HB, por lo que la prioridad para la mayor cantidad de canales utilizando un solo cableado comercial se le dará a las señales del bloque LA, respetando (como recomendación) las reglas 5.15, 5.16 y los permisos 5.2 sobre el uso de señales diferenciales debe proveerse una impedancia diferencial de $100 \Omega \pm 10 \%$.

Estándares de señalización LVDS y LVTTTL son los recomendados para estos dispositivos y se recomienda mantener al mínimo el sesgo o *skew*, menor al 10 % del intervalo de unidad objetivo.

TABLA VII.1 Ejemplos de tiempos UI (intervalo de unidad) para velocidades de transferencia de datos.

Targetted Data Rate	10% UI (in pico seconds)
100 Mbps	1000
200 Mbps	500
500 Mbps	200
1.0 Gbps	100
1.5 Gbps	66
2.0 Gbps	50

VIII. SIMULACIONES

Las simulaciones de la programación hardware corresponden a los resultados obtenidos de la síntesis o implementación. Solamente en el proyecto FE-I4 hizo falta de cálculos de pista, vías y líneas de transmisión.

VIII. 1. FLX.Felix

El proyecto por sí mismo requiere una gran cantidad de recursos. Como los pequeños ajustes para su correcto funcionamiento son a nivel lógico, la cantidad de recursos que toma de la FPGA luego de generar el archivo .bit es la siguiente:

Device Utilization Summary			
Slice Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	16,448	303,600	5%
Number of Slice Registers	12,446	607,200	2%
Number used as Flip Flops	12,437		
Number of RAMB36E1/FIFO36E1s	163	1,030	15%
Number using RAMB36E1 only	163		
Number of bonded IOBs	42	700	6%
Number of BUFG/BUFGCTRLs	17	32	53%

Fig. VIII.1 - Resultados de uso post-síntesis: FELIX en el VC707 con el INT2HOST para un solo E-Proc de 2 bits. La tabla es obtenida luego de la sintetización y generación del archivo .bit en ISE Project Navigator 14, en esta ocasión el INT2HOST tiene asignado su propio MMCM con una señal de reloj de entrada y tres de salida.

VIII. 2. INT2HOST

El módulo (para sus dos diseños) fue pensado para utilizarse eventualmente como un IP Core de Vivado. Su importación desde el formato de ISE no representó

mayor problema y sólo con las librerías básicas para su funcionamiento necesita los siguientes recursos de la FPGA del TilePPr.

Resource	Estimation	Available	Utilization %
LUT	850	303600	0.28
FF	1331	607200	0.22
BRAM	0.50	1030	0.05
IO	106	700	15.14
BUFG	4	32	12.50

Fig. VIII.2 - Resultados de uso post-síntesis: INT2HOST para un solo e-proc de 2 bits. La tabla es obtenida luego de la sintetización al importar el proyecto de ISE Project Navigator 14.7 a Vivado v2016.4 (64-bit).

Resource	Estimation	Available	Utilization %
LUT	1160	303600	0.38
FF	1432	607200	0.24
BRAM	1	1030	0.10
IO	25	700	3.57
BUFG	4	32	12.50

Fig. VIII.3 - Resultados de uso post-síntesis: INT2HOST para dos e-proc de 2 bits. La tabla es obtenida luego de la sintetización al importar el proyecto de ISE Project Navigator 14.7 a Vivado v2016.4 (64-bit).

Los recursos de BUFG se aprovecharán con las otras señales y buffers globales que maneja el FELIX interface (se evita malgastar los recursos MMCM / PLL). Como es de esperarse, la lógica que se ha agregado no impacta en mayor medida a los recursos de la XC7VX485T que se mencionan en las especificaciones de diseño de este trabajo.

Para simular los datos que son empaquetados por el *integrator_receiver.vhd* se pone en marcha el ChipScope con el *example_data_generator.vhd*. No se puede monitorear directamente la salida del GBT, pero es suficiente para verificar que las señales de control y e-procs están funcionando correctamente y con las frecuencias esperadas. Las señales se agregan al ChipScope y se le da una capacidad de almacenamiento de muestras lo suficientemente grande para observar toda la trama de datos proveniente del INT2HOST.

Una vez programada la tarjeta VC707 con la primera trama de datos propuesta, se hacen las respectivas lecturas con ChipScope:

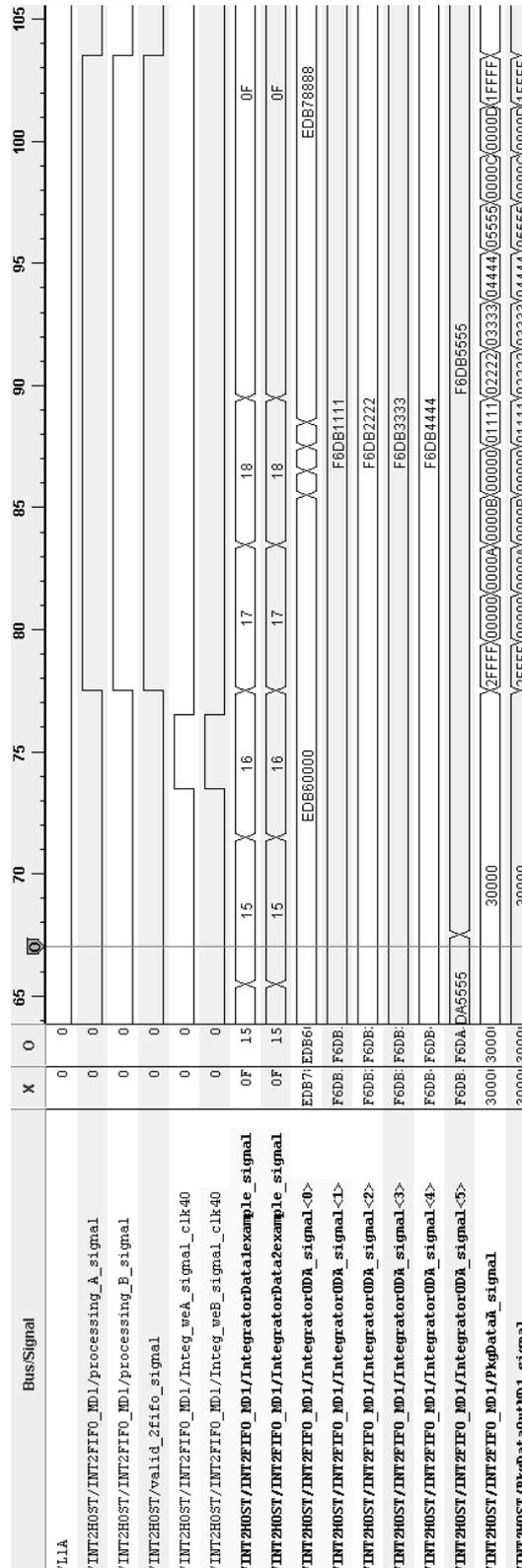


Fig. VIII.4 - Formas de onda por ChipScope capturadas: trama de datos generados por el módulo ExampleData del INT2HOST para un solo E-proc de 2.

VIII. 3. FE-I4

VIII. 3. 1 Placa electrónica para pruebas FMC-SCSI-HDMI-DisplayPort

Respecto al emisor y receptor, un mayor acercamiento de la simulación a la realidad se logra con la exportación de la programación hardware a un modelo IBIS con ayuda de las herramientas de VIVADO. Los modelos exportados son post-implementation y cuentan ya con las características de los terminales diferenciales¹⁴.

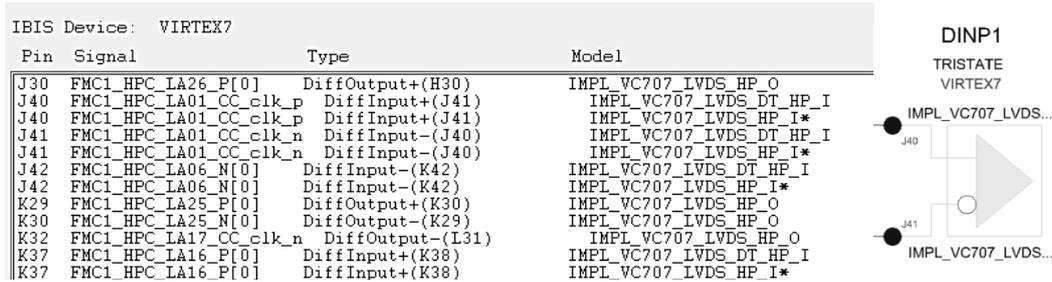


Fig. VIII.5 - Simulación con SigXplorer: modelo IBIS exportados de Vivado. Se toman los puertos IO diferenciales del tipo de modelo IbisDevice del VC707 y KC705¹⁵

Es buena práctica revisar los modelos que han sido exportados desde VIVADO según el IBIS Syntax [12] antes de realizar las simulaciones.

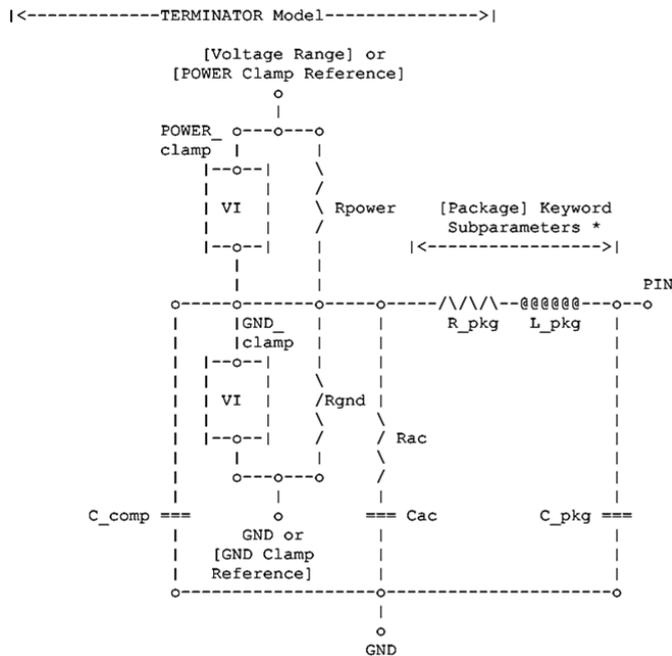


Fig. VIII.6 - Modelo IBIS del pin de entrada. Los parámetros y valores son asignados automáticamente por VIVADO cuando se exporta a IBIS después de la implementación.

¹⁴ Los terminales diferenciales son necesarios para asegurar la integridad de la señal.

¹⁵ Un pequeño bug surge de la importación del modelo XC7VX485T. Se corrige cambiando de posición la declaración del modelo C_comp del terminal diferencial según lo indica el IBIS Syntax.

```
[Model]          rterm_dt_hp
Model_type      Series
|variable      typ          min          max
C_comp         0.0000pF     0.0000pF     0.0000pF
|
[Temperature Range] 25.0000      85.0000      0.0
[Voltage Range]    1.8000V      1.7100V      1.8900V
[R Series]         88.5077      116.6164     67.0503
|
|End [Model] rterm_dt_hp|
|
[Model Selector]  LVDS_DT_HP_MS
LVDS_HP_I        Input
LVDS_DT_HP_I    Input
```

Fig. VIII.7 - Modelo IBIS del terminal diferencial en las entradas del XC7VX485T. Tomar nota del valor típico de la resistencia que se conecta entre los pines de entradas.

La placa electrónica de pruebas está pensada para manejar cuatro capas. El fabricante de PCB provee varias combinaciones posibles para facilitar el diseño que realizan sus clientes. Respecto a la simulación, se trabaja sobre la placa electrónica con la siguiente configuración:

	Subclass Name	Type	Material	Thickness (MIL)	Conductivity (mho/cm)	Dielectric Constant	Loss Tangent
1		SURFACE	AIR			1	0
2		DIELECTRIC	CONFORMAL_COAT	0.6	0	3.8	0
3	TOP	CONDUCTOR	COPPER	1.37795	595900	4	0
4		DIELECTRIC	FR-4	11.3386	0	4	0.035
5	PLANE_2	PLANE	COPPER	1.37795	595900	4	0.035
6		DIELECTRIC	FR-4	27.9528	0	4	0.035
7	PLANE_3	PLANE	COPPER	1.37795	595900	4	0.035
8		DIELECTRIC	FR-4	11.3386	0	4	0.035
9	BOTTOM	CONDUCTOR	COPPER	1.37795	595900	4	0
10		DIELECTRIC	CONFORMAL_COAT	0.6	0	3.8	0
11		SURFACE	AIR			1	0

Fig. VIII.8 - Simulación con SigXplorer: Especificación de las cuatro capas en el simulador.

Para calcular el retardo en la propagación de la línea de transmisión que simula ser el cableado [13] se emplea la ecuación:

$$t_d = \frac{d}{v_p} = (5.9 \text{ ft})(1.35 \text{ ns/ft}) = 7.97 \text{ ns} \quad \text{Ecuación (VIII.1)}$$

Las señales son de tipo diferencial y se pretende trabajar de acuerdo con esta modalidad. Las vías se consideran acopladas, cada una con un *pad* de 500 μm , taladro de 150 μm , *anti-pad* de 754 μm y los centros de las vías de cada par acoplado están separados 900 μm entre sí. Las pistas son microstrip coplanares acopladas y la longitud de cada una corresponde al diseño dado para la señal de reloj LA01_CC.

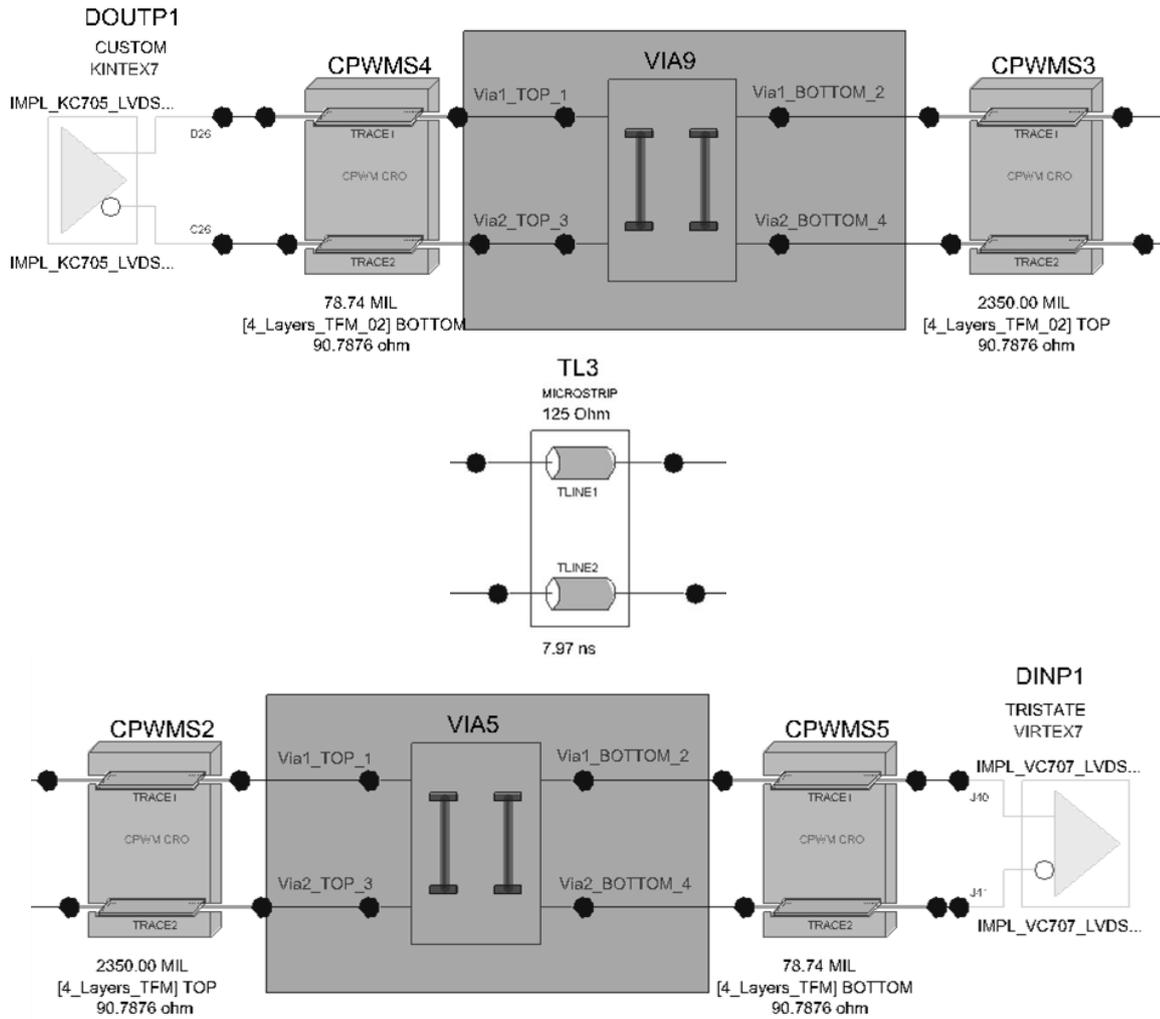


Fig. VIII.9 - Simulación con SigXplorer: Canal con la señal de reloj LA01_CC. Arriba, driver diferencial de la FPGA XC7K325T (pines D26 y C26). Al centro, modelo de cable SCSI de 1,8 metros, según los datos ofrecidos por la página de Hitachi-Cable. Abajo, receptor con su terminal diferencial de 100 Ω de la FPGA XC7VX485T (pines J40 y J1).

Driver	Receiver	GlitchTol [ns]	EyeHeight [mV]	EyeJitter [ns]	EyeWidth [ns]	Glitch	Monotonic	NoiseMargin [mV]
DESIGN.DOUTP1.D26	DESIGN.DINP1.J40	0.01982	558.139	0.381555	5.86845	PASS	PASS	160.799
DESIGN.DOUTP1.C26	DESIGN.DINP1.J41	0.01982	0.354076	5.78037	0.469628	PASS	PASS	NA
DESIGN.DOUTP1.D26_DESIGN.DOUTP1.C26_diff	DESIGN.DINP1.J40_DESIGN.DINP1.J41_diff	0.01982	881.761	0.110064	6.13994	PASS	PASS	310.135

Driver	Receiver	OvershootHigh [mV]	OvershootLow [mV]	PropDelay [ns]	SwitchDelay [ns]	SettleDelay [ns]
DESIGN.DOUTP1.D26	DESIGN.DINP1.J40	1765.07	950.512	1e+041	9.54885	10.0282
DESIGN.DOUTP1.C26	DESIGN.DINP1.J41	NA	NA	1e+041	NA	NA
DESIGN.DOUTP1.D26_DESIGN.DOUTP1.C26_diff	DESIGN.DINP1.J40_DESIGN.DINP1.J41_diff	542.56	-456.689	1e+041	9.66799	9.84716

Fig. VIII.10 - Simulación con SigXplorer: Tabla resumen con los resultados de la simulación.

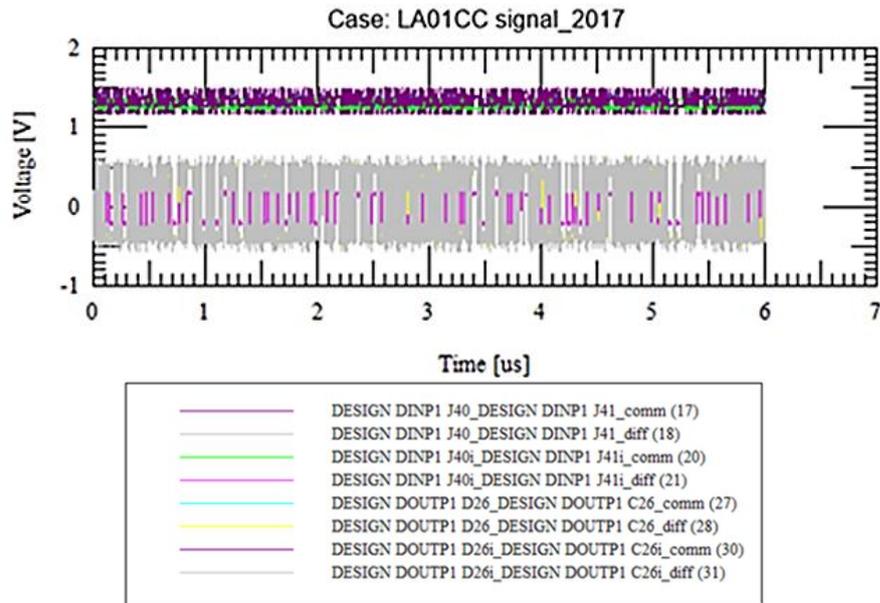


Fig. VIII.11 - Simulación con SigXplorer: Gráficas con los valores de tensión en modo común y diferencial al generar una secuencia pseudo aleatoria de 1024 bits.

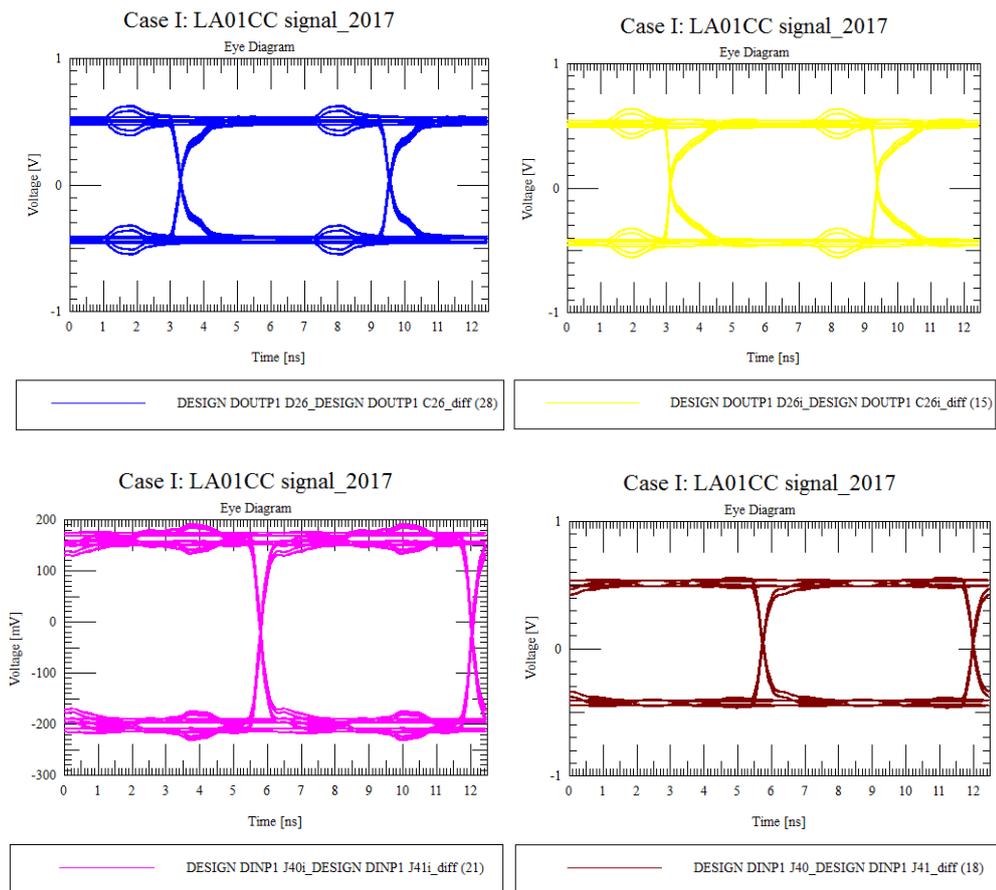


Fig. VIII.12 - Simulación con SigXplorer: Diagramas de ojo para el estudio de la señal de reloj LA01_CC. Frecuencia de reloj 160 MHz, 2 diagramas de ojo por gráfica, offset de reloj 16,5 ns e inicio de reloj 0 ns; todas las gráficas comparten la misma configuración para apreciar el desfase entre el driver y el receptor.

Sobre la programación hardware, se ve que son pocos los recursos que este diseño utiliza del sistema.

Todos los valores mostrados deben tomarse con el detalle que está siempre incluido un ILA de al menos doce entradas, con una señal de reloj de 40 MHz para muestreo y una profundidad de 4096 muestras para cada entrada. También debe considerarse que el MMCM será reemplazado por las señales de reloj que proporciona FLX.Felix cuando se utilice en el TilePPr.

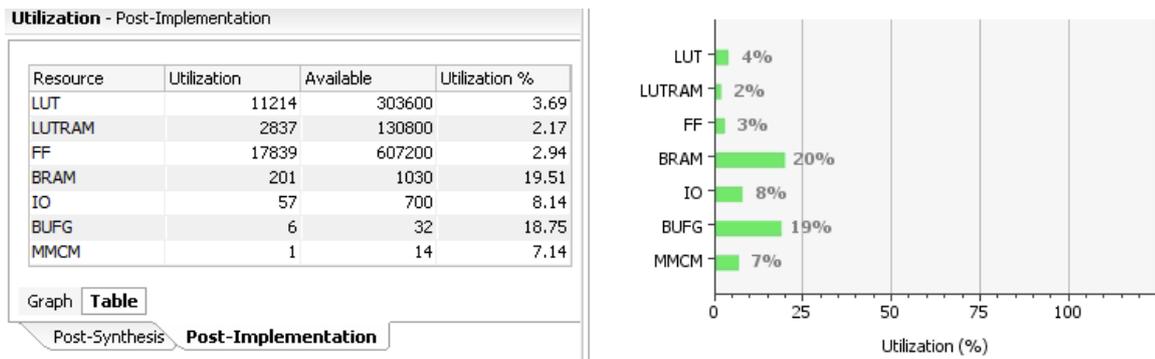


Fig. VIII.13 - Resultados de uso post-implementación: FE-I4 en la VC707. Los resultados incluyen un ILA de 36 sondas con una profundidad de almacenado de 4096 muestras cada una. Tiene la misma cantidad de FIFO built in que la KC705.

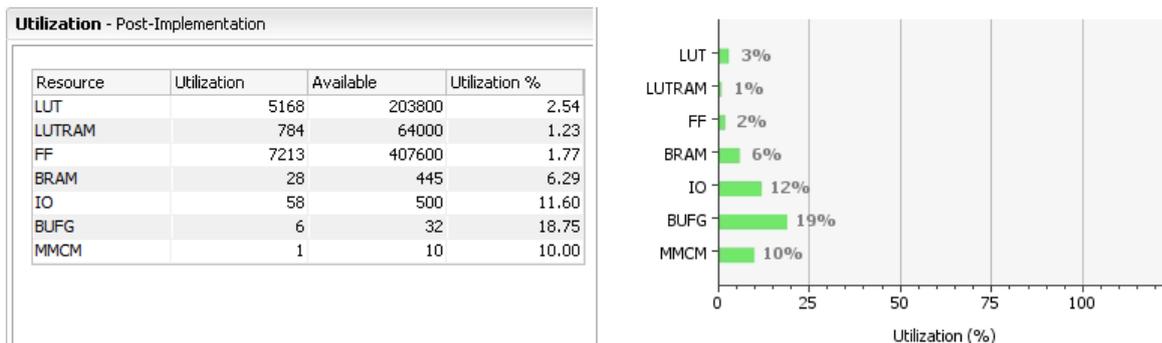


Fig. VIII.14 - Resultados de uso post-implementación: FE-I4 en la KC705. Los resultados incluyen un ILA de 12 sondas con una profundidad de almacenado de 4096 muestras cada una. Tiene la misma cantidad de FIFO built in que la VC707.

IX. ELECCIÓN DE COMPONENTES Y MOTIVACIÓN

IX. 1. FLX.Felix

En la programación de la FPGA se utilizan máquinas de estado en la fabricación de la trama de datos. La sincronización es vital en este trabajo, por lo que se puede llevar un mejor recuento y sucesión de eventos con máquinas de estado que dependan de cierto reloj.

Los demás componentes del FELIX como el SFP, el limpiador de jitter, los latiguillos y demás equipamiento fueron proporcionados previamente por el laboratorio TileCal de IFIC-Valencia y su justificación, además de lo explicado en la introducción, puede encontrarse en la bibliografía citada sobre el TileCal y el CERN.

TABLA IX.1 Especificaciones de relojes globales y PLL/MMCM en FPGA serie 7 de Xilinx.

FPGA	Speed Grade	Global Clock Network Maximum (BUFG)	I/O Clock Network Maximum (BUFIO)	Regional Clock Network Maximum (BUFR)	Horizontal Clock Network Maximum (BUFH)
Artix@-7	-1	464 MHz	600 MHz	315 MHz	464 MHz
Artix-7	-2	550 MHz	680 MHz	375 MHz	550 MHz
Kintex@-7	-1	625 MHz	710 MHz	450 MHz	625 MHz
Kintex-7	-2	710 MHz	800 MHz	540 MHz	710 MHz
Virtex@-7	-1	625 MHz	710 MHz	450 MHz	625 MHz
Virtex-7	-2	710 MHz	800 MHz	540 MHz	710 MHz

IX. 2. INT2HOST

El empaquetado en este caso también depende de máquinas de estado, pero también se depende de multiplexores y un divisor de frecuencia. El integrador trabaja con procesos a 80 MHz, síncronos con los 40 MHz de FELIX.

IX. 3. FE-I4

La interfaz FMC permite conectarse directamente con la FPGA a través de tarjetas electrónicas hechas a la medida que cumplan con la normativa correspondiente. Se trabaja directamente con la lógica programable y con los distintos bloques de interfaz de entrada y salida, haciéndola muy versátil y altamente configurable por el usuario.

Se elige la interfaz SCSI por la cantidad de canales que pueden utilizarse simultáneamente con un solo cable. Tanto la interfaz HDMI como el DisplayPort fueron pensadas para una mayor velocidad, aunque por su estructura solamente se pueden aprovechar pocos canales por un solo cable (solamente tres canales son diferenciales). Ante la posibilidad de elaborar proyectos que requieran una mayor velocidad en la transferencia de datos, se puede utilizar este mismo componente para hacer algunas pruebas, haciendo ajustes en los bloques de programación hardware.

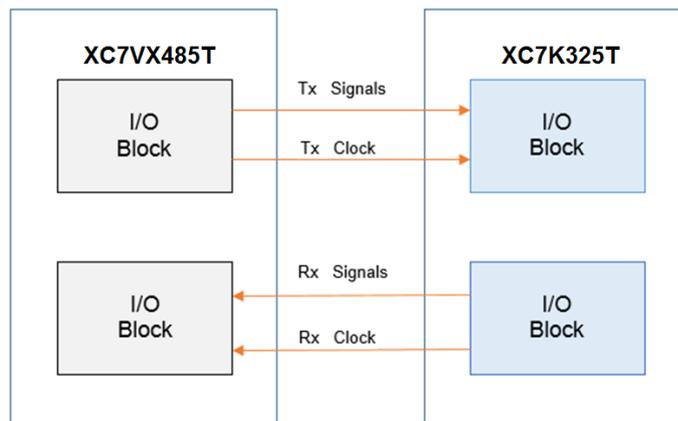


Fig. IX.1 - Estrategia de bloques de entrada y salida para aprovechar el mínimo disponible de señales de reloj. El uso de Tx y Rx mostrado es el previsto para la XC7VX485T.

IX. 4. Interfaces

El conector FMC puede adquirirse fácilmente con distribuidores de componentes electrónicos, pero su uso no es tan común como el HDMI o DisplayPort, que toleran grandes velocidades de transferencia y pueden proveer más de un conductor que puede aprovecharse por cada canal. En el caso del HDMI y DisplayPort, ya se disponen de pares pensados especialmente para señales diferenciales.

No sólo son accesibles, sino que estas interfaces difícilmente saldrían del mercado en los próximos años, asegurando disponibilidad de soporte por más años. Al ser tan populares comercialmente, hay mucha información en internet y recursos para su uso, como su huella en KiCad y en otros formatos.

El HDMI 2.0 dispone de 14.4 Gbps de ancho de banda, el DisplayPort de 17.28 Gbps y en general los SCSI de 5.12 Gbps (anchura de 16 bits, 160 MHz reloj

DDR). El ancho de banda es suficiente para las especificaciones del FE-I4, siempre que se respeten las distancias recomendadas por los fabricantes.

IX. 5. Terminales diferenciales programables

El nivel de tensión que maneja la FPGA de la VC707 para la interfaz FMC es de 1,8 V (LVCMOS18), mientras que en la KC705 es de 2,5 V (LVCMOS25). De utilizarse señales independientes se dañaría la FPGA del VC707 al someterla a un valor de tensión tan alto, por lo que la mejor salida es utilizar señales diferenciales.

Según el fabricante, pueden utilizarse estándares de señal diferencial en las interfaces LA, HA y HB del FMC siempre y cuando coincidan con el nivel de tensión asignado al bloque I/O de la señal individual. Gracias a esta condición, podemos comunicarnos entre la FPGA XC7VX485T y la XC7K325T. Obviamente, con pequeñas modificaciones los IP que se utilicen para esta parte podrán comunicar entre dos FPGA del mismo modelo siempre y cuando respeten el estándar LVDS y LVDS_25.

Con las FPGA que se van a trabajar y respecto a la integridad de la señal, el estándar a utilizar y el tipo de conector de alta eficiencia HP (High Performance) da la opción de utilizar o la terminación diferencial FD_100 para el que será conductor (driver) de la señal.

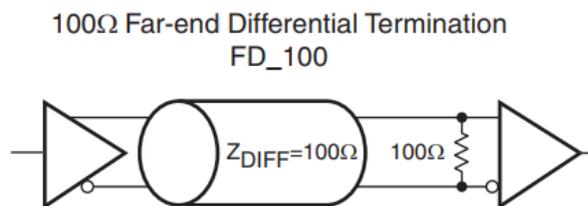


Fig. IX.2 - Terminación diferencial FD_100 declarada por defecto para estándar LVDS o LVDS_25 en modo conductor driver).

Si el puerto I/O actúa como receptor, dispone de unos terminales diferenciales programables que pueden utilizarse para alcanzar la integridad de la señal. Estos terminales se declaran en la programación hardware, en el IBUFDS o IBUFGDS [13] que se utilice.

Gracias a estos componentes es posible no incluir en la placa electrónica de pruebas FMC los terminales a los puertos diferenciales. Para posteriores diseños, esto puede cambiar sin hacer muchos cambios en los IP Core creados.

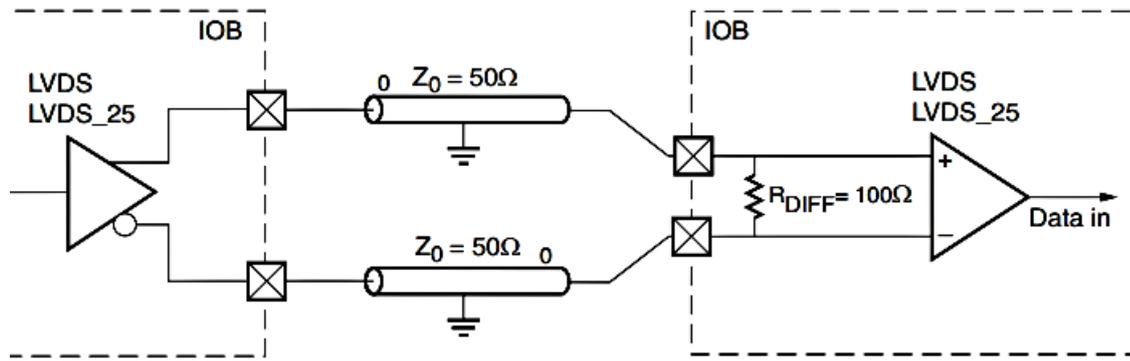


Fig. IX.3 - Terminación diferencial declarada con DIFF_TERM para estándar LVDS o LVDS_25 en modo entrada.

IX. 6. Bloques IO de la FPGA

La disposición de pines para la interfaz FMC se aprovecha con el menor uso de señales de reloj (SRCC o MRCC) siguiendo una estrategia de usar una señal de reloj por bloque y para una única dirección de transferencia. Como la interfaz que tiene la mayor cantidad de conductores es la SCSI de 68 pines, escogemos los bloques disponibles LA para asignar Tx y Rx para cada placa electrónica.

TABLA IX.2 Agrupación por bloques I/O de pines de FPGA para la interfaz SCSI: Rx del VC707

Transmisión desde la tarjeta KC705 hacia VC707		
Señal	XC7K325T (Bloque 16)	XC7VX485T (Bloque 19)
LA01_CC_P	D26	J40
LA02_P	H24	P41
LA03_P	H26	M42
LA04_P	G28	H40
LA05_P	G29	M41
LA06_P	H30	K42
LA07_P	E28	G41
LA09_P	B30	R42
LA10_P	D29	N38
LA12_P	C29	R40
LA14_P	B28	N39
LA15_P	C24	M36
LA16_P	B27	K37

TABLA IX.3 Agrupación por bloques I/O de pines de FPGA para la interfaz SCSI: Tx del VC707

Transmisión desde la tarjeta VC707 hacia KC705		
Señal	XC7K325T (Bloque 17)	XC7VX485T (Bloque 34)
LA17_CC_P	F20	L31
LA19_P	G18	W30
LA20_P	E19	Y29
LA21_P	A20	N28
LA22_P	C20	R28
LA23_P	B22	P30
LA24_P	A16	R30
LA25_P	G17	K29
LA26_P	B18	J30
LA28_P	D16	L29
LA29_P	C17	T29
LA30_P	D22	V30
LA31_P	G22	M28

TABLA IX.4 Asignación de canales por SCSI para la placa electrónica VC707

Canal	Tx / Rx	Señal
Canal 1	Tx	LA02
	Rx	LA21
Canal 2	Tx	LA03
	Rx	LA20
Canal 3	Tx	LA04
	Rx	LA19
Canal 4	Tx	LA05
	Rx	LA22
Canal 5	Tx	LA06
	Rx	LA23
Canal 6	Tx	LA07
	Rx	LA24
Canal 7	Tx	LA09
	Rx	LA25
Canal 8	Tx	LA10
	Rx	LA26
Canal 9	Tx	LA12
	Rx	LA28
Canal 10	Tx	LA14
	Rx	LA29
Canal 11	Tx	LA15
	Rx	LA30
Canal 12	Tx	LA16
	Rx	LA31
Señal de reloj	Tx	LA01_CC
	Rx	LA17_CC

TABLA IX.5 Agrupación por bloques I/O de pines de FPGA para la interfaz HDMI: Rx del VC707

Transmisión desde la tarjeta KC705 hacia VC707		
Señal	XC7K325T (Bloque 18)	XC7VX485T (Bloque 35)
HA00_CC_P	D12	E34
HA02_P	D11	E33
HA04_P	F11	F34
HA05_P	F15	G32
HA08_P	E14	J36

TABLA IX.6 Agrupación por bloques I/O de pines de FPGA para la interfaz HDMI: Tx del VC707

Transmisión desde la tarjeta VC707 hacia KC705		
Señal	XC7K325T (Bloque 18)	XC7VX485T (Bloque 35)
HA01_CC_P	H14	D35
HA09_P	F12	E32
HA18_P	K14	F39
HA21_P	J11	D37
HA22_P	L11	F36

TABLA IX.7 Asignación de canales por HDMI para la placa electrónica VC707

Canal	Tx / Rx	Señal
Canal 1	Tx	HA09
	Rx	HA02
Canal 2	Tx	HA18
	RX	HA04
Canal 3	Tx	HA21
	Rx	HA05
Canal 4	Tx	HA22
	Rx	HA08
Señal de reloj	Tx	HA01_CC
	Rx	HA00_CC

El funcionamiento entre un conector y otro no requiere de nuevos bloques IP en VIVADO, ya que solo cambia el medio y la administración de las señales entre los canales. El conector DisplayPort se ha reservado para establecer comunicación entre dos VC707, pues en la placa electrónica de evaluación el conector FMC es HPC y la KC705 solo cuenta con FMC LPC.

TABLA IX.8 Agrupación por bloques I/O de pines de FPGA para la interfaz DisplayPort: Rx del VC707

Transmisión desde la tarjeta VC707 hacia VC707		
Señal	XC7VX485T (Bloque 36)	XC7VX485T (Bloque 36)
HB06_CC_P	K23	K23
HB10_P	M22	M22
HB11_P	K22	K22

TABLA IX.9 Agrupación por bloques I/O de pines de FPGA para la interfaz DisplayPort: Tx del VC707

Transmisión desde la tarjeta VC707 hacia VC707		
Señal	XC7VX485T (Bloque 36)	XC7VX485T (Bloque 36)
HB17_CC_P	M24	M24
HB14_P	J21	J21
HB15_P	M21	M21

TABLA IX.10 Asignación de canales por DisplayPort para la placa electrónica VC707

Canal	Tx / Rx	Señal
Canal 1	Tx	HB14
	Rx	HB10
Canal 2	Tx	HB15
	RX	HB11
Señal de reloj	Tx	HB17_CC
	Rx	HB06_CC

IX. 7. Reloj y los IOSERDES2

El sistema FELIX depende de la sincronía con el reloj de 40 MHz proporcionado por el TTCex, con la ayuda de fibra óptica todas las placas electrónicas reciben la misma señal de reloj con un desfase despreciable¹⁶. Para asegurar el correcto funcionamiento de los IP Core en esta parte, se dejará por fuera la conexión del reloj TTCex con la premisa de que si el sistema responde adecuadamente con relojes diferentes, utilizar el mismo no marcará diferencia.

De manera que sea posible probar efectivamente cada interfaz independientemente, se asigna los pines de las FPGA para que cada una tenga al menos una señal de reloj.

¹⁶ Esto depende de la frecuencia de trabajo. Un desfase de unos cuantos picosegundos tiene un menor impacto en señales de 160 MHz que en señales de 10 GHz.

TABLA IX.11 Conectores asignados para uso de señales diferenciales de reloj

Conector en la FMC	Número / tipo de bloque IO en la FPGA XC7VX485T	Número / tipo de bloque IO en la FPGA XC7K325T	Interfaz en la placa de pruebas
LA01_CC	19 / HP	16 / HR	SCSI (U1)
LA17_CC	34 / HP	17 / HR	SCSI (U1)
HA00_CC	35 / HP	18 / HR	HDMI (J3)
HA01_CC	35 / HP	18 / HR	HDMI (J3)
HB00_CC	36 / HP	--	HDMI (J6)
HB06_CC	36 / HP	--	DisplayPort (J7)
HB07_CC	36 / HP	--	DisplayPort (J7)

Para serializar y deserializar se aprovecha la herramienta de Vivado IO Interface Wizard, un IP Core que llama al uso el ISERDES2 y el OSERDES2 en tipo NETWORKING para las FPGA de la serie 7.

Aunque esta herramienta presume su uso inmediato y de fácil configuración, la programación de los terminales diferenciales así como los buffers que servirán para manipular señales de reloj es mucho más fácil con un IP core completamente modificable. Ciertas normas deben mantenerse para el correcto funcionamiento:

Para el OSERDES2 solo se permiten arreglos de reloj del puerto CLK por BUFIO y de CLKDIV por BUFR. Otra opción es CLK y CLKDIV por una salida de reloj CLKOUT [0:6] del mismo MMCM o PLL. El IP Core diseñado ya tendrá incorporado el IDELAY para solo ajustar los taps según el reloj de referencia, por lo que es la configuración de BUFIO y BUFR la que se usa en el trabajo.

Para el ISERDES2 las condiciones son las mismas, solo que el BUFR incluye la función BUFR_DIVIDE (valor 8) pues debe ser el mismo reloj el que debe proporcionar la frecuencia más rápida y la más lenta.

IX. 8. Elección de FIFO

Con la señal de reloj recibida de la placa electrónica externa, los procesos que dependen de ella trabajan en un dominio de reloj diferente al del sistema. La programación hardware debe hacer propia la trama de datos que recibe pasándola a su dominio de reloj e incluso a una frecuencia notablemente diferente, por lo que se utilizan FIFO con dominios de reloj independientes.

Particularmente son dos las opciones de FIFO que se obtienen por bloques IP en VIVADO y que cuentan con la mayor cantidad de señales de control y funciones adicionales. La memoria de tipo *Built in* tiene en general los mismos beneficios que la *Block RAM*, y según las especificaciones de ambas FPGA se pueden disponer de ellas sin mayor problema. No se han habilitado las funcionalidades *ECC Support* ni *First Word Fall Through* aunque quedan disponibles para su habilitación por utilizarse las FIFO *Built in* en este trabajo.

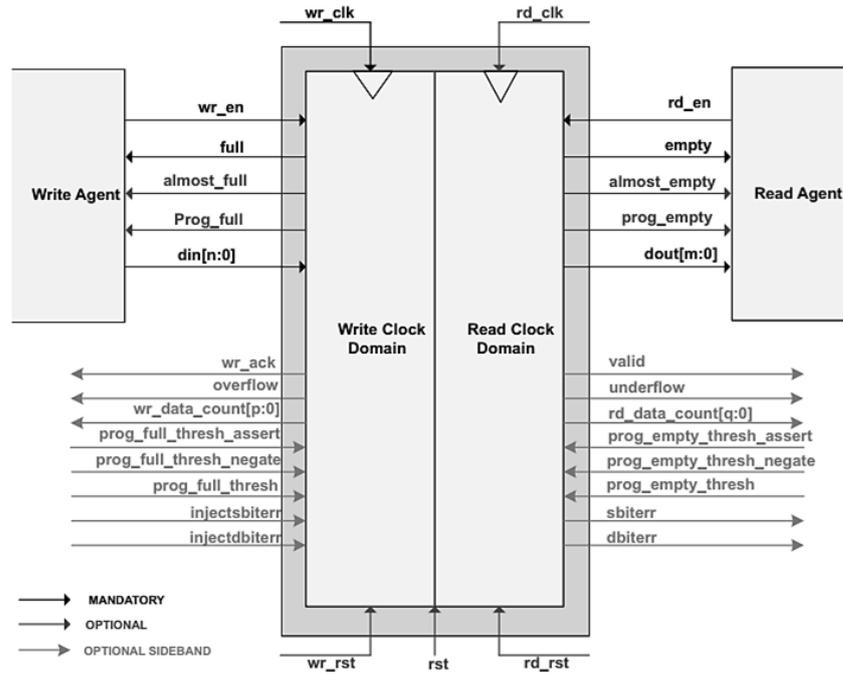


Fig. IX.4 - Diagrama de señales de un FIFO con interfaz nativa. Nótese la separación de los dominios de reloj.

TABLA IX.12 Beneficios en la configuración de memoria según la FIFO.

	Independent Clocks	Common Clock	Small Buffering	Medium-Large Buffering	High Performance	Minimal Resources
Built-in FIFO	✓	✓		✓	✓	✓
Block RAM	✓	✓		✓	✓	✓
Shift Register		✓	✓		✓	
Distributed RAM	✓	✓	✓		✓	

TABLA IX.13 Configuraciones que se pueden utilizar para las FIFO con dominios de reloj independientes.

Clock Domain	Memory Type	Non-symmetric Aspect Ratios	First-word Fall-Through	ECC Support	Embedded Register Support
Independent	Block RAM	✓	✓	✓	✓
Independent	Distributed RAM		✓		
Independent	Built-in FIFO	✓	✓	✓	✓

X. INTERCONEXIÓN Y SOFTWARE

X. 1. Conexiones existentes antes del proyecto

La interconexión de las placas de prueba y otros periféricos se explica de manera funcional. Las conexiones físicas existentes son:

- Interconexión entre los puertos PRI REF P, PRI REF N del dispositivo CDCE62005 y los puertos USER CLK P, USER CLK N de la placa PPr: Dos latiguillos coaxiales M17/113-RG316 MIL-DTL-17, 600 mm, SMA macho a SMA macho, Impedancia nominal 50 Ω , retraso 4,80 ns/m. Esta conexión es la que porta la señal de reloj de 160 MHz.
- Interconexión entre los puertos YP0, YN0 del dispositivo CDCE62005 y los puertos MGT CLK P, MGT CLK N de la placa PPr: Dos latiguillos coaxiales M17/113-RG316 MIL-DTL-17, 300 mm, SMA macho a SMA macho, impedancia nominal 50 Ω , retraso 4,80 ns/m. Esta conexión es la que porta la señal de reloj de 120 MHz.
- Interconexión entre los puertos SFP de la placa PPr y la placa del servidor local: Conectores FTRJ8519P1BNL – SFP (mini-GBIC) módulo transceptor – Gigabit Ethernet, 2 Gb Canal fibra (onda corta). Un transceptor en cada placa.
Cable Patch de Fibra Dúplex Multimodo 50/125 LSZH.
- Interconexión entre la placa TTC_FMC del TilePPr y la TTCex
Fibra óptica 50/125, 240 m.

X. 2. Programas

Programas software que son necesarios y permitieron la culminación de este trabajo:

- a) Programas comerciales instalados en la computadora portátil
 - Sistema operativo Windows 8.1 Pro de 64 bits.
 - ISE Design Suite 14.7
 - ChipScope Pro Analyzer, versión 14.7 P.20131013
 - iMPACT. Versión 14.7 (nt64) P.20131013
 - CDCE62005 Control GUI, versión 1.4.8

- b) Programas comerciales instalados en la computadora de escritorio
 - Sistema operativo: Kernel Linux 2.6.32-696.1.1el6.x86_64, GNOME 2.28.2
 - Python 2.6.6 con sus librerías gtk
 - Glade 3.8.5

- c) Programas instalados en el servidor de laboratorio IFIC TileCal
 - Sistema operativo Windows 8.1 Pro de 64 bits.
 - Vivado 2016.4
 - ISE Design Suite 14.7

- d) Programas de acceso gratuito
 - KiCad versión 4.0.6, por KiCad Developers Team.
 - LibreCAD versión 2.1.3.
 - Cadence 16.6.

- e) Programas desarrollados y proporcionados por el CERN o por el laboratorio TileCal del IFIC-Valencia
 - setup.sh (para la ejecución del elinkconfig, fdaq, fcheck, etc.)

- f) Programas desarrollados para el proyecto:
 - Unboxing v.0.4
 - Unboxing v.1.1

XI. DISEÑO DE LA PLACA DE PRUEBAS

XI. 1. Condiciones de diseño eléctrico

Para comenzar con el diseño un dato de utilidad es la longitud de onda a la frecuencia de 160 MHz asumiendo la velocidad de propagación ideal. Otros valores pueden calcularse a partir de este valor, aún considerando que la velocidad de propagación no es ideal.

$$\lambda = \frac{v_p}{f} = \frac{3 \cdot 10^8}{160 \cdot 10^6} = 1.875 \text{ m} \approx 1.88 \text{ m} \quad \text{Ecuación (XI.1)}$$

$$d_1 = \frac{\lambda}{2\pi} = \frac{1.88 \text{ m}}{2\pi} \approx 0.30 \text{ m} \quad \text{Ecuación (XI.2)}$$

La distancia (pistas y cable) entre ambas placas durante las pruebas es menor o igual a la longitud de onda. Por la diferencia entre las longitudes de las pistas, no debería causar mayor diferencia pues la mayoría de sus tramos son directos, con diferencias entre ellos menores a 3 centímetros (excepto los diferenciales, en los cuales se procura mantener la misma longitud). Esto puede atacarse en proyectos que requieran de mayor frecuencia trazando pistas con ajustes en su longitud manteniéndolas a todas iguales.

Ninguna de las pistas tiene una longitud mayor que 30 centímetros, para ello deberían recorrer el perímetro de la PCB más de una vez.

Para evaluar el comportamiento del nuevo módulo, se fabrica una PCB que servirá de pruebas para distintas interfaces de tipo serial y comercialmente accesibles. Asegurando que cualquier perturbación sea ajena a la nueva PCB, seguimos las recomendaciones vistas a lo largo del programa del Máster y de diferentes fabricantes, como Texas Instruments.

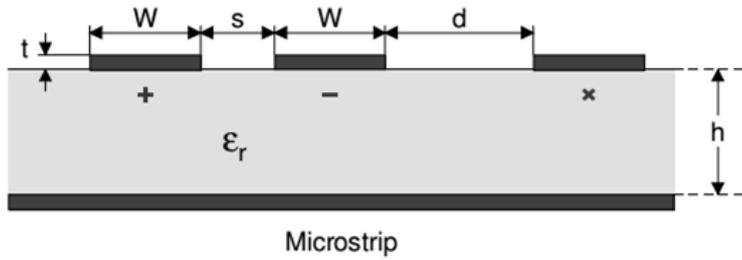


Fig. XI.1 - Geometría física de tramos diferenciales.

Las fórmulas que corresponden a la figura anterior son:

$$Z_0[\Omega] = \frac{88.75}{\sqrt{\epsilon_r + 1.47}} \cdot \ln\left(\frac{5.97 \cdot h}{0.8 \cdot W + t}\right) \quad \text{Ecuación (XI.3)}$$

$$Z_{DIFF}[\Omega] = 2 \cdot Z_0 \left(1 - 0.48 \cdot e^{-0.96 \frac{s}{h}}\right) \quad \text{Ecuación (XI.4)}$$

El fabricante de PCB predeterminado por el IFIC-Valencia ya utiliza un esquema de plantillas y reglas básicas para distintos diseños, por lo que se toma esta información y se realizan los cálculos para obtener una impedancia diferencial no menor a 100 Ω. Buscando entre el catálogo se encuentra la opción del “Artículo 186 ML4”.

SID Factory: Schopfheim	Article: <input type="text" value="186"/> <input type="text" value="ML4"/>	Provided: <input type="text" value="Wodke, Alexander"/>		
	Customer: <input type="text"/>	Date: <input type="text" value="17.04.2015"/>		
Processtechnology: B: undefiniert				
Material Text	Mat. Nr.	µm	Stackup	Process overview
A-RS Kupferfolie-018my 330x490mm	50200238	<input type="text" value="18"/>	VS	
A-RS-FR4-Prepreg-2116-TG135	50200534	<input type="text" value="288"/>		
A-RS-FR4-Prepreg-7628-TG135	50200465	<input type="text" value="0"/>		
A-RS-FR4-ML-0.71mm-035+035-TG135	50200375	<input type="text" value="35"/> <input type="text" value="710"/> <input type="text" value="35"/>	L2 L3	
A-RS-FR4-Prepreg-7628-TG135	50200465	<input type="text" value="288"/>		
A-RS-FR4-Prepreg-2116-TG135	50200534	<input type="text" value="0"/>		
A-RS Kupferfolie-018my 330x490mm	50200238	<input type="text" value="18"/>	RS	

Fig. XI.2 - Descripción del Artículo ML4 para la solicitud de PCB.

Con la ayuda de una hoja de cálculo de Ms-Excel y las ecuaciones (XI.3) y (XI.4) se obtienen los resultados:

	A	B	C	D	E	F	G
4							
5			mils	μm		Z_{DIFF}	114.7 Ω
6	s		3.9	100		Z_0	87.4 Ω
7	H		11.3	288		ϵ_r	4.5
8	W		5.9	150			
9	t		1.4	35			
10	Entre GND y GND		28.0	710			

Fig. XI.3 - Captura de pantalla: cálculo de impedancia diferencial.

Los valores están dentro de las recomendaciones y se respeta esta distancia entre tramos para señales diferenciales, al igual que el uso de antipads. La distancia mínima entre señales diferenciales adyacentes y blindaje en la placa de prueba es de 300 μm (entre LA21N y LA14P), es decir, $d \geq 3s$.

La tarjeta es pasiva, sin terminaciones ni fuentes adicionales, capas externas que manejan la señal y capas internas que serán planos con la referencia GND. Sólo servirá de adaptación para interconectar los FMC de cada placa electrónica. La mayor consideración en este aspecto es el GND, la cantidad de caminos de retorno de las señales diferenciales que estará limitado en parte para sacar el mayor provecho de los pares diferenciales.



Fig. XI.4 - Ejemplo de la guía de enrutados para vías según TI. A la izquierda, enrutado correcto usando pistas diferenciales. A la derecha, espaciado entre las vías y las pistas. Ante la ausencia de otros elementos en la placa, estas son las guías más aplicadas.

Se consideran los *anti-pads* y distancias entre vías que conducen señales diferenciales para que compartan el área de despeje [15].



Fig. XI.5 - Vista de los *anti-pads* para las vías de las señales diferenciales en el conector J6 (HDMI). La capa interna GND es de color amarillo en el diseño KiCad, vías de color gris y pistas de color rojo y verde.

Se usan planos de doble función (tierra-retorno de la señal).

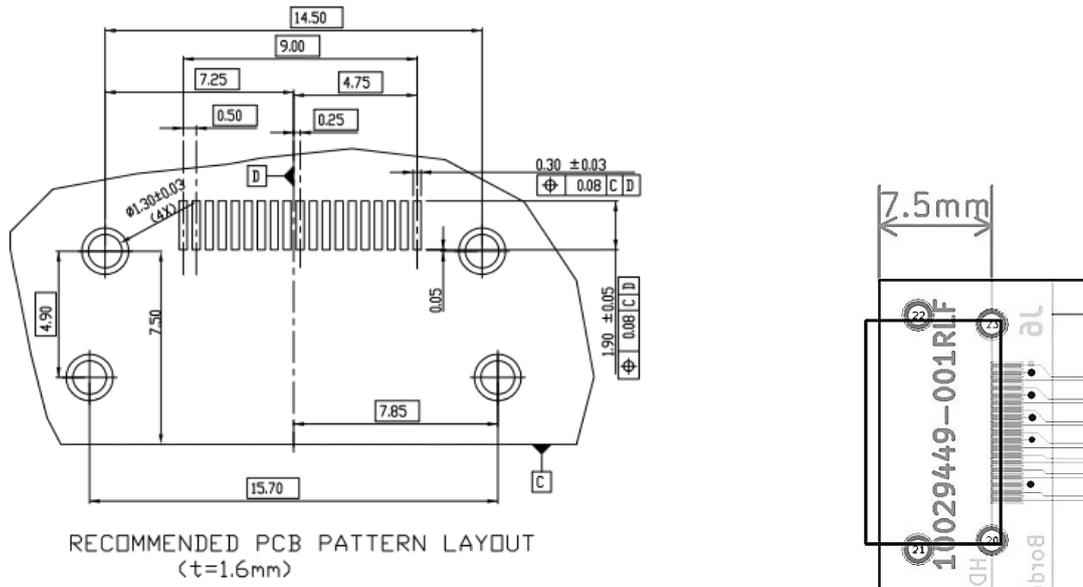


Fig. XI.8 - Dimensiones de la interfaz HDMI proporcionada por el distribuidor. A la izquierda, detalle de la hoja de especificaciones donde lo destacable es la distancia de 7,5 mm entre el centro del agujero de la patilla trasera hasta el borde de la PCB y el grosor recomendado de la PCB (1,6 mm). A la derecha, detalle del diseño en KiCad visto viendo a la cara inferior de la PCB.

Factor común es que el distribuidor de cada interfaz recomienda el grosor de la PCB de 1,55 -1,6 mm, condición que cumple el diseño.

XI. 3. Condiciones de diseño térmico

La placa electrónica no disipará mayor potencia, por lo que los pasantes incluidos en la PCB por recomendación de la ANSI/VITA 57.1-2008 son suficientes: dos sin platear, hechos con taladro de 2,5 mm que atraviesan todas las capas de cobre.

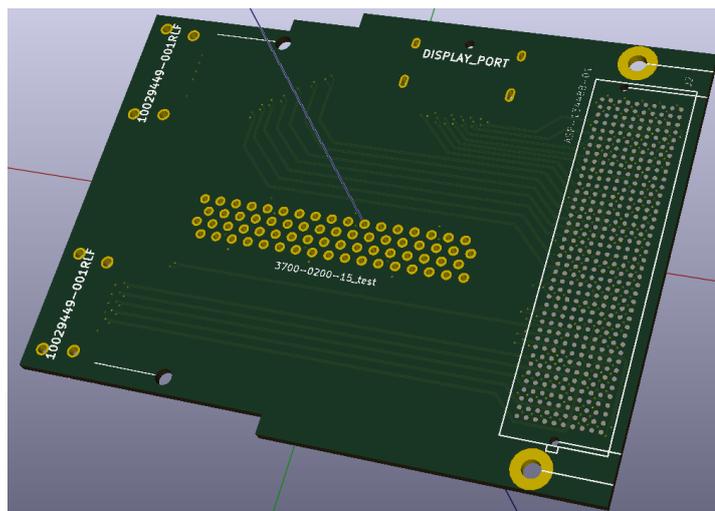


Fig. XI.9 - Diseño de la placa electrónica de pruebas: vista de la parte inferior del modelo en tres dimensiones de la placa.

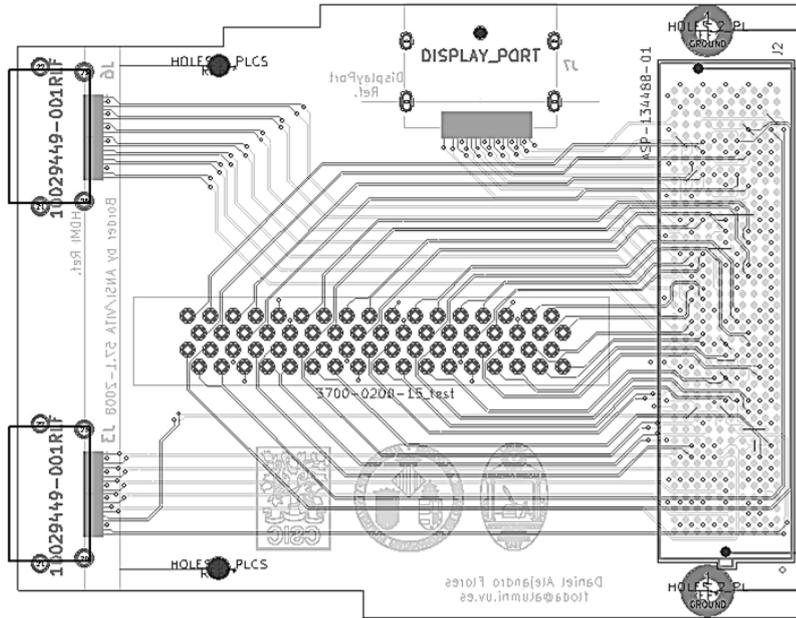


Fig. XI.10 - Diseño de la placa electrónica de pruebas: vista de las capas exteriores, vías, conectores y serigrafía.

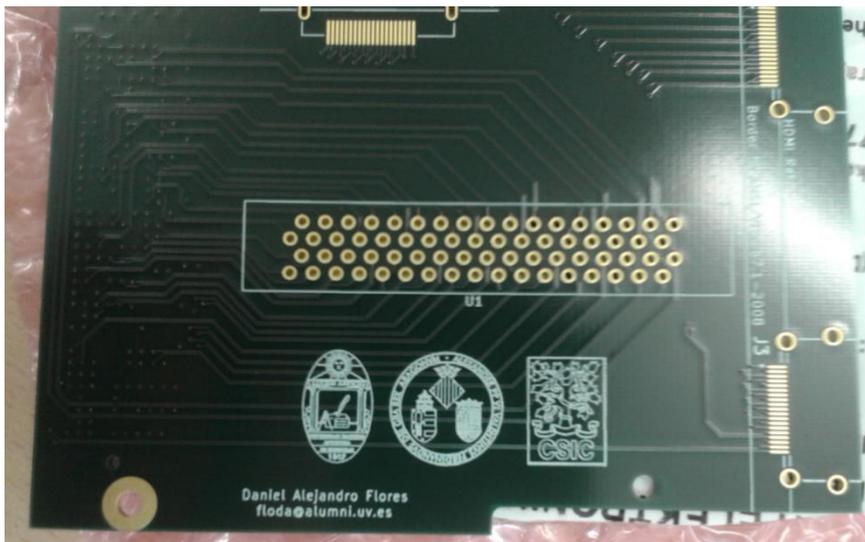


Fig. XI.11 - Placa electrónica para pruebas: vista de la cara superior.

XII. PRUEBAS Y RESULTADOS EXPERIMENTALES

XII. 1. Resultados antes de las modificaciones al felix_interface.vhd y elaboración de procedimiento

La configuración por defecto con la que inicia la aplicación elinkconfig no es compatible con la programación hardware de la XC7VX485T del TilePPr. Al inicio del trabajo los errores del FELIX se caracterizaban de la siguiente forma:

```
1 Pulse L1A signal (pw= 50 ns, per= 100 ns c/u)
tile_config_pkg.vhd
constant nSampleMaxFELIXReadout: positive:= 32;
-----
[tilecal@tical02 bin]$ fcheck test_02_trenpulsosfirmwarenuevo-170323-183201-1.dat
----> Blk 0: first 32 bits: abcd 1, last 16 bits: 23fa, eLinkNr 0x001, seqNumber 0
ERROR Blk 1: first 32 bits: abcd 801, last 16 bits: 0, eLinkNr 0x001, seqNumber 1
-----
Elink G-g-p Max L1Id Blocks Width Bytes null first last both middle timeout oob
nchunks minSize maxSize
0x001 = 0-0-1 0 2 0 0
Total number of Elinks 1, width 2:0, 4:0, 8:0, 16:0, unknown:1
```

Fig. XII.1 - Resultado experimental 01: análisis fcheck de la lectura de un pulso L1A. Nótese que el error ocurre por el trailer del segundo bloque generado por el TileDAQ.

```
8 Pulse L1A signal (pw= 50 ns, per= 100 ns c/u)
tile_config_pkg.vhd
constant nSampleMaxFELIXReadout: positive:= 8;
-----
[tilecal@tical02 bin]$ fcheck test_01_trenpulsosfirmwarenuevo-170323-133824-1.dat
----> Blk 0: first 32 bits: abcd 1, last 16 bits: 2046, eLinkNr 0x001, seqNumber 0
ERROR Blk 1: first 32 bits: abcd 801, last 16 bits: 0, eLinkNr 0x001, seqNumber 1
-----
Elink G-g-p Max L1Id Blocks Width Bytes null first last both middle timeout oob
nchunks minSize maxSize
0x001 = 0-0-1 0 2 0 0
Total number of Elinks 1, width 2:0, 4:0, 8:0, 16:0, unknown:1
```

Fig. XII.2 - Resultado experimental 02: análisis fcheck de la lectura de ocho pulsos L1A. Nótese que el error ocurre por el trailer del segundo bloque generado por el TileDAQ.

```

8 Pulse L1A signal (pw= 50 ns, per= 100 ns c/u)
tile_config_pkg.vhd
constant nSampleMaxFELIXReadout: positive:= 32;
-----

File test_01_trenpulsosfirmwarenuevo-170323-123917-1.dat contains 13312 bytes (13 K blocks +0)

Elink  G-g-p Max L1Id  Blocks  Width  Bytes  null first  last  both  middle timeout  oob
nchunks minSize maxSize
0x001 = 0-0-1      0      5      0      0
0x01D = 0-3-5      0      1      0      0
0x056 = 1-2-6      0      1      0      0
0x203 = 8-0-3      0      1      0      0
0x3CD = 15-1-5     0      4      0      0
0x51B = 20-3-3     0      1      0      0
Total number of Elinks 6, width 2:0, 4:0, 8:0, 16:0, unknown:6
[tilecal@tical02 bin]$ fcheck test_01_trenpulsosfirmwarenuevo-170323-123917-1.dat
----> Blk  0: first 32 bits: abcd 1, last 16 bits: 23fa, eLinkNr 0x001, seqNumber 0
----> Blk  1: first 32 bits: abcd 801, last 16 bits: 2178, eLinkNr 0x001, seqNumber 1
----> Blk  2: first 32 bits: abcd 1001, last 16 bits: 20f6, eLinkNr 0x001, seqNumber 2
----> Blk  3: first 32 bits: abcd 1801, last 16 bits: 21de, eLinkNr 0x001, seqNumber 3
----> Blk  4: first 32 bits: abcd 2001, last 16 bits: 2801, eLinkNr 0x001, seqNumber 4
ERROR Blk  5: first 32 bits: 7856 abcd, last 16 bits: 3001, eLinkNr 0x3CD, seqNumber 21
##### [BLCK] E=3CD Blk=6: seqnr 21, expected 22

ERROR Blk  6: first 32 bits: b02 abcd, last 16 bits: 3801, eLinkNr 0x3CD, seqNumber 21
##### [BLCK] E=3CD Blk=7: seqnr 21, expected 22

ERROR Blk  7: first 32 bits: fa79 abcd, last 16 bits: 4001, eLinkNr 0x3CD, seqNumber 21
##### [BLCK] E=3CD Blk=8: seqnr 21, expected 22

ERROR Blk  8: first 32 bits: 8482 abcd, last 16 bits: abcd, eLinkNr 0x3CD, seqNumber 21
##### [BLCK] E=3CD Blk=8: seqnr 21, expected 22

ERROR Blk  9: first 32 bits: 3412 7856, last 16 bits: abcd, eLinkNr 0x056, seqNumber 15
ERROR Blk 10: first 32 bits: 1303 1203, last 16 bits: abcd, eLinkNr 0x203, seqNumber 2
ERROR Blk 11: first 32 bits: 51c 51b, last 16 bits: abcd, eLinkNr 0x51B, seqNumber 0
ERROR Blk 12: first 32 bits: 1e 1d, last 16 bits: abcd, eLinkNr 0x01D, seqNumber 0
-----

Elink  G-g-p Max L1Id  Blocks  Width  Bytes  null first  last  both  middle timeout  oob
nchunks minSize maxSize
0x001 = 0-0-1      0      5      0      0
0x01D = 0-3-5      0      1      0      0
0x056 = 1-2-6      0      1      0      0
0x203 = 8-0-3      0      1      0      0
0x3CD = 15-1-5     0      4      0      0
0x51B = 20-3-3     0      1      0      0
Total number of Elinks 6, width 2:0, 4:0, 8:0, 16:0, unknown:6

```

Fig. XII.3 - Resultado experimental 03: análisis fcheck de la segunda lectura de ocho pulsos L1A. Nótese que el error ocurre por el trailer del penúltimo bloque generado por el TileDAQ.

Este error genera una corrupción de datos en todos los siguientes bloques y fragmentos. Ajustando las señales de reloj del felix_interface.vhd y con la configuración del TileDAQ esto se subsana.

Para mayores tiempos de muestreo, como 20 minutos para 1000 L1A por segundo, con el programa unboxing_v1_1.py se obtiene una mayor cantidad de fragmentos de la que se podría esperar cuando el L1A era generado por el TTCex que cuando se generaban por simulación. Ante esta situación se hicieron varias pruebas y se analizaron varias señales por ChipScope, encontrando aleatoriamente la presencia de un L1A fuera de frecuencia, probablemente por un bug en el generador de pulsos ajeno a la VC707.

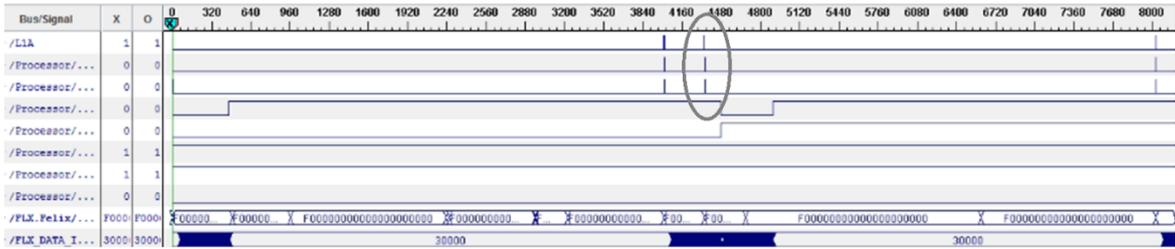


Fig. XII.4 - Resultado experimental: presencia de pulso L1A fuera de la frecuencia asignada (resaltado por óvalo).

Ciertos errores no se detectan por el fcheck, eran aquellos que provenían del TilePPr. Si el bloque que es enviado por el TileDAQ al servidor está completo, mantiene la secuencia correcta y el conteo de fragmentos es correcto, difícilmente la función fcheck indicará la presencia de error. Para ello se utiliza el software unboxing_v1_1.py desarrollado en este trabajo.

```
[tilecal@tical02 bin]$ fcheck test_PPR_08_1Hz_adq600s-170718-143611-1.dat
File test_PPR_08_1Hz_adq600s-170718-143611-1.dat contains 102400 bytes (100 1K blocks +0)

Elink  G-g-p  Max L1Id  Blocks  Width  Bytes  null  first  last  both  middle  timeout  oob  nchunks  minSize  maxSize
0x001 = 0-0-1  0      100    0      100732  0     99    98    377   0       0       0     475    211    212
Total number of Elinks 1, width 2:0, 4:0, 8:0, 16:0, unknown:1
[tilecal@tical02 bin]$
```

Fig. XII.5 - Resultado experimental 04: análisis fcheck de pulsos L1A a 60 Hz. Ningún error a nivel de bloque es detectado por fcheck.

The screenshot shows the unboxing_v1_1.py application window. It is divided into several sections:

- GENERAL ANALYSIS:** Shows file path, file name (PPR_08_1Hz_adq600s-170718-143611-1), file size (102400 bytes), and number of 16-byte chains (6400).
- BLOCK ANALYSIS:** Shows the number of blocks (100) and provides options to create or open block-formatted files.
- Fragment Analysis:** Shows the number of fragments (476.8), headers found (356), trailers found (355), wrong lectures of headers (12839), trailers with errors (1), and error percentage (25.3356%).
- Channel Analysis:** A button to enable the channel analysis menu.
- Status:** Shows "Created .txt file fragment formatted."

Fig. XII.6 - Resultado experimental 05: análisis de unboxing_v1_1.py de pulsos L1A a 60 Hz. Se confirma el dimensionamiento de bloques y fragmentos a nivel de bloque con la función fcheck, pero el software creado en este trabajo detecta algunos errores a nivel de fragmento.

```

00e000f0000010101020103010401050106010701080109010a010b010c01
20402050206020702080209020a020b020c020d020e020f02000001000200
00a000b000c000d000e000f00000101010201030104010501060107010801
20b020c020d020e020f0243218765424056123400000000000004d306ee00
101020103010401050106010701080109010a010b010c010d010e010f0100
702080209020a020b020c020d020e020f0200000100020003000400050006
00d000e000f0000010101020103010401050106010701080109010a010b01
20e020f02432187653c40561234000000000000059a070900000000010002
10401050106010701080109010a010b010c010d010e010f01000201020202
20a020b020c020d020e020f02000001000200030004000500060007000800
000010101020103010401050106010701080109010a010b010c010d010e01
76500354056781234000000000000048d071e000000000100020003000400
701080109010a010b010c010d010e010f0100020102020203020402050206
20d020e020f0200000100020003000400050006000700080009000a000b00
103010401050106010701080109010a010b010c010d010e010f0100020102
812340000000000000420073500000000100020003000400050006000700
a010b010c010d010e010f0100020102020203020402050206020702080209
    
```

Fig. XII.7 - Resultado experimental 06: archivo .txt en formato de bloques generado por unboxing_v1_1.py. Cada línea de texto corresponde a un bloque de datos y resaltado en color amarillo está el error detectado en los fragmentos: la secuencia de cabecera x"5678" x"1234" carece del byte x"78".

Se descarta que el origen del problema sea el TileDAQ y con el ChipScope no se encuentra este fallo en la señal FLX_DATA_IN, por lo que se aísla la fuente del problema entre la entrada al módulo FLX.Felix del TilePPr y la entrada al TileDAQ. Haciendo debugging se descubre que el bug está en la lógica que controla a la FIFO del archivo FIFO2Elink.vhd.

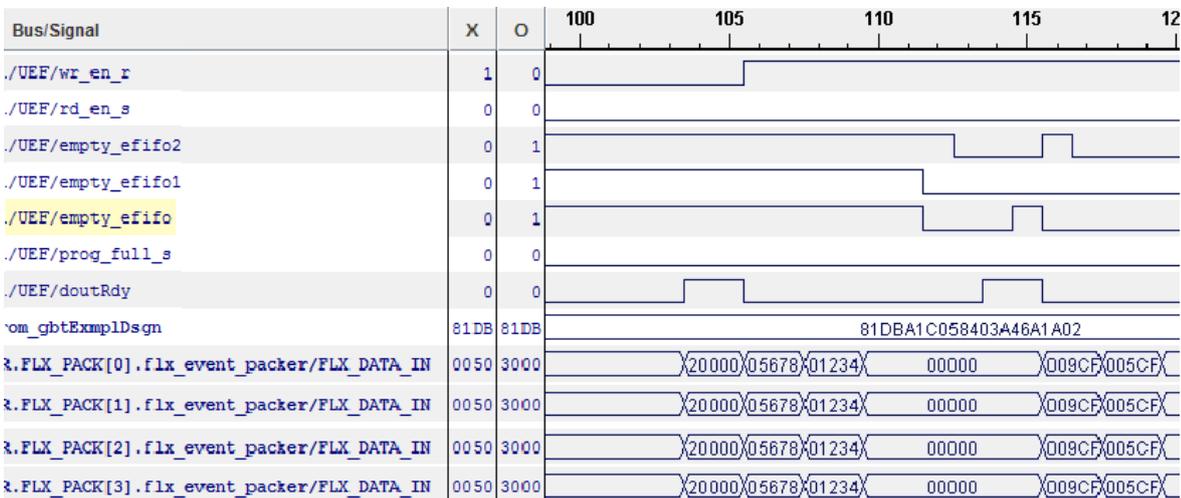


Fig. XII.8 - Resultado experimental 07 a: señales de las FIFO capturadas por ChipScope con muestras a 80 MHz. Señal de que la FIFO se vacía durante el paso de la cabecera de la trama.

Como la frecuencia de lectura es en ocasiones ligeramente superior a la de escritura y la señal read enable depende de si la FIFO está llena o vacía, simplemente se agrega un case para indicar que está vacía aun cuando tenga al menos una palabra todavía. Es importante saber que este código solamente

controla la señal *empty* enviada cuando la FIFO se está llenando, al vaciarse no tiene efecto.

```

-----
-- empty_efifo signal controlled by RD_DATA_COUNT
-----

process (state,empty_efifo_signal,rd_data_count,rd_clk)
begin
  if rising_edge (rd_clk) then
    case (state) is
      when idle =>
        empty_efifo <= '1';
        if empty_efifo_signal='0' and
rd_data_count > x"0" then
          state <= case1;
        else
          state <= idle;
        end if;
      when case1 =>
        empty_efifo <= '0';
        if empty_efifo_signal='0' then
          state <= case1;
        else
          empty_efifo <= '1';
          state <= idle;
        end if;
      when others =>
        empty_efifo <= '1';
        state <= idle;
    end case;
  end if;
end process;

```

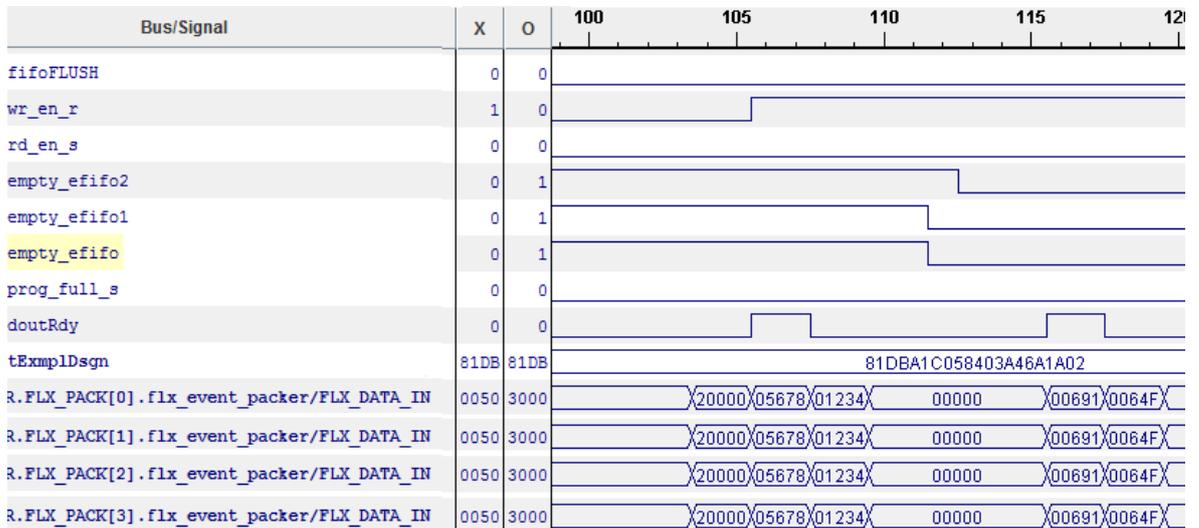


Fig. XII.9 - Resultado experimental 07 b: señales de las FIFO capturadas por ChipScope con muestras a 80 MHz. Señal normal de la FIFO.

Otros bugs y modificaciones se hacen sobre el código del FELIX que se programa en la XC7VX485T del TilePPr, siendo los mencionados los más relevantes y que su corrección ha llevado al correcto funcionamiento del sistema de comunicación para agregar los nuevos módulos del integrador y del FE-I4. Para una única señal de trigger L1A, los datos adquiridos y guardados en el archivo con nombre TFM_02_un_solo_pulso-170608-192047-1.dat:

```

==> BLOCK 0 (E=001=0-0-1 seq=0):
  0:  1  0 cd ab 56 78 12 34  0  0  0  0  0  3  0 76
 16:  0  1  1  0  0  0  1  0  2  0  3  0  4  0  5  0
 32:  6  0  7  0  0  1  1  1  2  1  3  1  4  1  5  1
 48:  6  1  7  1  0  2  1  2  2  2  3  2  4  2  5  2
 64:  6  2  7  2  0  0  1  0  2  0  3  0  4  0  5  0
 80:  6  0  7  0  0  1  1  1  2  1  3  1  4  1  5  1
 96:  6  1  7  1  0  2  1  2  2  2  3  2  4  2  5  2
112:  6  2  7  2 43 21 87 65 74 60  a0  0  0  0  0
128:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
144:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
160:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
176:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
192:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
208:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
224:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
240:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
256:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
272:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
288:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
304:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
320:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
336:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
352:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
368:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
384:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
400:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
416:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
432:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
448:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
464:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
480:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
496:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

      .
      .
      .

 976:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 992:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
1008:  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

```

Fig. XII.10 - Trama de datos número 1. Imagen obtenida del terminal cuando se ejecuta la función “fcheck -F 1” mientras está en ejecución el programa setup.sh.

Para tres señales de trigger consecutivas de L1A, los datos adquiridos y guardados en el archivo con nombre TFM_01_tren_tres_pulsos_switch-170608-173553-1.dat.

Sistema de transferencia de datos en el instrumento TilePPr del proyecto TileCal

```

==> BLOCK 0 (E=001=0-0-1 seq=0):
  0:  1 0 cd ab 56 78 12 34 0 0 0 0 0 3 a e5
 16: 42 8f 1b 0 0 0 1 0 2 0 3 0 4 0 5 0
 32:  6 0 7 0 8 0 9 0 a 0 b 0 c 0 d 0
 48:  e 0 f 0 0 1 1 1 2 1 3 1 4 1 5 1
 64:  6 1 7 1 8 1 9 1 a 1 b 1 c 1 d 1
 80:  e 1 f 1 0 2 1 2 2 2 3 2 4 2 5 2
 96:  6 2 7 2 8 2 9 2 a 2 b 2 c 2 d 2
112:  e 2 f 2 0 3 1 3 2 3 3 3 4 3 5 3
128:  6 3 7 3 8 3 9 3 a 3 b 3 c 3 d 3
144:  e 3 f 3 0 4 1 4 2 4 3 4 4 4 5 4
160:  6 4 7 4 8 4 9 4 a 4 b 4 c 4 d 4
176:  e 4 f 4 0 5 1 5 2 5 3 5 4 5 5 5
192:  6 5 7 5 8 5 9 5 a 5 b 5 c 5 d 5
208:  e 5 f 5 0 6 1 6 2 6 3 6 4 6 5 6
224:  6 6 7 6 8 6 9 6 a 6 b 6 c 6 d 6
240:  e 6 f 6 0 7 1 7 2 7 3 7 4 7 5 7
256:  6 7 7 7 8 7 9 7 a 7 b 7 c 7 d 7
272:  e 7 f 7 0 8 1 8 2 8 3 8 4 8 5 8
288:  6 8 7 8 8 8 9 8 a 8 b 8 c 8 d 8
304:  e 8 f 8 0 9 1 9 2 9 3 9 4 9 5 9
320:  6 9 7 9 8 9 9 9 a 9 b 9 c 9 d 9
336:  e 9 f 9 0 a 1 a 2 a 3 a 4 a 5 a
352:  6 a 7 a 8 a 9 a a a b a c a d a
368:  e a f a 0 b 1 b 2 b 3 b 4 b 5 b
384:  6 b 7 b 8 b 9 b a b b b c b d b
400:  e b f b 0 0 1 0 2 0 3 0 4 0 5 0
416:  6 0 7 0 8 0 9 0 a 0 b 0 c 0 d 0
432:  e 0 f 0 0 1 1 1 2 1 3 1 4 1 5 1
448:  6 1 7 1 8 1 9 1 a 1 b 1 c 1 d 1
464:  e 1 f 1 0 2 1 2 2 2 3 2 4 2 5 2
480:  6 2 7 2 8 2 9 2 a 2 b 2 c 2 d 2
496:  e 2 f 2 0 3 1 3 2 3 3 3 4 3 5 3
512:  6 3 7 3 8 3 9 3 a 3 b 3 c 3 d 3
528:  e 3 f 3 0 4 1 4 2 4 3 4 4 4 5 4
544:  6 4 7 4 8 4 9 4 a 4 b 4 c 4 d 4
560:  e 4 f 4 0 5 1 5 2 5 3 5 4 5 5 5
576:  6 5 7 5 8 5 9 5 a 5 b 5 c 5 d 5
592:  e 5 f 5 0 6 1 6 2 6 3 6 4 6 5 6
608:  6 6 7 6 8 6 9 6 a 6 b 6 c 6 d 6
624:  e 6 f 6 0 7 1 7 2 7 3 7 4 7 5 7
640:  6 7 7 7 8 7 9 7 a 7 b 7 c 7 d 7
656:  e 7 f 7 0 8 1 8 2 8 3 8 4 8 5 8
672:  6 8 7 8 8 8 9 8 a 8 b 8 c 8 d 8
688:  e 8 f 8 0 9 1 9 2 9 3 9 4 9 5 9
704:  6 9 7 9 8 9 9 9 a 9 b 9 c 9 d 9
720:  e 9 f 9 0 a 1 a 2 a 3 a 4 a 5 a
736:  6 a 7 a 8 a 9 a a a b a c a d a
752:  e a f a 0 b 1 b 2 b 3 b 4 b 5 b
768:  6 b 7 b 8 b 9 b a b b b c b d b
784:  e b f b 43 21 87 65 14 63 56 78 12 34 0 0
800:  0 0 0 3 a eb 42 90 1b 0 0 0 1 0 2 0
816:  3 0 4 0 5 0 6 0 7 0 8 0 9 0 a 0
832:  b 0 c 0 d 0 e 0 f 0 0 1 1 1 2 1
848:  3 1 4 1 5 1 6 1 7 1 8 1 9 1 a 1
864:  b 1 c 1 d 1 e 1 f 1 0 2 1 2 2 2
880:  3 2 4 2 5 2 6 2 7 2 8 2 9 2 a 2
896:  b 2 c 2 d 2 e 2 f 2 0 3 1 3 2 3
912:  3 3 4 3 5 3 6 3 7 3 8 3 9 3 a 3
928:  b 3 c 3 d 3 e 3 f 3 0 4 1 4 2 4
944:  3 4 4 4 5 4 6 4 7 4 8 4 9 4 a 4
960:  b 4 c 4 d 4 e 4 f 4 0 5 1 5 2 5
976:  3 5 4 5 5 5 6 5 7 5 8 5 9 5 a 5
992:  b 5 c 5 d 5 e 5 f 5 0 6 1 6 2 6
1008: 3 6 4 6 5 6 6 6 7 6 8 6 9 6 e4 20
==> BLOCK 1 (E=001=0-0-1 seq=1):

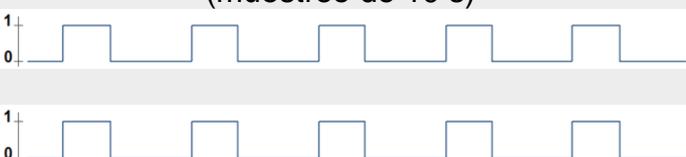
```

Fig. XII.11 - Trama de datos número 2. Imagen obtenida del terminal cuando se ejecuta la función "fcheck -F 3" mientras está en ejecución el programa setup.sh.

La información presentada en los bloques es congruente con lo programado en la FPGA del TilePPr.

XII. 2. Independencia del INT2HOST en el TilePPr: configuración inicial

TABLA XII.1 Condiciones de prueba iniciales para el L1A y el INT2HOST

Código de la prueba	Condiciones de prueba	Rutas a verificar
T_01_01	<p>Característica de la señal L1A: Pulso único (muestreo de 5 s)</p>  <p>Fuente de pulsos L1A: PPr RunParams(0) (15 downto 8) = 03 Carpeta del proyecto: FELIX_un_solo_pulso_dataclk_40</p>	<p>GBT Link 0 Egroup 0 epath1: 8 (0) eproc8bits</p>
T_01_02	<p>Característica de la señal L1A: Tren de tres pulsos (muestreo de 5 s)</p>  <p>Fuente de pulsos L1A: PPr</p> <p>Carpeta del proyecto: FELIX_tren_de_pulsos_o_canal_A</p>	<p>GBT Link 0 Egroup 0 epath1: 8 (0) eproc8bits</p>
T_01_03	<p>Característica de la señal L1A: Dos trenes no consecutivos de cinco pulsos (muestreo de 10 s)</p>  <p>Característica de la señal INT2HOST: Trama de datos de un solo eproc de 2 bits</p> <p>Fuente de pulsos L1A: PPr Carpeta del proyecto: proyecto_integrador_00</p>	<p>GBT Link 0 Egroup 0 epath1: 8 (0) eproc8bits</p> <p>GBT Link 0 Egroup 2 epath1: 2 (32) eproc2bits</p>

Tres pruebas de fácil examinación permiten ver de qué manera se comporta la programación hardware en la FPGA de la VC707. La primera prueba solo evalúa el comportamiento de un pulso único para una trama de datos de 116 bytes,

configurando el elink a un tamaño finito de datos por bloque y de tiempo de espera. La segunda prueba evalúa tres pulsos consecutivos de L1A pero con tramas de datos de 788 bytes, el tamaño aproximado que se espera para las pruebas de certificación y la práctica. En el caso de la tercera prueba, son enviados 10 fragmentos de 116 bytes mientras son enviados todas las tramas de datos del integrador: ninguno de los canales debe interferirse por la presencia del otro.

TABLA XII.2 Resultados de prueba iniciales para el L1A y el INT2HOST

Código de la prueba	Ruta	Requisitos	Cumplimiento
T_01_01	eproc8	Cantidad de fragmentos = 1	Si
T_01_02	eproc8	Cantidad de fragmentos = 3	Si
T_01_03	eproc8	Cantidad de fragmentos ¹⁷ = 8	Si
	eproc2	Independencia respecto a otro eproc	Si

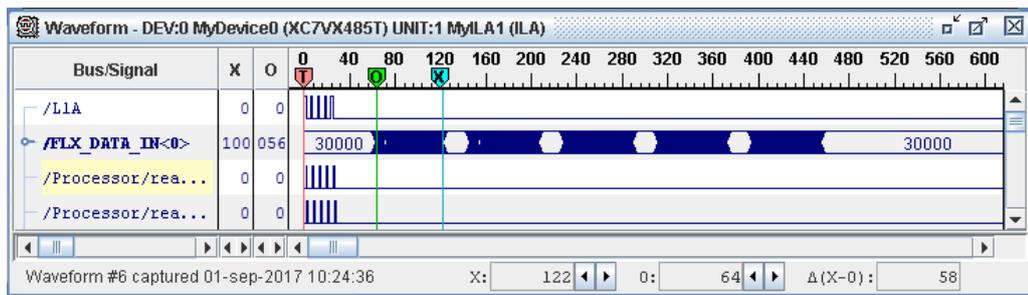


Fig. XII.12 - Captura del eproc8 de la prueba T_01_03 con muestras de 25 ns. El fragmento tiene una cantidad de 58 muestras (1450 ns), medido desde el inicio de la cabecera hasta terminar el trailer. La distancia entre el comienzo del EOP hasta el final del siguiente SOP es de 25 muestras (625 ns).

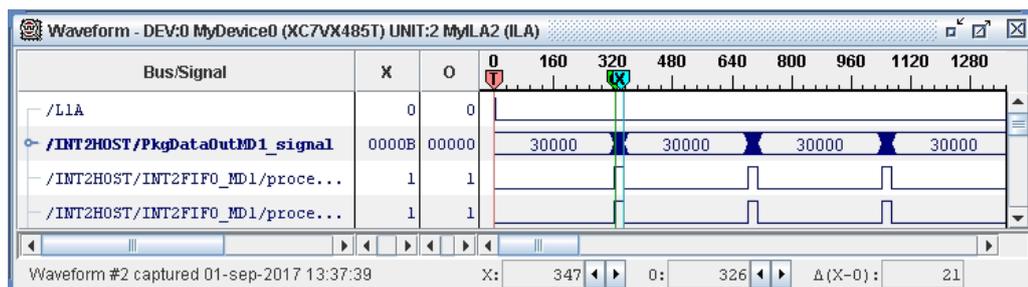


Fig. XII.13 - Captura del eproc2 de la prueba T_01_03 con muestras de 12,5 ns. El fragmento tiene una cantidad de 22 muestras (275 ns), medido desde el inicio de la cabecera hasta terminar el trailer. La distancia entre el comienzo del EOP hasta el final del siguiente SOP es de 338 muestras (4225 ns).

¹⁷ Recordad que la configuración de elink solamente permite los bloques enteros para ser enviados al servidor local, por lo que se deben enviar al menos nueve fragmentos de 232 hexadecimales, 116 bytes o 928 bits.

TABLA XII.3 Resultados comprobados con el fcheck (servidor local):

Código de la prueba	Ruta	Requisito	Resultado		
			ChipScope	Calculado	Medido
T_01_01	eproc8	Bloques con errores	-	0	0
		Bloques sin errores	-	1	1
		Tamaño total en bytes	116	116	116
		Cantidad de nChunks completos	1	1	1
T_01_02	eproc8	Bloques con errores	-	0	0
		Bloques sin errores	-	2	2
		Tamaño total en bytes	2364	2048	2048
		Cantidad de nChunks completos	3	2	2
T_01_03	eproc8	Bloques con errores	-	0	0
		Bloques sin errores	-	1	1
		Tamaño total en bytes	1160	1024	1002
		Cantidad de nChunks completos	10	8	8
	eproc2	Bloques con errores	-	0	0
		Bloques sin errores	-	50850.6	52418
		Tamaño total en bytes	-	52071008	48914732
		Cantidad de nChunks completos	-	2366864	2223396

TABLA XII.4 Condiciones para la variación de la frecuencia de L1A

Código	Frecuencia de L1A	Rutas a verificar	Duración de la muestra
T_02_01	1 Hz	GBT Link 0 Egroup 0 epath1: 8 (0) eproc8bits	100 s
T_02_02	10 Hz		100 s
T_02_03	100 Hz		100 s
T_02_04	1 kHz		100 s
T_02_05	10 kHz		100 s
T_02_06	100 kHz		100 s
T_02_07	200 kHz		10 s
Carpeta del proyecto:		proyecto_integrador_00	

TABLA XII.5 Resultados de la variación de la frecuencia de L1A

Código de la prueba	Ruta	Requisito	Resultado	
			Esperado	Medido
T_02_01	eproc8	Bloques con errores	0	0
		Bloques sin errores	-	11
		Tamaño total en bytes	11600	11010
		Cantidad de nChunks completos	100	94
		Velocidad	-	0.0 MB/s
T_02_02	eproc8	Bloques con errores	0	0
		Bloques sin errores	-	115
		Tamaño total en bytes	116000	115090
		Cantidad de nChunks completos	1000	992
		Velocidad	-	0.0 MB/s
T_02_03	eproc8	Bloques con errores	0	0
		Bloques sin errores	-	1159
		Tamaño total en bytes	1160000	1159908
		Cantidad de nChunks completos	10000	9999
T_02_04	eproc8	Bloques con errores	0	0
		Bloques sin errores	-	11593
		Tamaño total en bytes	11600000	11602092
		Cantidad de nChunks completos	100000	100018
		Velocidad	-	0.1 MB/s
T_02_05	eproc8	Bloques con errores	0	0
		Bloques sin errores	-	115832
		Tamaño total en bytes	116000000	115922850
		Cantidad de nChunks completos	1000000	999334
		Velocidad	-	1.2 MB/s
T_02_06	eproc8	Bloques con errores	0	0
		Bloques sin errores	-	1048747
		Tamaño total en bytes ¹⁸	1160000000	1153787146
		Cantidad de nChunks completos ¹⁹	10000000	9926071
		Velocidad	-	11.8 – 12.0 MB/s
T_02_07	eproc8	Bloques con errores	0	0
		Bloques sin errores	-	228191
		Tamaño total en bytes	232000000	228369974
		Cantidad de nChunks completos	2000000	1968706
		Velocidad	-	22.9 – 23.7 MB/s

¹⁸ Divididos en dos archivos por el servidor local: primer archivo 1049569546, segundo archivo 104217600.

¹⁹ Divididos en dos archivos por el servidor local: primer archivo 9048013 bloques, segundo archivo 878058 bloques.

XII. 3. BER: Tasa de error binario

Los resultados experimentales no han arrojado ningún error que fuera detectado por las herramientas software de las que se dispone. Sin embargo, el número de muestras es (obviamente) finito, por lo que existen métodos estadísticos que permiten estimar, con un determinado nivel de confianza, la cantidad de bits de error entre la cantidad de bits totales, esta relación es la BER. Como en este caso ninguno de los bits posee errores y para atribuirle un valor razonable a la BER utilizamos las ecuaciones [16]:

$$BER = \frac{N_{Err}}{N_{bits}} \quad \text{Ecuación (XII.1)}$$

$$CL = 1 - e^{-N_{bits} \cdot BER} \quad \text{Ecuación (XII.2)}$$

Un nivel de confianza CL de 95.45 % es aceptable en la mayoría de ensayos y mediciones [17] por lo que se buscará mantener este valor. Con estas condiciones y con los resultados registrados, se estima la BER.

TABLA XII.6 Tasa de error binario según las pruebas experimentales

Código de la prueba	bytes	N_{bits}	BER	CL
T_01_01	116	928	3.50×10^{-03}	0.96
T_01_02	2048	16384	2.00×10^{-04}	0.96
T_01_03 (eproc 8 bits)	1002	8016	3.90×10^{-04}	0.96
T_01_03 (eproc 2 bits)	52418	419344	7.50×10^{-06}	0.96
T_02_01	11010	88080	3.70×10^{-05}	0.96
T_02_02	115090	920720	3.50×10^{-06}	0.96
T_02_03	1159908	9279264	3.50×10^{-07}	0.96
T_02_04	11602092	92816736	3.50×10^{-08}	0.96
T_02_05	115922850	927382800	3.50×10^{-09}	0.96
T_02_06	1153787146	9230297168	3.50×10^{-10}	0.96
T_02_07	228369974	1826959792	1.70×10^{-09}	0.96

La prueba de mayor interés es la de 100 kHz T_02_06, porque los datos de los eventos seleccionados por el sistema *trigger* de nivel 1 son transmitidos a los ROD ubicados en el terminal trasero a una tasa máxima mantenida de 100 kHz [18].

XII. 4. Resultados experimentales del FE-14

Los resultados experimentales del FE-14 entre la placa electrónica VC707 y la KC705 se resumen en pruebas de dos tipos: pruebas de lazo cerrado y pruebas de conteo de cabeceras y colas. Las formas de onda se obtienen del bloque IP ILA incorporado en la librería VIVADO y la mayoría se hicieron con muestreos a 40 MHz.

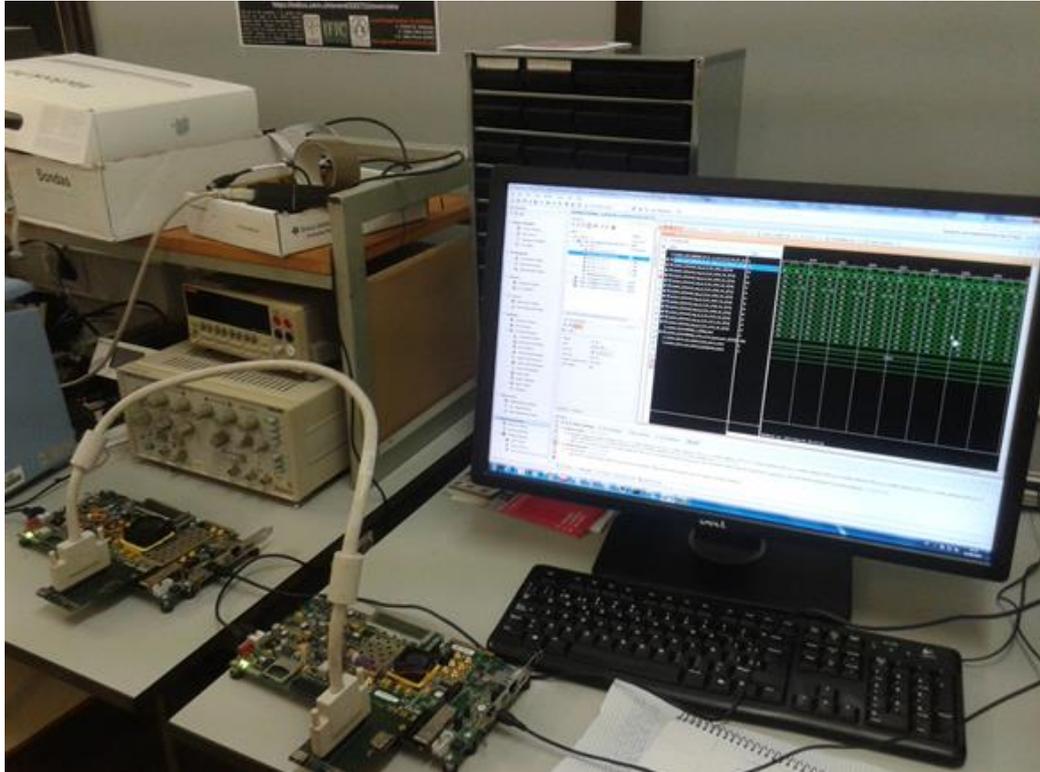


Fig. XII.14 - Monitoreo por ILA de las tramas de datos transferidas para su verificación. El cable SCSI que se muestra en la figura es el recomendado para esta aplicación.

XII. 4. 1 Configuración de FIFO

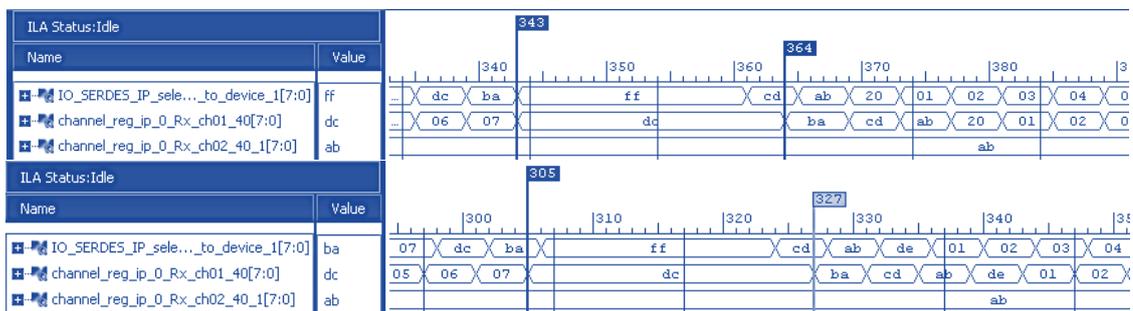


Fig. XII.15 - Formas de onda por ILA capturadas: comparación para demostrar los desfases que ocurren en el envío de la trama de datos de la XC7K325T a la XC7VX485T. Las muestras corresponden a un reloj de 40 MHz. El control de la FIFO no está configurado, la lectura y escritura son continuas.

La cantidad de palabras con las que se pretende llenar la memoria de FIFO antes de comiencen a leerse debe proveer de alguna latencia para que el lector

final pueda leer de forma continua sin que se vacíe o se llene la memoria de FIFO. Los datos de anchura 8 bits y generados con un reloj a 20 MHz pasarán a ser datos de anchura 8 bits con un reloj a 40 MHz (eproc), esto implica que ineludiblemente se darán periodos de latencia así que lo mejor es administrarlos directamente.

Para este trabajo se deja que la memoria de la FIFO almacene dos fragmentos de 14 bytes, correspondiente a la trama de datos propuesta en las especificaciones finales.

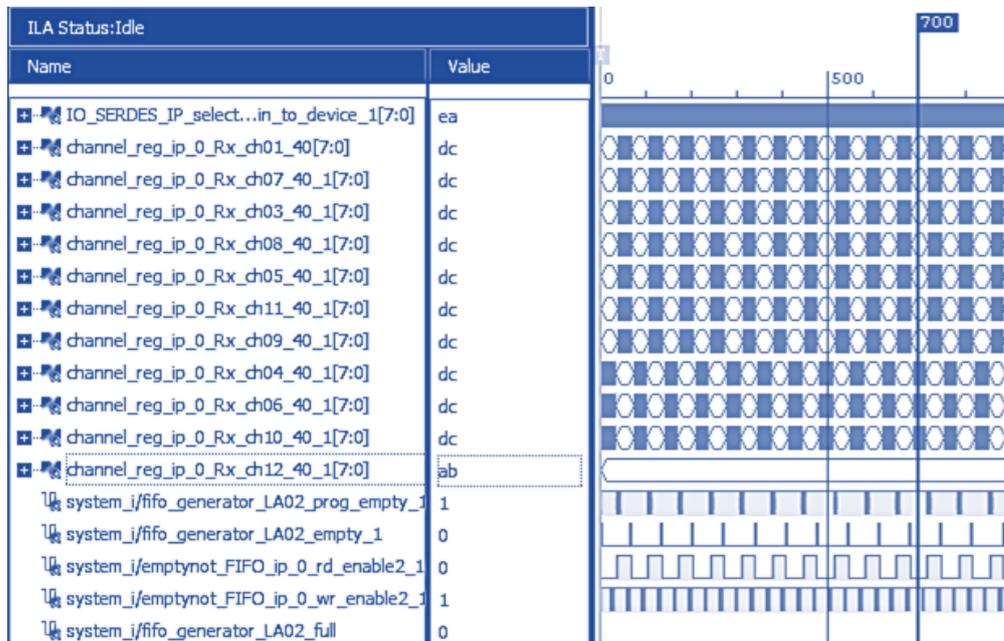


Fig. XII.16 - Formas de onda por ILA capturadas: conjunto de los canales con el correcto funcionamiento del control de la FIFO. Las muestras corresponden a un reloj de 40 MHz.

XII. 4. 2 Lazo cerrado

Esta prueba consiste en reenviar directamente a la KC705 lo que la VC707 recibió previamente. En este modo, la lectura y escritura de FIFO es continua y se aprovecha la segunda palabra clave x"CD" dentro de la trama propuesta.

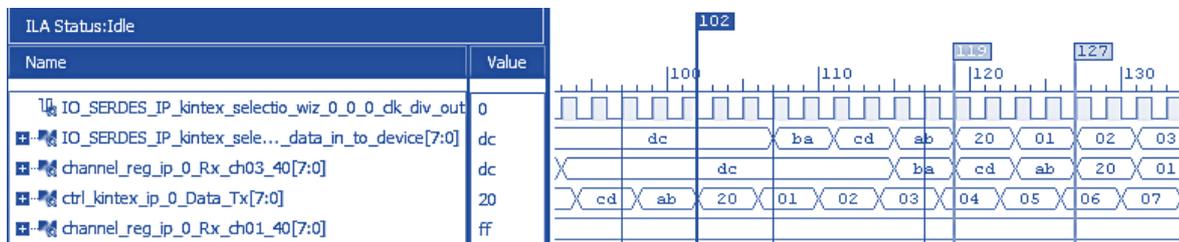


Fig. XII.17 - Formas de onda por ILA capturadas: trama de datos enviada por el canal 1 de la XC7K325T, procesada por la XC7VX485T y reenviada a la XC7K325T. Las muestras corresponden a un reloj de 40 MHz, por lo que una trama enviada desde el generador de datos en la KC705, procesada por la VC707 y recibida por el ILA a la salida de una FIFO en la KC705 tarda 25 muestras (625 ns), la trama tiene una duración de 48 muestras (1200 ns).

XII. 4. 3 Conteo de cabeceras y colas: EMC en el cable SCSI

Cuando se utiliza un cable SCSI que no posee ninguna protección frente a la contaminación electromagnética, aparentemente la trama de datos permanece intacta, pero al realizar el conteo de las tramas transferidas se nota como algunos canales sufren las consecuencias.

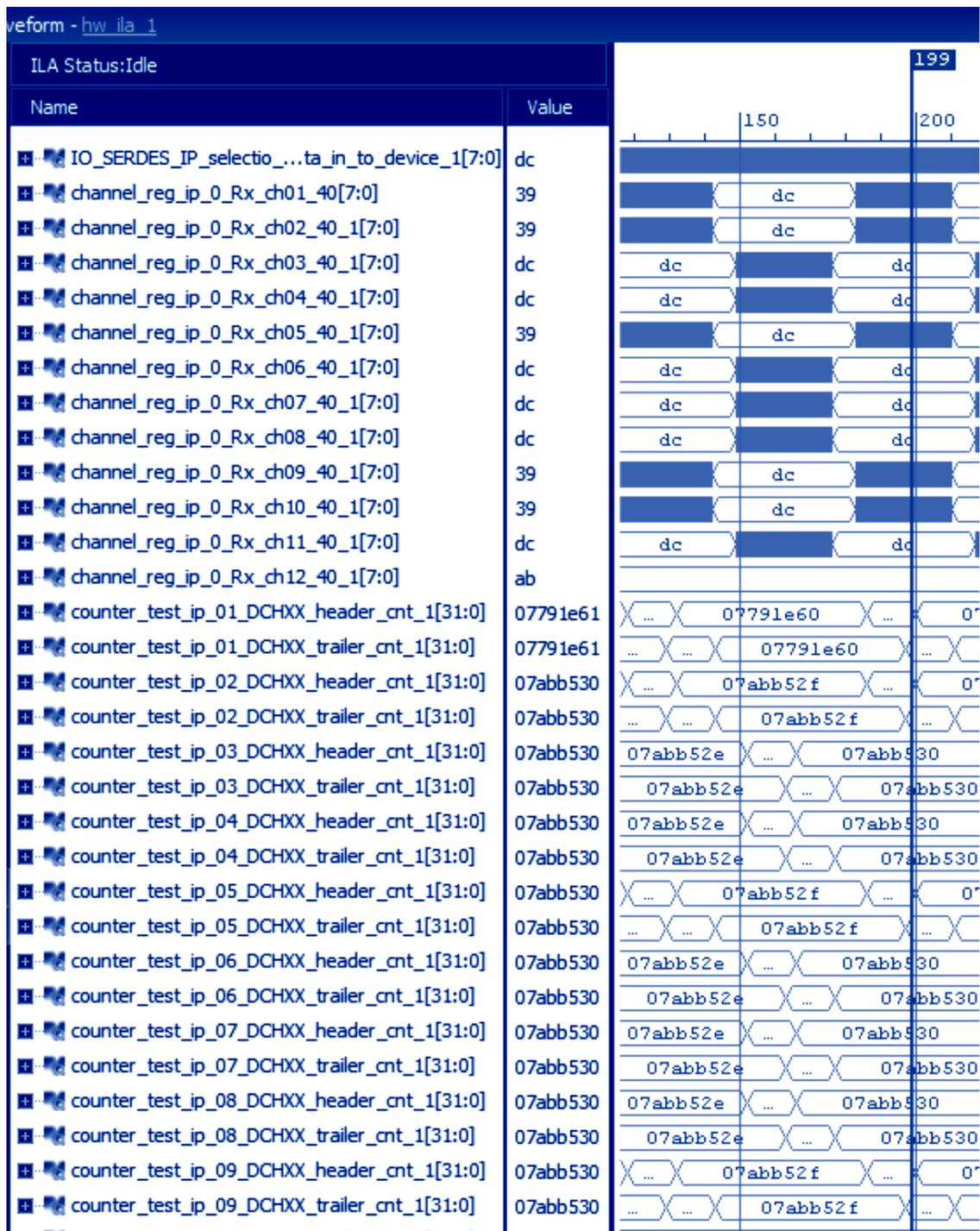


Fig. XII.18 - Formas de onda por ILA capturadas: uso de cable SCSI sin protección frente a EMC. Las muestras corresponden a un reloj de 40 MHz.

Cambiando el cable expuesto por uno que haga frente a esta magnitud de influencia, se logra enviar efectivamente las tramas de datos por todos los canales.

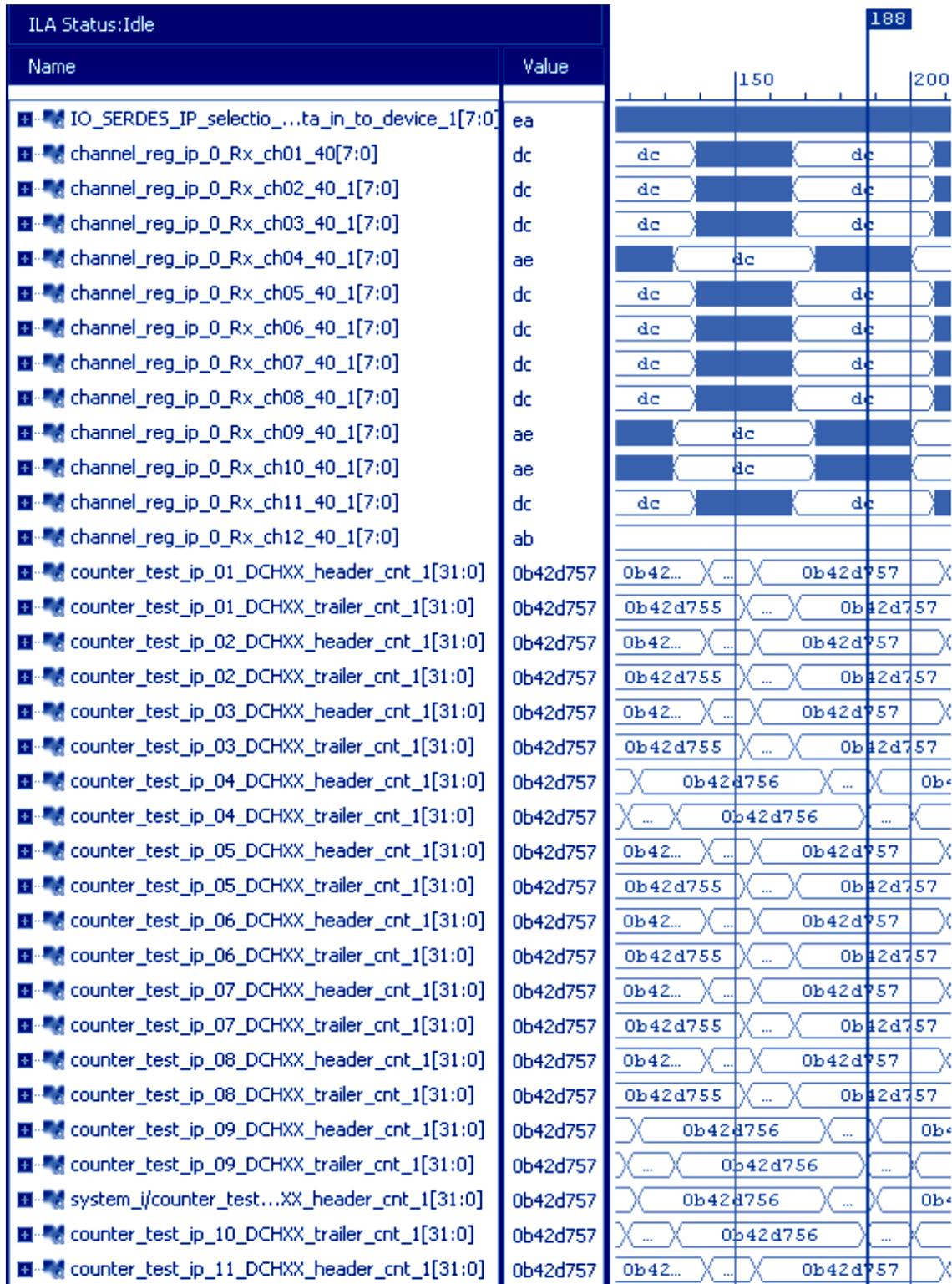


Fig. XII.19 - Formas de onda por ILA capturadas: uso de cable SCSI con protección frente a EMC. Las muestras corresponden a un reloj de 40 MHz, el conteo es cíclico y lo mostrado es el resultado de 24 horas de transmisión.

XII. 4. 4 Diferencia entre la frecuencia de la KC705 y la VC707

La diferencia entre las frecuencias del VC707 (40 MHz) y del KC705 (20 MHz) hace que las primeras tramas recuperadas de FIFO tengan un pequeño desfase en la cantidad de palabras que almacena entre los distintos canales, sin perder información. Tan solo unas tramas más adelante el sistema se ha estabilizado y el desfase es a lo mucho de una trama, la cual queda almacenada en la memoria del bloque FIFO hasta que se habilite la señal de lectura (esto se verifica comparando entre todos los canales el contador que se incluye la cabecera de cada trama).

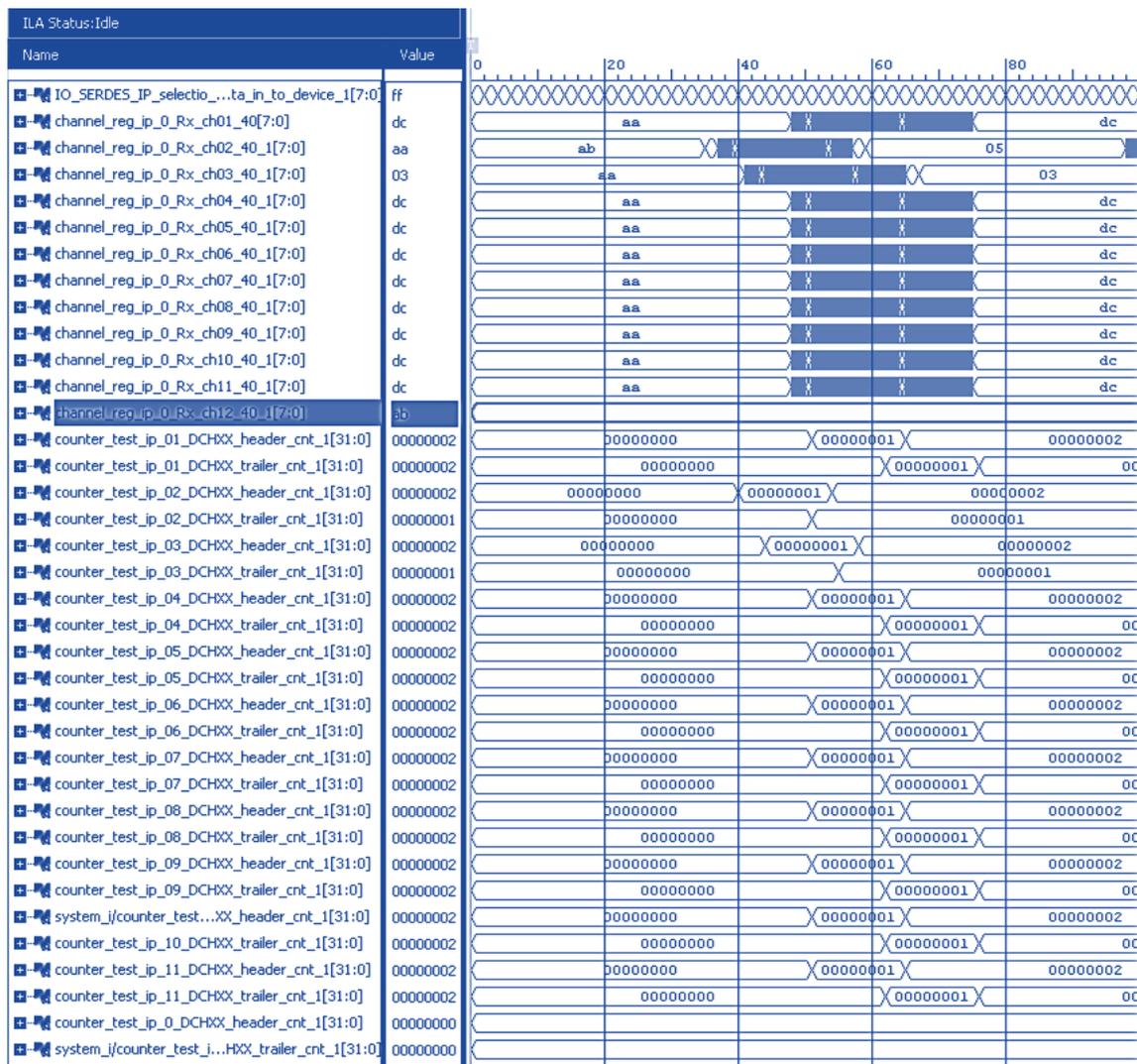


Fig. XII.20 - Formas de onda por ILA capturadas: primera trama recuperada por FIFO. Obsérvese el desfase del canal 2 y canal 3, en tres o cinco tramas después que el sistema se ha estabilizado.

La cantidad de tramas a enviar se ajusta en el bloque IP *DataGenerator_example_ip_v1* antes de su síntesis. Para esta prueba se ajusta el valor a *x"77359400"*, es decir, enviará 2000000000 tramas de 16 bytes (28 GB²⁰ entregados por FIFO del receptor) donde cada byte es generado con una frecuencia

²⁰ Utilizando la relación: 1073741824 bytes = 1 Gigabyte.

de 20 MHz, por lo que la prueba durará aproximadamente 26 minutos con 40 segundos. El bloque IP que genera las tramas de prueba puede ajustarse para que transmita con un bucle infinito que se detendrá hasta que la placa electrónica se reinicie.

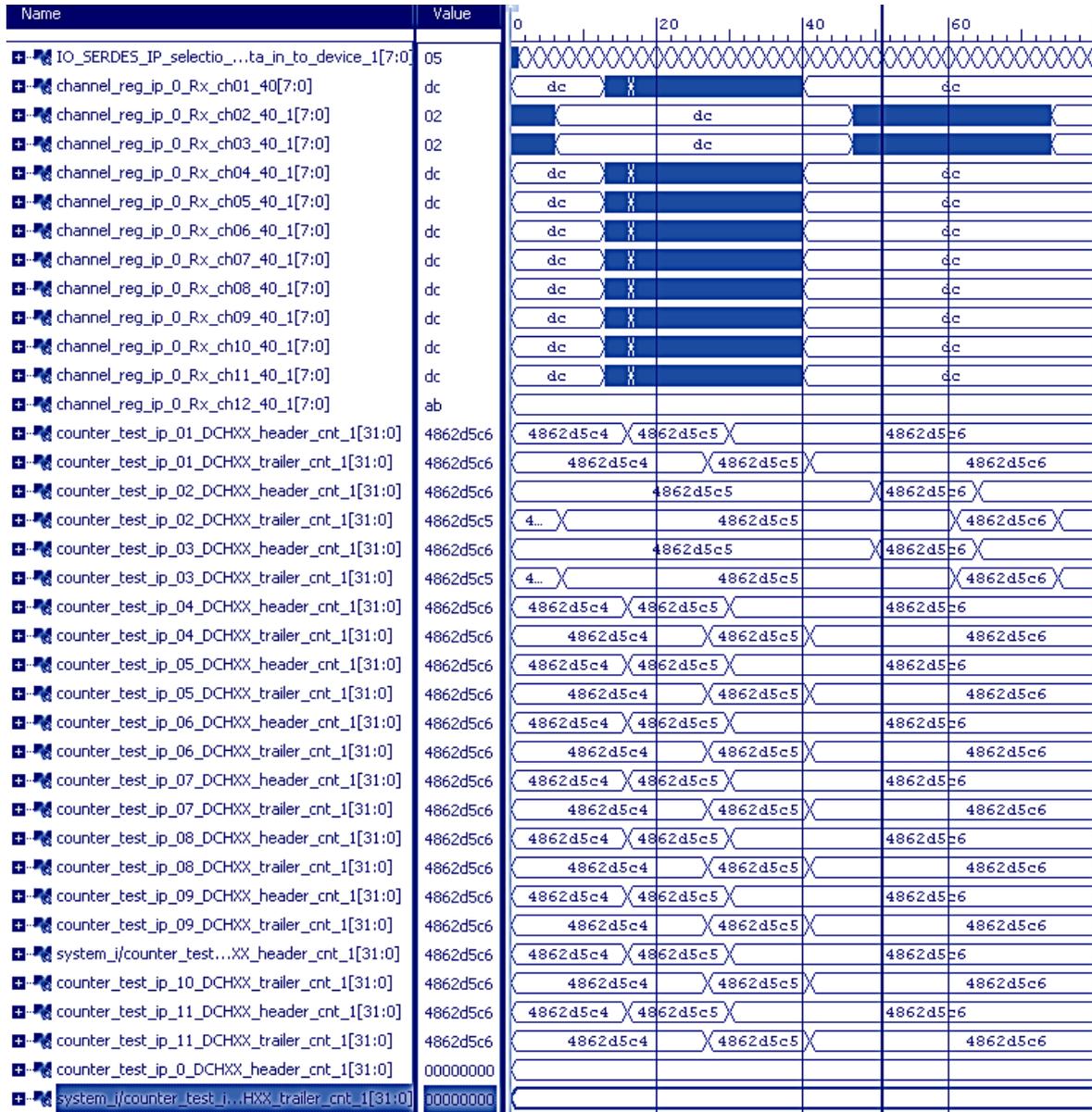


Fig. XII.21 - Formas de onda por ILA capturadas: tramas recuperadas de FIFO. El desfase inicial se mantiene durante toda la transferencia, sin pérdida de datos.

Se puede realizar el mismo cálculo de la BER que se realizó para el FLX.Felix y el INT2HOST. Como no hay pérdida de datos, los bits totales enviados y contados por canal son 256×10^9 , por lo que con nivel de confianza de 95.45 %, la BER es:

$$BER = \frac{-\ln(1-CL)}{N_{bits}} = \frac{-\ln(1-0.9545)}{256 \times 10^9} = 12.1 \times 10^{-11} \quad \text{Ecuación (XII.3)}$$

Si se diera por perdida la trama que ha quedado almacenada en la memoria de la FIFO de los canales 2 y 3, la BER se calcula directamente:

$$BER = \frac{N_{Err}}{N_{bits}} = \frac{16 \times 8}{256 \times 10^9} = 5.0 \times 10^{-10} \quad \text{Ecuación (XII.4)}$$

Name	Value
IO_SERDES_IP_selectio...ta_in_to_device_1[7:0]	ab
channel_reg_ip_0_Rx_ch01_40[7:0]	dc
channel_reg_ip_0_Rx_ch02_40_1[7:0]	dc
channel_reg_ip_0_Rx_ch03_40_1[7:0]	dc
channel_reg_ip_0_Rx_ch04_40_1[7:0]	dc
channel_reg_ip_0_Rx_ch05_40_1[7:0]	dc
channel_reg_ip_0_Rx_ch06_40_1[7:0]	dc
channel_reg_ip_0_Rx_ch07_40_1[7:0]	dc
channel_reg_ip_0_Rx_ch08_40_1[7:0]	dc
channel_reg_ip_0_Rx_ch09_40_1[7:0]	dc
channel_reg_ip_0_Rx_ch10_40_1[7:0]	dc
channel_reg_ip_0_Rx_ch11_40_1[7:0]	dc
channel_reg_ip_0_Rx_ch12_40_1[7:0]	ab
counter_test_ip_01_DCHXX_header_cnt_1[31:0]	77359400
counter_test_ip_01_DCHXX_trailer_cnt_1[31:0]	77359400
counter_test_ip_02_DCHXX_header_cnt_1[31:0]	773593ff
counter_test_ip_02_DCHXX_trailer_cnt_1[31:0]	773593ff
counter_test_ip_03_DCHXX_header_cnt_1[31:0]	773593ff
counter_test_ip_03_DCHXX_trailer_cnt_1[31:0]	773593ff
counter_test_ip_04_DCHXX_header_cnt_1[31:0]	77359400
counter_test_ip_04_DCHXX_trailer_cnt_1[31:0]	77359400
counter_test_ip_05_DCHXX_header_cnt_1[31:0]	77359400
counter_test_ip_05_DCHXX_trailer_cnt_1[31:0]	77359400
counter_test_ip_06_DCHXX_header_cnt_1[31:0]	77359400
counter_test_ip_06_DCHXX_trailer_cnt_1[31:0]	77359400
counter_test_ip_07_DCHXX_header_cnt_1[31:0]	77359400
counter_test_ip_07_DCHXX_trailer_cnt_1[31:0]	77359400
counter_test_ip_08_DCHXX_header_cnt_1[31:0]	77359400
counter_test_ip_08_DCHXX_trailer_cnt_1[31:0]	77359400
counter_test_ip_09_DCHXX_header_cnt_1[31:0]	77359400
counter_test_ip_09_DCHXX_trailer_cnt_1[31:0]	77359400
system_i/counter_test...XX_header_cnt_1[31:0]	77359400
counter_test_ip_10_DCHXX_trailer_cnt_1[31:0]	77359400
counter_test_ip_11_DCHXX_header_cnt_1[31:0]	77359400
counter_test_ip_11_DCHXX_trailer_cnt_1[31:0]	77359400
counter_test_ip_0_DCHXX_header_cnt_1[31:0]	00000000
system_j/counter_test_j...HXX_trailer_cnt_1[31:0]	00000000

Fig. XII.22 - Formas de onda por ILA capturadas: conteo final de tramas recuperadas de FIFO. Como es de esperarse, una de las tramas está almacenada en la FIFO y la señal de lectura no se habilitará hasta que llegue otra.

XIII. PRUEBAS DE VALIDACIÓN Y CERTIFICACIÓN

Los productos de este trabajo no son destinados al comercio sino que tienen su uso en aplicaciones específicas en el área de la investigación. El aseguramiento de la calidad se logra al solicitar productos solamente de fabricantes certificados y proveedores aprobados por el IFIC-Valencia.

Aunque no se hagan pruebas de certificación, se realizan validaciones programando la FPGA de la TilePPr en lugar de la VC707 y conectándola con la placa electrónica principal y el terminal delantero. Estos equipos electrónicos están disponibles en el laboratorio TileCal del IFIC-Valencia y son los mismos que se instalan en los cajones del TileCal. Cualquier ensayo que se haga con ellos es igual²¹ que hacerlo directamente en el CERN.

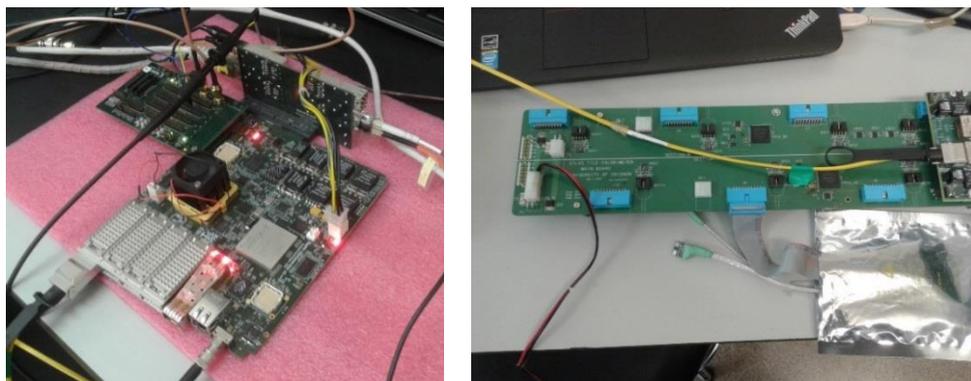


Fig. XIII.1 - Vista del TilePPr desarrollado por el IFIC-Valencia y de la tarjeta electrónica principal que envía las señales del QSFP1 y del integrador. Nótese que la pequeña tarjeta electrónica conectada a la tarjeta electrónica principal es la tarjeta con el ADC para simular las lecturas del integrador.

²¹ Estimar la reproducibilidad (según VIM) de cada ensayo queda fuera del alcance de este trabajo. Sin embargo, se han obtenido los mismos resultados programando las FPGA ya instaladas en el CERN gracias al equipo TileCal de IFIC-Valencia.

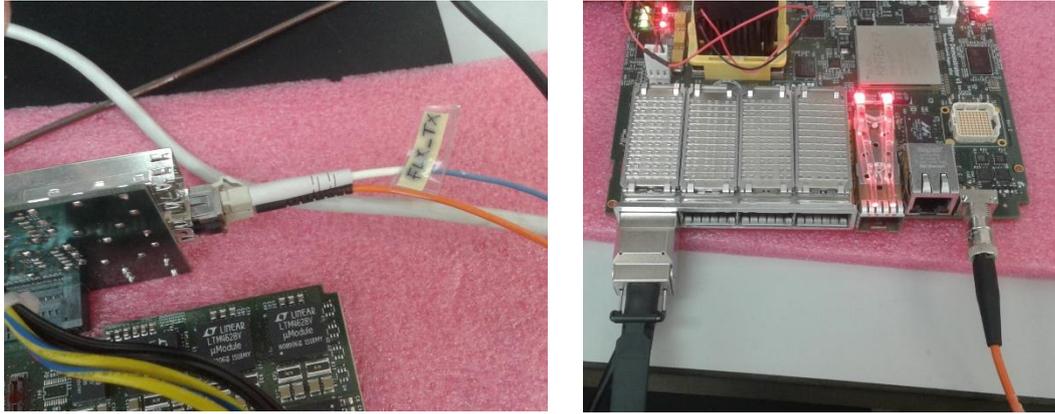


Fig. XIII.2 - Vista de las conexiones por fibra al TilePPR. A la izquierda, conexión SFP desde y hacia el TileDAQ del FELIX. A la derecha, conector QSFP1 y fibra conectada al TTCex.

XIII. 1. FLX.Felix

XIII. 1. 1 Respuesta del sistema ante distintas frecuencias de L1A

Similares a las pruebas experimentales, con la diferencia que se utilizan tramas de 788 bytes de anchura con datos provenientes del QSFP1. De la misma manera, el integrador utiliza tramas de datos de 34 bytes de anchura, presentes durante toda la transmisión pero limitadas únicamente por la activación de los eproc por la función *elinkconfig* del *setup.sh*.

TABLA XIII.1 Condiciones para la variación de la frecuencia de L1A en pruebas de certificación

Código	Frecuencia de L1A	Rutas a verificar	Duración de la muestra
T_03_01	1 Hz	GBT Link 0 Egroup 0 epath1: 8 (0) eproc8bits	1000 s
T_03_02	1 kHz		10 s
T_03_03	100 kHz		100 s
T_03_04	100 kHz	1 eprocs 8 bits 2 eprocs 2 bits	100 s
Carpeta del proyecto:		proyecto_integrador_00	

TABLA XIII.2 Resultados de la variación de la frecuencia de L1A en pruebas de validación

Código de la prueba	Ruta	Requisito	Resultado
			Medido
T_03_01	eproc8	Bloques con errores	0
		Bloques sin errores	776
		Tamaño total en bytes	787976
		Cantidad de nChunks completos	999
		Velocidad	0.0 MB/s
T_03_02	eproc8	Bloques con errores	0
		Bloques sin errores	7745
		Tamaño total en bytes	7864512
		Cantidad de nChunks completos	9980
		Velocidad	0.8 MB/s
T_03_03	eproc8	Bloques con errores	0
		Bloques sin errores	2963146
		Tamaño total en bytes ²²	3008869994
		Cantidad de nChunks completos	3818360
		Velocidad	30.0 – 30.7 MB/s
T_03_04	eproc8 eproc2	Bloques con errores	0
		Bloques sin errores	2963078 (total eproc8) 154300 (total eproc 2)
		Tamaño total en bytes	3008800944 (eproc8) 148810170 (eproc2)
		Cantidad de nChunks completos	3818273 (total eproc8) 4133614 (total eproc 2)
		Velocidad	31.6 – 32.5 MB/s

²² Divididos en tres archivos por el servidor local: primer archivo 1048635 bloques, segundo archivo 1049114 bloques, tercer archivo 865397 bloques.

TABLA XIII.3 Tasa de error binario según las pruebas de validación

Código de la prueba	bytes	Nbits	BER	CL
T_03_01	787976	6303808	5.00 x10 ⁻⁰⁷	0.96
T_03_02	7864512	62916096	5.00 x10 ⁻⁰⁸	0.96
T_03_03	3008869994	24070959952	1.30 x10 ⁻¹⁰	0.96
T_03_04 (eproc 2 bits)	3008800944	24070407552	1.30 x10 ⁻¹⁰	0.96
T_03_04 (eproc 2 bits)	148810170	1190481360	2.70 x10 ⁻⁰⁹	0.96

XIII. 2. INT2HOST

Las placas electrónicas que se conectan a la placa electrónica principal pueden ser programadas para enviar datos como si los adquiriera el DAQ del integrador. En esta prueba los datos enviados forman una función de pendiente conocida, aunque los mismos se actualizan con una frecuencia mucho menor a los procesos con los que funciona el módulo del integrador_receiver.vhd.

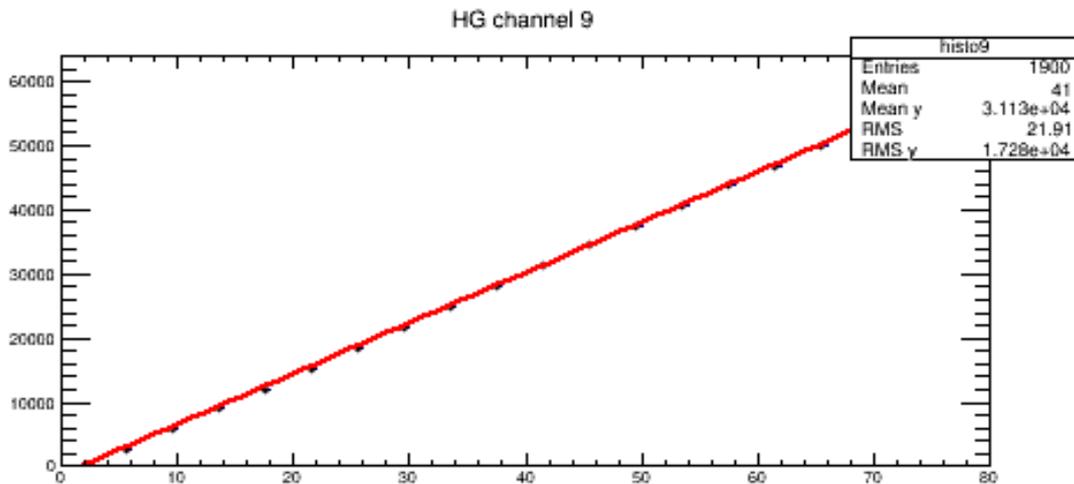


Fig. XIII.3 - Gráfica de valores que se obtienen del módulo ADC conectado al canal 9 (lado B) de la placa electrónica principal. Para ejecutar la generación y lectura de los datos por el IPBus se utiliza el comando python Integrator_readout_All_del.py desde el terminal conectado por cable de red al integrador.

No	Cont Lado	Lado	Cont 0	Datos 0	Cont 1	Datos 1	Cont 2	Datos 2	Cont 3	Datos 3	Cont 4	Datos 4	Cont 5	Datos 5
1	8a00	000a	8a00	71d0	8a01	f521	8a01	6fa8	8a01	73c8	8a00	7008	8a00	7000
2	8a01	000a	8a01	71d0	8a02	f521	8a02	6fa8	8a01	73c8	8a01	7008	8a01	7000
3	8a02	000a	8a02	71d0	8a03	f521	8a03	6fa8	8a03	73c8	8a02	7008	8a02	7000
4	8a03	000a	8a03	71d0	8a04	f521	8a04	6fa8	8a03	73c8	8a03	7008	8a03	7000
5	8a04	000a	8a04	71d0	8a05	f521	8a05	6fa8	8a05	73c8	8a04	7008	8a04	7000
6	8a05	000a	8a05	71d0	8a06	f521	8a06	6fa8	8a05	73c8	8a05	7008	8a05	7000
7	8a06	000a	8a06	71d0	8a07	f521	8a07	6fa8	8a07	73c8	8a06	7008	8a06	7000
8	8a07	000a	8a07	71d0	8a08	f521	8a08	6fa8	8a07	73c8	8a07	7008	8a07	7000
9	8a08	000a	8a08	71d0	8a09	f521	8a09	6fa8	8a09	73c8	8a08	7008	8a08	7000
10	8a09	000a	8a09	71d0	8a0a	f521	8a0a	6fa8	8a09	73c8	8a09	7008	8a09	7000
11	8a0a	000a	8a0a	71d0	8a0b	f521	8a0b	6fa8	8a0b	73c8	8a0a	7008	8a0a	7000

No	Cont Lado	Lado	Cont 0	Datos 0	Cont 1	Datos 1	Cont 2	Datos 2	Cont 3	Datos 3	Cont 4	Datos 4	Cont 5	Datos 5
1	8f50	000b	8f50	7200	8f51	f4df	8f51	7008	8f52	7400	8f50	7008	8f50	7220
2	8f51	000b	8f51	7200	8f52	f4df	8f52	7008	8f52	7400	8f51	7008	8f51	7220
3	8f52	000b	8f52	7200	8f53	f4df	8f53	7008	8f54	7400	8f52	7008	8f52	7220
4	8f53	000b	8f53	7200	8f54	f4df	8f54	7008	8f54	7400	8f53	7008	8f53	7220
5	8f54	000b	8f54	7200	8f55	f4df	8f55	7008	8f56	7400	8f54	7008	8f54	7220
6	8f55	000b	8f55	7200	8f56	f4df	8f56	7008	8f56	7400	8f55	7008	8f55	7220
7	8f56	000b	8f56	7200	8f57	f4df	8f57	7008	8f58	7400	8f56	7008	8f56	7220
8	8f57	000b	8f57	7200	8f58	f4df	8f58	7008	8f58	7400	8f57	7008	8f57	7220
9	8f58	000b	8f58	7200	8f59	f4df	8f59	7008	8f5a	7400	8f58	7008	8f58	7220
10	8f59	000b	8f59	7200	8f5a	f4df	8f5a	7008	8f5a	7400	8f59	7008	8f59	7220
11	8f5a	000b	8f5a	7200	8f5b	f4df	8f5b	7008	8f5c	7400	8f5a	7008	8f5a	7220

Fig. XIII.4 - Lecturas individuales del lado A y el lado B del integrador enviadas por el sistema FELIX. Una vez desempaquetado y convertido a un archivo en formato .txt, cada fragmento se ha dejado en una fila de MS-Excel para hacer los análisis estadísticos correspondientes.

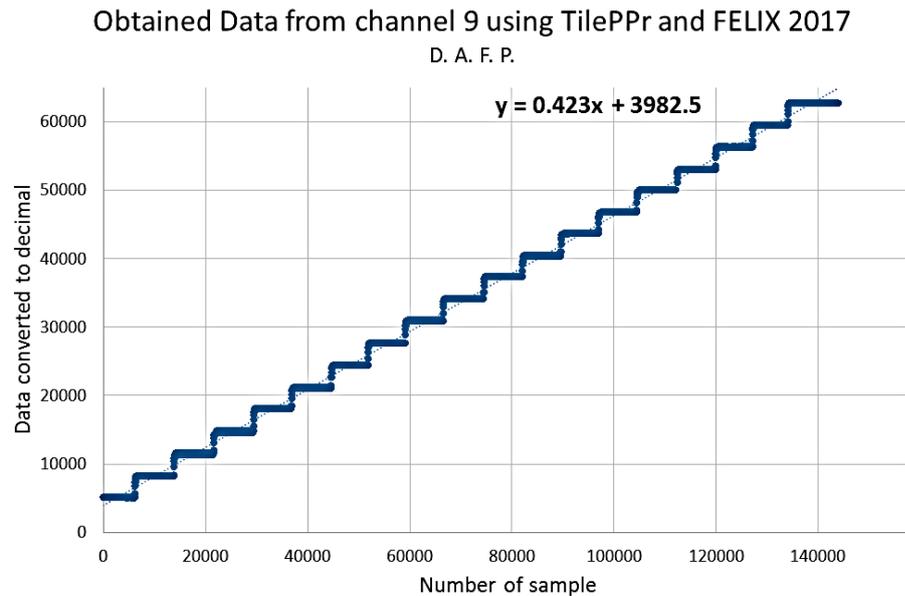


Fig. XIII.5 - Gráfica de valores que se obtienen del módulo ADC conectado al canal 9 (lado B) a través de FELIX. La velocidad con la que se actualizan los datos del ADC es de una muestra por cada 890 μ s y la velocidad de los procesos del integrador_receiver.vhd es de 80 MHz.

Los datos enviados en formato hexadecimal por la computadora del servidor local coinciden con los adquiridos por la función *fdac*.

XIII. 3. FE-I4

Para la prueba de validación, se deja que el sistema transfiera durante un intervalo de tiempo de aproximadamente 24 horas (1440 minutos) enviando

108000000000 tramas de 16 bytes (1512 GB entregados por FIFO del receptor).
Los resultados llevan a calcular la BER con el resultado:

$$BER = \frac{-\ln(1-CL)}{N_{bits}} = \frac{-\ln(1-0.9545)}{13.824 \times 10^{12}} = 2.23 \times 10^{-13} \quad \text{Ecuación (XIII.1)}$$

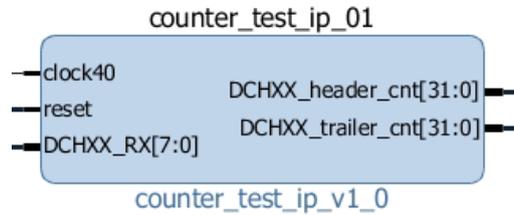


Fig. XIII.6 - Bloque IP creado en este trabajo: counter_test_ip_01. Cuenta las cabeceras y colas de las tramas de datos con una señal de reloj de 40 MHz. Una vez que detecta una cabecera, no cuenta ninguna otra si no detecta una cola previamente.

Las placas electrónicas de prueba son sometidas a pruebas eléctricas gratuitas proporcionadas por el fabricante. En estas pruebas, se aplica tensión de medición para comprobar el circuito para detectar rupturas o cortocircuito.

XIV. MANUAL DE USUARIO

Cada parámetro configurado en el TileDAQ puede ser ajustado según el ensayo o protocolo de comunicación que se planea realizar, tantas son las combinaciones que para asegurar la repetibilidad en las pruebas se establece un procedimiento de puesta en marcha del sistema FELIX donde se ha comprobado la transferencia exitosa de datos.

El documento donde se guarda el procedimiento es el “FELIX STARTUP OPERATION STEP BY STEP GUIDE”, el mismo pretende poseer la mayoría de las características de los procedimientos técnicos que debe tener un laboratorio de ensayos acreditado por la norma ISO/IEC 17025:2009, dejando una referencia para futuras documentaciones y un mayor acercamiento al sistema de calidad.

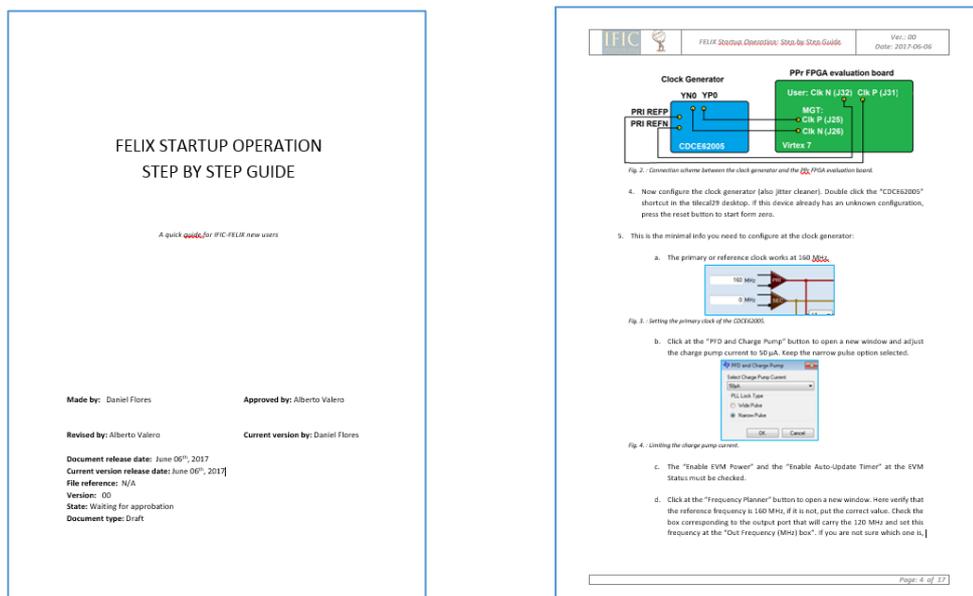
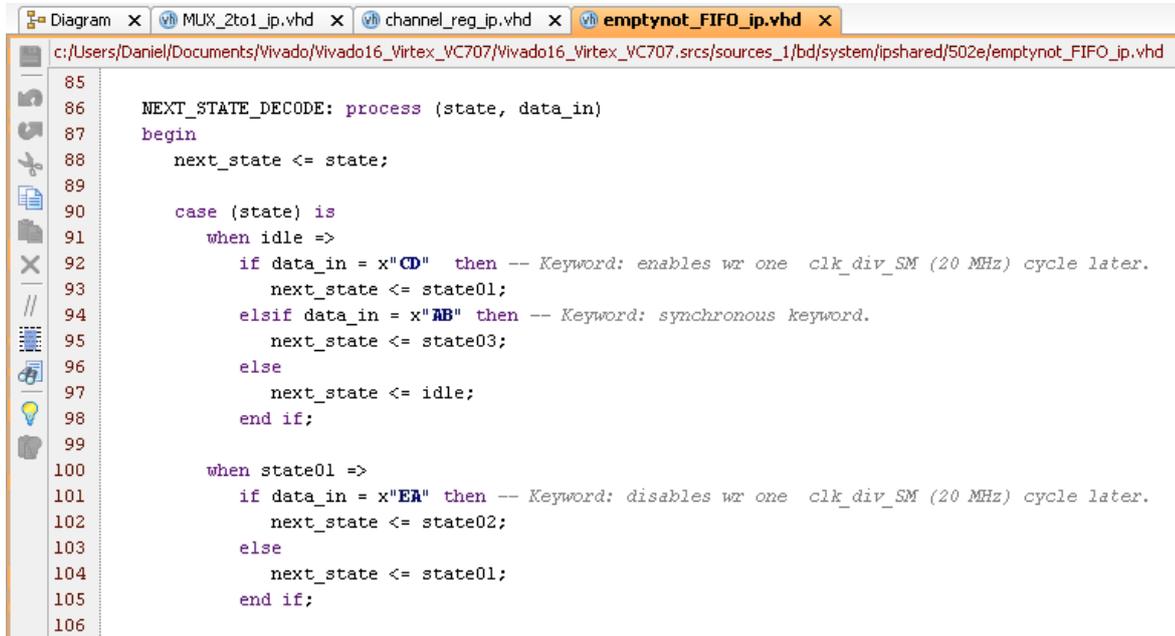


Fig. XIV.1 - Referencia documental para la puesta en marcha del sistema FELIX en el IFIC-Valencia.

Respecto a los nuevos módulos que incluye este trabajo, poseen instrucciones claras para su uso en los archivos .vhd que los conforman, escritas en idioma inglés y dirigidas a los miembros del Proyecto TileCal y cooperadores del CERN.



```
85
86 NEXT_STATE_DECODE: process (state, data_in)
87 begin
88     next_state <= state;
89
90     case (state) is
91     when idle =>
92         if data_in = x"CD" then -- Keyword: enables wr one clk_div_SM (20 MHz) cycle later.
93             next_state <= state01;
94         elsif data_in = x"AB" then -- Keyword: synchronous keyword.
95             next_state <= state03;
96         else
97             next_state <= idle;
98         end if;
99
100    when state01 =>
101        if data_in = x"EA" then -- Keyword: disables wr one clk_div_SM (20 MHz) cycle later.
102            next_state <= state02;
103        else
104            next_state <= state01;
105        end if;
106
```

Fig. XIV.2 – Captura con las instrucciones de las palabras clave que controlan los estados del bloque IP emptynot_FIFO.

XV. COSTES DE FABRICACIÓN

Se comparte un resumen sobre el costo fijo para ejecutar un proyecto de este tipo, el mayor porcentaje del gasto corresponde al servidor y a las dos tarjetas electrónicas de evaluación. Para distinguir un poco entre los costos que lleva el equipamiento utilizado para las pruebas (los cuales fueron cubiertos antes de iniciar este trabajo)²³ y los costos adicionales para cumplir los objetivos de este trabajo, se agrupan en las siguientes tablas:

TABLA XV.1 Horas de trabajo ²⁴

Fecha de precios: agosto de 2017			
Descripción	salario/día (€)	Cantidad estimada de días	Salario (€)
Honorarios profesionales	35.39	100	3539.00
		Total (€)	3539.00

²³ Un componente de gran valor que no fue incluido en la tabla es el TTCex ni el TilePPr.

²⁴ Referencia de 1.5 SMI, es decir 1415.20 € según la página <http://www.salariominimo.es/> con fecha 28 de agosto de 2017.

TABLA XV.2 Coste de los componentes para la placa electrónica de prueba del FE-14

Fecha de precios: 13 de julio de 2017			
Descripción	Precio Unitario (€)	Cantidad	Precio (€)
Conector HDMI (PCB)	1.12	16	17.92
Conector ASP-134602-01	19.36	4	77.44
Conector DisplayPort (PCB)	2.79	12	33.48
Conector SCSI ULTRA 3700-0200-15	7.7	4	30.80
Cable display port (5 m)	16.28	3	48.84
Cable HDMI	4.54	3	13.62
Cable SCSI	9.00	1	9.00
Placa electrónica de pruebas	52.92	4	211.68
Total (€)			442.78

TABLA XV.3 Coste estimado para el equipamiento básico del laboratorio TileCal

Fecha de precios: 27 de abril de 2017			
Descripción	Precio Unitario (€)	Cantidad	Precio (€)
Tarjeta de referencia	3495.00	2	6990.00
Tarjeta secundaria	190.00	2	380.00
Generador de reloj con limpieza de jitter	209.00	1	209.00
Osciloscopio	1200.00	1	1200.00
Servidor local (Host o PC)	8460.01	1	8460.01
Envío de servidor local	131.97	1	131.97
Disco rígido del servidor (250 GB)	190.58	2	381.16
Garantía del disco rígido del servidor	23.62	2	47.24
Memoria RAM para el servidor	70.82	1	70.82
Mouse	9.95	2	19.90
Teclado	8.83	1	8.83
Computadora portátil (LT)	429.00	1	429.00
Monitor	75.00	1	75.00
Latiguillo coaxial hembra a hembra tipo SMA	11.47	12	137.64
Transceptor SFP-10G-SR	22.13	4	88.52
Latiguillo de fibra óptica (20 m) LC a LC	16.32	4	65.28
Cable USB-mini USB	3.72	4	14.88
Cable de red Cat 5-e	26.00	4	104.00
Sub-total			18813.25
Imprevistos y contingencias	10 % del sub-total		1881.33
Caja chica	5 % del sub-total		940.66
Total (€)			21635.24

XVI. CONCLUSIONES Y TRABAJOS FUTUROS

Entre las experiencias y conclusiones que han llevado consigo el desarrollo y la culminación de este trabajo están:

- Las nuevas tecnologías en las transferencias de datos por fibra óptica proveen una gran velocidad que es limitada básicamente por la electrónica del emisor y receptor. Aunque se dispone de tal capacidad, en el área de la investigación, millones de muestras que se registran por segundo saturarían fácilmente los servidores locales con datos redundantes, por lo que la capacidad de seleccionar y discriminar información antes de ser enviada al servidor es tan importante como la máxima velocidad de transferencia.
- Todo el hardware programable en la FPGA síncrono a una única señal de reloj de 40 MHz y a sus derivadas, hacen posible que se puedan manejar distintas tecnologías dentro de un mismo circuito integrado, haciendo una transición suave entre generaciones, verificando y haciendo ajustes con simple programación en vhdl.
- Es primordial hacer una planificación de los recursos hardware según las regiones de reloj de la FPGA. El manejo de los pares MRCC y SRCC en el bloque de entradas y salidas facilita la administración de múltiples canales y sus *delays* de entrada.
- En la deserialización se evidencia las diferencias de fase y frecuencia entre los osciladores de dos placas distintas, aún cuando se han programado a la misma frecuencia nominal. Esta pequeña diferencia toma importancia cuando se transfiere una gran cantidad de tramas, sobre todo si la longitud de las mismas también es notable.

- Se necesitan de FIFOs con diferente señal de reloj para escritura y lectura cuando se trabaja con distintos dominios de reloj: el externo recuperado con los datos serializados y el propio del sistema. Los mecanismos de control para su llenado y vaciado toman mayor importancia cuando la frecuencia de la señal de reloj final es diferente a la frecuencia con la cual fueron generados los datos.
- Los conceptos de las tecnologías de la comunicación como el diseño de pistas y vías para señales diferenciales complementados con software de uso libre permiten diseñar placas electrónicas básicas funcionales en el orden de los MHz, a pesar de que no se dispongan de simuladores para evaluar la integridad de la señal. Con diseños más complejos y/o a mayores frecuencias, lo mejor es invertir un poco en estas herramientas.
- El desarrollo de proyectos ambiciosos de investigación es posible gracias al esfuerzo, estudio y cooperación de cada uno de los miembros en un equipo de trabajo como el del grupo TileCal en IFIC-Valencia.

Respecto a los trabajos futuros:

- La versión en VIVADO de la programación hardware de la TilePPr. La herramienta provee de soluciones para su importación, es labor del usuario verificar la correcta importación de fichero, versiones de bloques IP, conversión de *constraints* .ucf a .xdc a través de PlanAhead, revisión de todos los *delays* y tiempos en la versión .xdc, etc.
- Nueva versión de la placa electrónica FMC dedicada al terminal seleccionado para el FE-I4. Los mismos bloques IP de VIVADO pueden utilizarse con diferentes declaraciones en el archivo .xdc.
- Interfaz software que maneje las tres funcionalidades presentes en este trabajo: la interfaz FELIX, el integrador y el FE-I4. Puede ser idéntico a la funcionalidad *elinkconfig* del servidor local FELIX.
- Evaluar la importancia de los ensayos requeridos de EMC que puedan realizarse en un laboratorio acreditado. Aunque la placa electrónica es pasiva, se puede evaluar su respuesta ante la presencia de campos electromagnéticos.
- Evaluar las posibilidades de los pines I/O: son configurables como entradas y salidas por el usuario. El proyecto VIVADO para transmisión y recepción es adaptable a cualquier asignación de puertos, por lo que existen muchas configuraciones entre los terminales de entrada y salida que pueden utilizarse y que no se han probado en este trabajo.
- Planificar los *delays* en las interfaces I/O con la nueva placa electrónica.
- El trabajo conjunto entre la Universidad Nacional Autónoma de Honduras con el IFIC-Valencia, específicamente con el proyecto TileCal, para lo cual ya se han solicitado 25 licencias de VIVADO Design Suite. Diseños

electrónicos que se requieran en el proyecto u otro aporte pueden presentarse como trabajos finales por los estudiantes de la Facultad de Ingeniería de la UNAH, así como programas de intercambio o seminarios.

XVII. REFERENCIAS BIBLIOGRÁFICAS

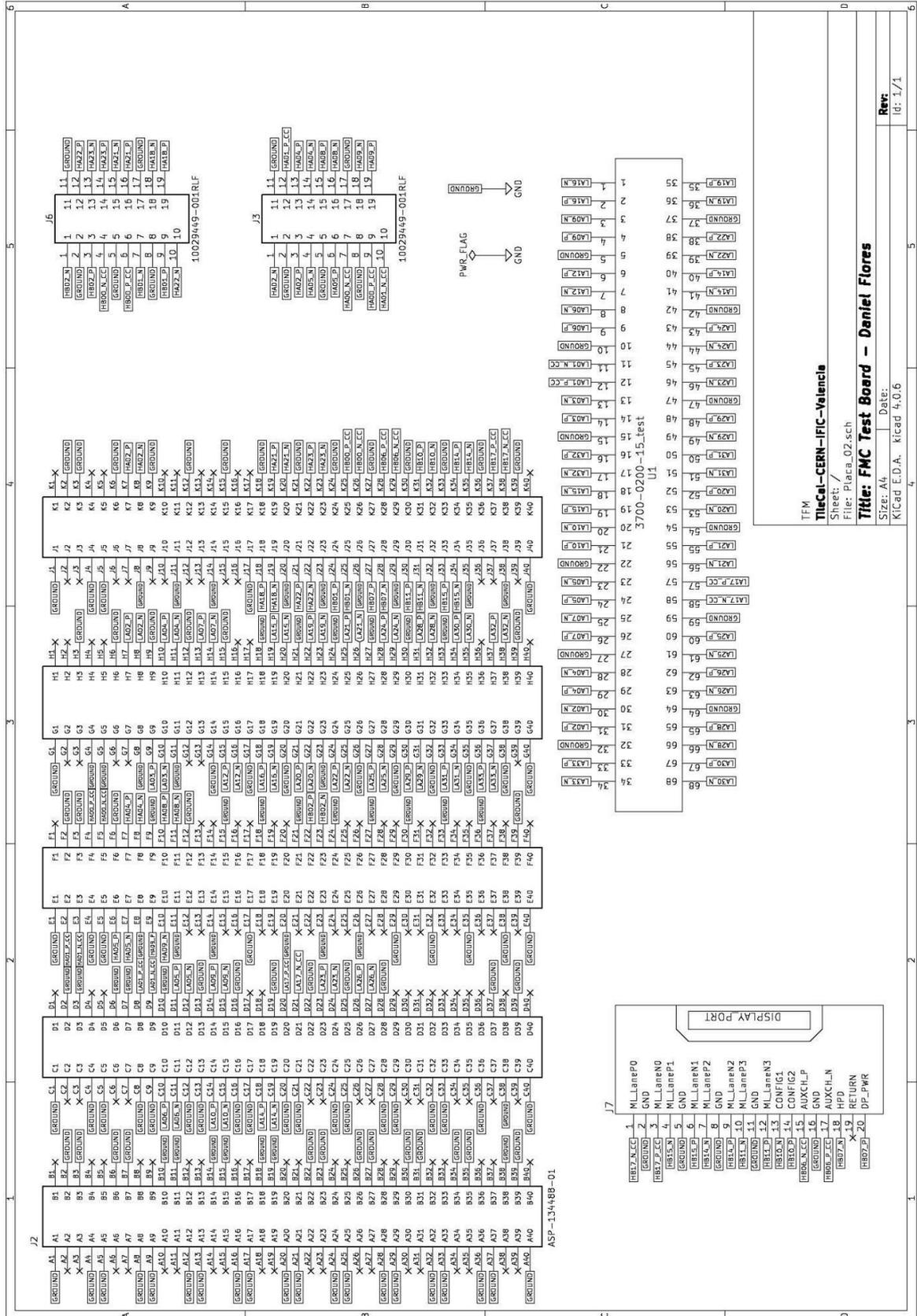
- [1] Xilinx, «VC707 Evaluation Board for the Virtex-7 FPGA: User Guide,» 12 Agosto 2016. [En línea].
https://www.xilinx.com/support/documentation/boards_and_kits/vc707/ug885_VC707_Eval_Bd.pdf.
- [2] Xilinx, «VC709 Evaluation Board for the Virtex-7 FPGA: User Guide,» 12 Agosto 2016. [En línea].
https://www.xilinx.com/support/documentation/boards_and_kits/vc709/ug887-vc709-eval-board-v7-fpga.pdf.
- [3] Xilinx, «KC705 Evaluation Board for the Kintex-7 FPGA: User Guide,» 8 Julio 2016. [En línea].
https://www.xilinx.com/support/documentation/boards_and_kits/kc705/ug810_KC705_Eval_Bd.pdf.
- [4] Texas Instruments, «CDCE62005 3:5 Clock Generator, Jitter Cleaner with Integrated Dual VCOs,» Julio 2016. [En línea].
<http://www.ti.com/lit/ds/symlink/cdce62005.pdf>.
- [5] Textronix, «Tektronix MSO4054 Datasheet,» 28 Octubre 2009. [En línea].
<http://www.testequipmenthq.com/datasheets/TEKTRONIX-MSO4054-Datasheet.pdf>.
- [6] Conseil Européen pour la Recherche Nucléaire - CERN, «ATLAS Experiment / Discover,» [En línea]. <https://atlas.cern/discover>.
- [7] Instituto de Física Corpuscular, «ATLAS TileCal Valencia Group,» [En línea].
<https://ific.uv.es/tical/>.
- [8] E. Valdes Santurio, «Upgrade of Tile Calorimeter of the ATLAS Detector,» 15 Junio 2016. [En línea]. <https://cds.cern.ch/record/2160251/files/ATL-TILECAL-PROC-2016-005.pdf>.
- [9] C. Fischer, «Study of TileCal Scintillator Irradiation using the Minimum Bias Integrators,» 13 Junio 2016. [En línea].
<https://cds.cern.ch/record/2160267/files/ATL-TILECAL-PROC-2016-006.pdf>.

- [10] Conseil Européen por la Recherche Nucléaire - CERN, «The FE-I4B Integrated Circuit Guide,» 30 Diciembre 2012. [En línea]. https://indico.cern.ch/event/261840/contributions/1594374/attachments/462649/641213/FE-I4B_V2.3.pdf.
- [11] American National Standards Institute, Inc., *ANSI/VITA 57.1-2008*, VMEbus International Trade Association, 2008.
- [12] R. Leventhal, «IBIS Model Syntax,» [En línea]. http://extras.springer.com/2006/978-0-387-24159-3/Appendix%20E%20Key%20Concepts%20of%20the%20IBIS%20Model/Appendix%20E1%20IBIS_Syntax.pdf. [Último acceso: 2017].
- [13] H. C. Manchester, «30 AWG Shielded SCSI NexZ0™,» [En línea]. <http://www.hca.hitachi-cable.com/products/hca/products/round/SCSI/pdfs/30-AWG-Shielded-SCSI-Nexzo.pdf>. [Último acceso: 2017].
- [14] Xilinx, «Xilinx 7 Series FPGA and Zynq-7000 All Programmable SoC Libraries Guide for HDL Designs,» 2 Octubre 2013. [En línea]. https://www.xilinx.com/support/documentation/sw_manuals/xilinx14_77series_hdl.pdf.
- [15] F. Contu, «The Do's and Don'ts of High Speed Serial Design in FPGAs,» 2013. [En línea]. http://www.keysight.com/upload/cmc_upload/All/5XilinxTheDos.pdf?&cc=ES&lc=eng.
- [16] KEYSIGHT Technologies, «Total Jitter Measurement at Low Probability Levels, Using Optimized BERT Scan Method,» 2005. [En línea]. <http://literature.cdn.keysight.com/litweb/pdf/5989-2933EN.pdf?id=682410>.
- [17] Joint Committee for Guides in Metrology - BIPM, «JCGM 100:2008 Evaluation of measurement data — Guide to the expression of uncertainty in measurement,» Septiembre 2008. [En línea]. www.bipm.org/utis/common/documents/jcgm/JCGM_100_2008_E.pdf.
- [18] A. Valero Biot, «Data acquisition and processing in the ATLAS Tile Calorimeter Phase-II Upgrade Demonstrator,» 02 Febrero 2017. [En línea]. <https://cds.cern.ch/record/2243768/files/ATL-TILECAL-PROC-2017-001.pdf>.
- [19] A. Valero Biot, «A new read-out architecture for the ATLAS Tile Calorimeter,» 21 Diciembre 2015. [En línea]. <https://cds.cern.ch/record/2117093/files/ATL-TILECAL-PROC-2015-025.pdf>.
- [20] S. Bonacini, K. Kloukinas y P. Moreira, «e-link: A Radiation-Hard Low-Power Electrical Link for Chip-to-Chip Communication,» 09 Febrero 2010. [En línea]. <https://cdsweb.cern.ch/record/1235849/files/p422.pdf>.
- [21] «GBT-FPGA Tutorial - 1st edition,» 27 Junio 2016. [En línea]. <https://indico.cern.ch/event/532419/>.
- [22] T. Instruments, «HDMI Design Guide - High-Speed Interface Products,» Junio 2007. [En línea]. https://e2e.ti.com/cfs-file/__key/telligent-evolution-components-attachments/00-138-01-00-00-10-65-80/Texas-Instruments-HDMI-Design-Guide.pdf.
- [23] R. Pro, «RS-Online: Latiguillo SCSI RS Pro 2m,» [En línea]. <http://docs-europe.electrocomponents.com/webdocs/1586/0900766b81586579.pdf>. [Último acceso: 2017].

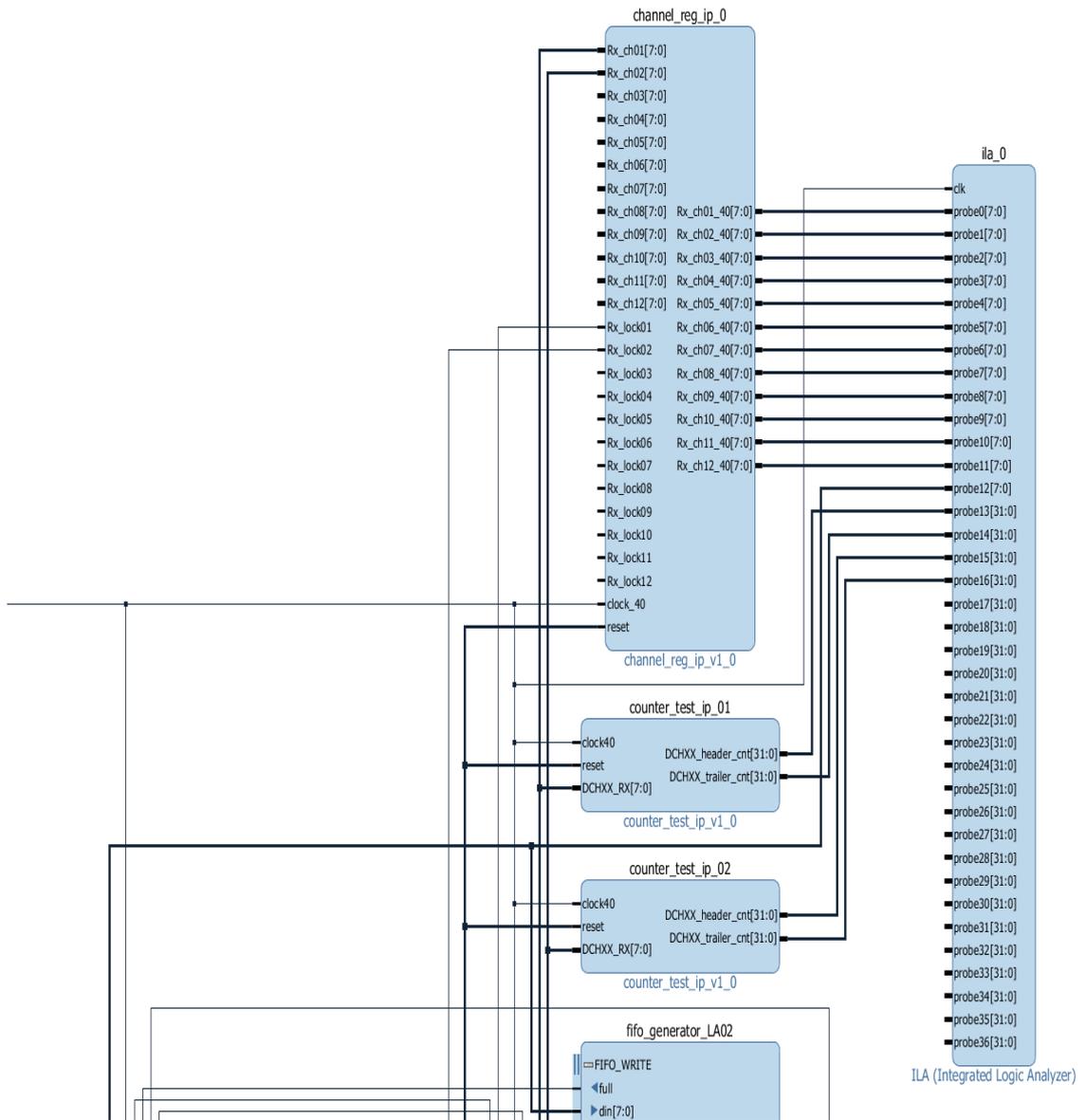
- [24] S. Ryu, «FELIX: The detector readout upgrade of the ATLAS experiment,» 6 Agosto 2015. [En línea]. <https://cds.cern.ch/record/2048651/files/ATL-DAQ-SLIDE-2015-558.pdf?version=1>.
- [25] Xilinx, «7 Series FPGAs: User Guide,» 1 Marzo 2017. [En línea]. https://www.xilinx.com/support/documentation/user_guides/ug472_7Series_Clocking.pdf.
- [26] Würth Elektronik, «Guía de Diseño WEdirekt para circuitos impresos de la tienda online,» [En línea]. http://www.wedirekt.es/web/media/16_wedirekt/spezifikationen/leiterplatten_daten_downloads/Basic_Design-Guide_0117_ES_web.pdf. [Último acceso: 2017].
- [27] Xilinx, «Virtex-7 T and XT FPGAs Data Sheet: DC and AC Switching Characteristics,» 06 Abril 2017. [En línea]. https://www.xilinx.com/support/documentation/data_sheets/ds183_Virtex_7_Data_Sheet.pdf.
- [28] Xilinx, «FIFO Generator v13.1: LogiCORE IP Product Guide,» 05 Abril 2017. [En línea]. https://www.xilinx.com/support/documentation/ip_documentation/fifo_generator/v13_1/pg057-fifo-generator.pdf.
- [29] FCIconnect, «HDMI Type A receptacle,» 15 Junio 2012. [En línea]. <http://portal.fciconnect.com/Comergent/fci/drawing/10029449.pdf>.
- [30] SAMTEC, «ASP-134486-01,» 11 Septiembre 2007. [En línea]. <http://suddendocs.samtec.com/prints/asp-134486-01-mkt.pdf>.
- [31] MOLEX Incorporated, «SD-47272-001,» 26 Marzo 2010. [En línea]. http://www.molex.com/pdm_docs/sd/472720001_sd.pdf.
- [32] MH CONNECTORS LTD, «3700-0200-15,» [En línea]. <http://datasheet.octopart.com/3700-0200-15-MH-Connectors-datasheet-28359889.pdf>. [Último acceso: 2017].

XVIII. PLANOS Y ANEXOS

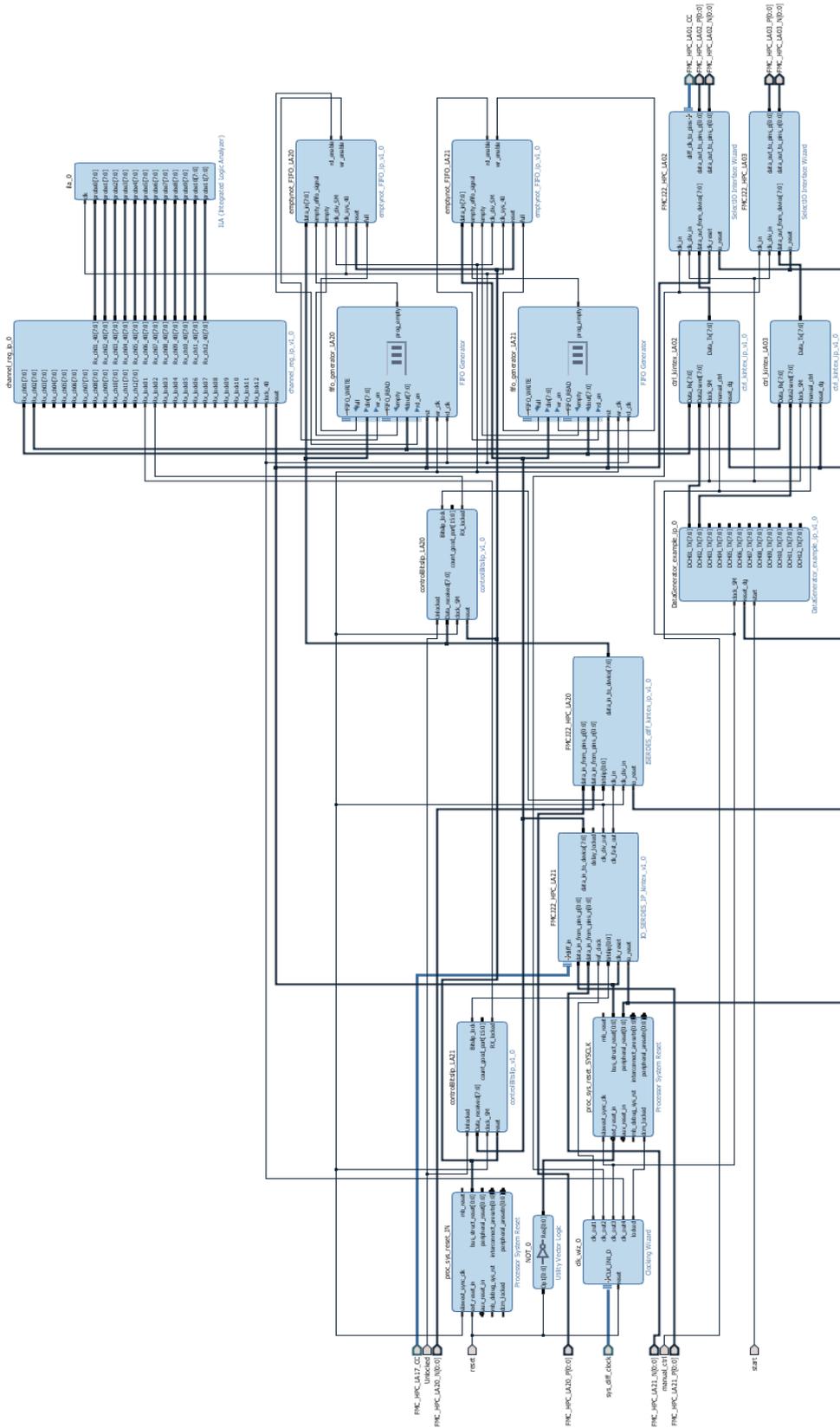
ANEXO I. Vista del esquemático de la placa de pruebas.



ANEXO III. Interface FE-I4 en el VC707: validación de dos canales RX

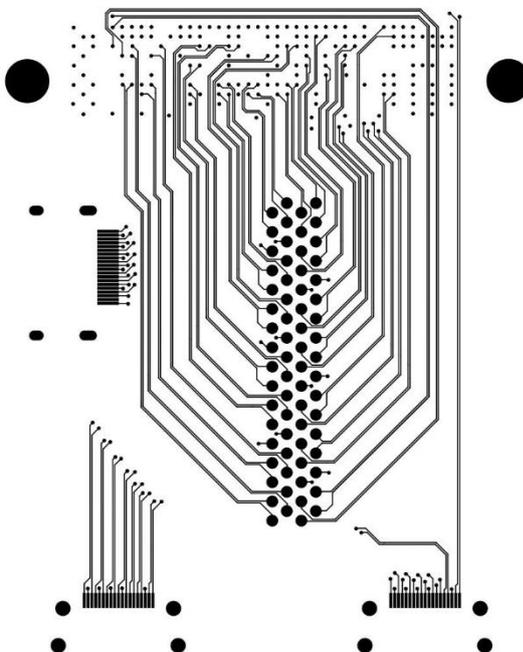


ANEXO IV. Interface FE-I4 en el KC705: dos canales RX y dos canales TX

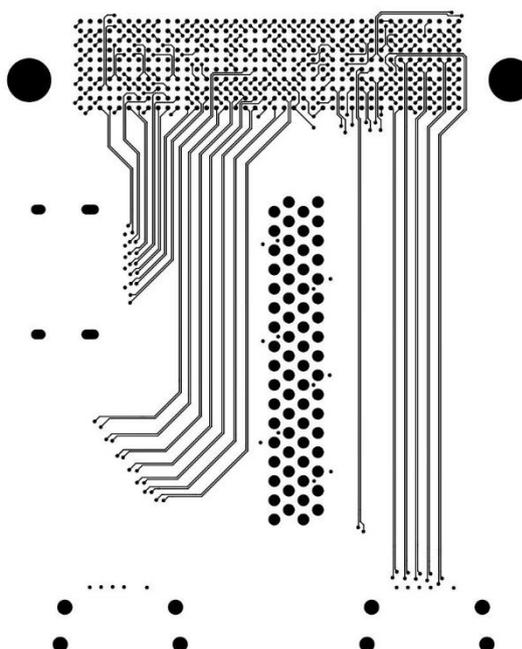


ANEXO V. Vista de las pistas en la placa de pruebas (captura de los archivos Gerber).

Capa de cobre F.Cu:



Capa de cobre B.Cu:



ANEXO VI. Orden de compra para la placa de pruebas.

	After data check
Position Nr.1	
Type	PCB
PCB name	FMC connections
Technology	Standard
Your P.O. number	-
Delivery time	7 working days
Place of manufacture	Germany
Number of layers	4
Order size	4 pieces
Shape:	Other shapes
PCB/ Panel size	69,00 * 88,00 mm
Drill diameter	> 0.10 mm
Smallest routing tool	>= 1.60 mm
Material / Thickness	FR4 / 1.55 mm
Material type	TG 135
Structures outside	> 100 µm
Structures inside	> 100 µm
Chamfer	no
Edge plating	no
Plugged via	no
Outer copper	35 µm
Inner copper	35 µm
Surface finish	Nickel/Gold (Ni/Au)
Additional surface	none
Solder resist	Top + Bottom
Silk screen	Top + Bottom
Silk screen color	white
Additional documentation	none
E-test	yes
UL Marking	No, no UL marking required
Scheduled shipping date :	26.07.2017
Value of goods	211,68 EUR
Price per unit	52,92 EUR
Payment method	per invoice
Shipping method	DHL Standard

XIX. Lista de Acrónimos y Siglas

ATLAS: A Toroidal LHC Apparatus / Aparato Toroidal del LHC.

BE: Back end / terminal trasero.

BRAM: Block RAM / Bloque RAM.

CERN: European Organization for Nuclear Research/Organización Europea para la Investigación Nuclear

CLB: Configurable Logic Block / Bloque lógico configurable.

CLIC: Compact Linear Collider / Colisionador Compacto Lineal.

CPM: Compact Processing Modules / Módulos de procesamiento compacto

DAQ: Data acquisition / Adquisición de datos.

DCS: Detector Control System / Sistema de control de detector

DMA: Direct Memory Acces / Acceso directo a memoria.

EB: Extended Barrel / Barril extendido

EMC: electromagnetic compatibility / compatibilidad electromagnética.

FE: Front end / terminal frontal.

FELIX: Front- End Link eXchange / Enlace de intercambio en el límite frontal.

FIFO: First in, first out / Primero en entrar, primero en salir.

FMC: FPGA Mezzanine Card / Tarjeta intermedia FPGA

FPGA: Field Programmable Gate Array / Arreglo de puertas programable en campo

FE: front end / límite frontal.

GBT: GigaBit Transceiver / Transceptor Gigabit

GUI: Graphical User Interface / Interfaz Gráfica de usuario

HDL: Hardware description language / lenguaje de descripción de hardware

HDLC: High-level Data Link Control / Control de enlace de datos de alto nivel

HL: High Luminosity / Alta Luminocidad

HG: High-gain/ Alta ganancia

HV: High Voltage / Alto voltaje

IFIC: Instituto de Fisica Corpuscular

ILA: Integrated Logic Analyzer / Analizador lógico integrado

IP: Intellectual Property / Propiedad intelectual

ISE: Integrated Synthesis Environment / Entorno de síntesis integrado

I/O: Input/Output / Entrada/salida.

LB: Lumi-blocks / Lumi bloques.

LHC: Large Hadron Collider / Gran Colisionador de Hadrones

LG: Low-Gain / Baja ganancia

LVPS: Low Voltage Power Source / Fuente de potencia de bajo voltaje

MB: Minimum Bias System / Sistema del sesgo mínimo

MCMM: Mixed-Mode Clock Manager / Administrador de reloj en modo mixto

MOIE: Máster Oficial en Ingeniería Electrónica

MRCC: Multi-region Clock Capable / Capaz de un reloj de múltiples regiones

PCB: Printed Circuit Board / Tarjeta de circuito impreso

PLL: Phase-locked loop / bucle de bloqueo de fase

PMT: Photomultiplier tubes / Tubos fotomultiplicadores.

PPR: Preprocessor / Preprocesador

QSFP: Quad SFP / SFP cuádruple.

ROD: Read Out Driver / *Driver* de lectura

ROB: Read-out Buffers / *Buffers* de lectura

RTL: Register-transfer level / Nivel de transferencia de registro.

SFP: Small Form-factor Pluggable / Factor de forma conectable pequeño

SMA: SubMiniature version A / Subminiatura versión A

SRCC: Single Region Clock Capable / Capaz de un reloj con una sola región

SCSI: Small Computer System Interface / Small Computer System Interface

TileCal: Hadronic Tile Calorimeter / Calorímetro de Baldosas Hadrónico.

TTC: time and trigger controler / Controlador de tiempo y disparo

VHDL: VHSIC Hardware Description Language / Lenguaje de descripción hardware de un VHSIC

VHSIC: Very High Speed Integrated Circuit / Circuito integrado de alta velocidad

