

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

ESTUDIO, DISEÑO E IMPLEMENTACIÓN DEL MÓDULO
DE PREPROCESADO DE DATOS DEL SISTEMA READ OUT
DRIVER PARA EL CALORÍMETRO TILECAL DEL
EXPERIMENTO ATLAS/LHC DEL CERN

JOSÉ TORRES PAÍS

UNIVERSITAT DE VALENCIA
Servei de Publicacions
2005

Aquesta Tesi Doctoral va ser presentada a València el dia 22 de Juny de 2005 davant un tribunal format per:

- D. Enrique Sanchís Peris
- D. Gabino Almonacid Puche
- D. Joan Cabestany Moncusí
- D. Antonio Ferrer Soria
- D. Jesús Soret Medel

Va ser dirigida per:

D. Vicente González Millán

©Copyright: Servei de Publicacions
José Torres País

Depòsit legal:

I.S.B.N.:84-370-6360-4

Edita: Universitat de València
Servei de Publicacions
C/ Artes Gráficas, 13 bajo
46010 València
Spain
Telèfon: 963864115

Estudio, Diseño e Implementación
del Módulo de Preprocesado de
Datos del Sistema Read Out Driver
para el Calorímetro TileCal del
Experimento ATLAS/LHC del CERN

José Torres País

Tesis Doctoral
Departamento Ingeniería Electrónica
Universidad de Valencia, Marzo 2005

A Kony
A mi Familia

AGRADECIMIENTOS

Esta Tesis Doctoral ha sido desarrollada casi en su totalidad en el Laboratorio Europeo de Física de Partículas (CERN), ubicado en Ginebra (Suiza). Desde el punto de vista técnico y personal ha sido una experiencia muy gratificante, en la cual han colaborado muchas personas. Me gustaría expresar mi gratitud a todos ellos.

Ante todo quisiera agradecer al Dr. Enrique Sanchis, director del Grupo de Investigación de Diseño de Sistemas Digitales y Comunicaciones (DSDC) de la Universidad de Valencia, la oportunidad que me ha ofrecido para la realización esta Tesis y todo el esfuerzo que ha realizado para la concepción y buen desarrollo de este proyecto. También gracias por todos los consejos recibidos.

Al Dr. Vicente González, director de esta Tesis, por su inestimable apoyo, ayuda y colaboración en todo lo referente a la integración y desarrollo del prototipo en el proyecto ATLAS/LHC. Gracias por las horas de desesperación compartidas en el laboratorio donde parecía que nada funcionaba.

A Jose Castelo, Ximo Poveda y demás gente del IFIC por toda la ayuda prestada en el diseño del software para el prototipo, la integración del mismo en el sistema de adquisición del CERN. Especialmente por las pruebas del último TestBeam donde se sacrificaron muchas horas para llegar a tiempo.

Gracias al Dr. Stefan Haas, del ATLAS Electronics Design Group del CERN, por toda la ayuda prestada con la integración del interface ODIN en las FPGAs. Así como por su inestimable apoyo en el estudio de integridad de la señal en la tarjeta. Sus continuos mensajes y su apoyo fueron fundamentales.

No quiero olvidarme de mis compañeros de cafés y momentos de tertulia de la Universidad. Gracias a Jesús Soret, Gloria Torralba, Juan Ramón Alarcón, Julio Martos, Jose Antonio Gómez e Inma Tortajada por saber escuchar. Y en especial a Tomás García, esto ha sido como un niño para nosotros.

Agradecer, por supuesto, el continuo apoyo y cariño durante tantos años de mis Padres y de mi Hermana. Sólo ellos saben todo lo que hemos sufrido para llegar aquí.

Muchas gracias a Kony, por el ánimo, la compañía y el amor dado en todo momento. Sin ella esta Tesis no tendría sentido.

Y como no, para todos aquellos que me gustaría que estuvieran compartiendo este momento.

ÍNDICE

Prólogo	ix
<u>CAPÍTULO 1 EXPERIMENTO ATLAS/LHC Y EL SUBDETECTOR TILECAL DEL CERN</u>	
1. INTRODUCCIÓN	1-1
2. EXPERIMENTO LHC	1-1
2.1 Importancia del LHC	1-3
2.2 Descripción del acelerador	1-5
3. DETECTOR DE PARTÍCULAS ATLAS	1-6
3.1 Estructura del detector de partículas	1-7
3.2 Funcionamiento de ATLAS	1-8
4. TRIGGER Y ADQUISICIÓN DE DATOS	1-10
4.1 El primer nivel de trigger	1-10
4.2 El segundo nivel de trigger	1-11
4.3 El tercer nivel de trigger	1-12
5. CALORÍMETRO HADRÓNICO (TILECAL)	1-15
5.1 Descripción del calorímetro	1-15
5.2 Aspectos mecánicos, ópticos y electrónicos	1-17
6. BIBLIOGRAFÍA	1-22
<u>CAPÍTULO 2 PROTOTIPO DEL SISTEMA ROD Y DESARROLLO DEL MÓDULO DE TRANSICIÓN TM4PLUS1</u>	
1. INTRODUCCIÓN	2-1
2. PROTOTIPO DEL SISTEMA READ OUT DRIVER (ROD)	2-1
2.1 Descripción del sistema ROD	2-1
2.2 ROD Motherboard	2-4
2.3 DSP Processing Units	2-6
2.4 Algoritmo de Filtrado Óptimo	2-7
3. MÓDULO DE TRANSICIÓN (TM4PLUS1)	2-10
3.1 Descripción del Módulo de Transición	2-10
3.2 Interface S-LINK	2-12

3.3 Dispositivos Lógicos Programables	2-14
3.4 Metodología Diseño FPGAs	2-18
3.5 Reformatting Altera FPGA	2-19
3.5.1 Bloque FIFO	2-19
3.5.2 Bloque connect FIFO	2-20
3.5.3 Bloque control S-LINK	2-23
3.5.4 Bloque data multiplexing, control & filtering unit	2-23
3.5.5 Bloque auxiliary FPGA	2-26
3.5.6 Bloque reformatting FPGA control & test	2-27
3.4 Auxiliary Altera FPGA	2-27
3.4.1 Bloque S-LINK	2-27
3.4.2 Bloque reformatting FPGA	2-28
3.4.3 Bloque J2B connection	2-28
3.4.4 Bloque data control unit	2-28
3.4.5 Bloque auxiliary FPGA clock, control & test	2-28
3.4.6 Bloque ODIN LSC (G-LINK)	2-28
5. FORMATO DE DATOS DEL ROD	2-30
5.1 Estructura de los Datos de Salida	2-30
6. BIBLIOGRAFÍA	2-33

CAPÍTULO 3 SISTEMA ROD FINAL Y DESARROLLO DE LA OPTICAL MULTIPLEXER BOARD

1. INTRODUCCIÓN	3-1
2. SISTEMA READ OUT DRIVER (ROD) FINAL	3-1
2.1 Descripción de la ROD Motherboard	3-1
2.2 Descripción de la Staging FPGA	3-4
3. OPTICAL MULTIPLEXER BOARD (OMB)	3-6
3.1 Estudios de Radiación en el Front End de TileCal	3-6
3.2 Descripción de la Optical Multiplexer Board	3-8
3.2.1 Descripción del Conector de Fibra Óptica de Entrada/Salida	3-11
3.2.2 Descripción del Deserializador	3-11
3.2.3 Descripción de la FPGA de CRC	3-14
3.2.4 Descripción de la FPGA de VME	3-15
3.2.5 Descripción del Serializador	3-17
3.2.6 Generación y distribución de los relojes en la OMB	3-19
3.2.7 Distribución de los datos en la OMB	3-21
3.2.8 Distribución de la alimentación en la OMB	3-22
3.2.9 Conectores J1/P1 y J2/P2	3-22
3.2.10 Señales del Panel Frontal	3-23
3.3 Desarrollo y Costes de la OMB	3-24

3.3.1 Realización de los esquemáticos	3-24
3.3.2 Diseño del circuito impreso	3-27
3.3.3 Rutado del circuito impreso	3-29
3.3.4 Costes de la OMB	3-32
3.4 La OMB como Multiplexora de Datos	3-32
3.4.1 Bloque Reception Fiber A-B	3-33
3.4.2 Bloque CRC Check, Data Multiplexing & Control Unit	3-34
3.4.3 Bloque Transmission	3-36
3.4.4 Bloque VME FPGA	3-37
3.4.5 Bloque CRC Channel 2 FPGA	3-37
3.4.6 Bloque CRC FPGA Clock and Test	3-37
3.4.7 Bloque Connector Test	3-37
3.5 La OMB como Inyectora de Datos	3-38
3.5.1 Bloque External Trigger & Busy	3-38
3.5.2 Bloque Clock Divider	3-38
3.5.3 Bloque Event Generator	3-39
3.5.4 Bloque Transmission	3-39
3.5.5 Bloque Injector Test	3-39
3.5.6 Bloque Connector Test	3-39
3.6 Protocolo VME en la OMB	3-40
3.6.1 Bloque Prerod_Genadd	3-40
3.6.2 Bloque CR_CSR	3-41
3.6.3 Bloque VME_Registers	3-41
4. ESTUDIOS SOBRE INTEGRIDAD DE LA SEÑAL	3-42
4.1 Fundamentos del Diseño Digital para Alta Velocidad	3-42
4.1.1 Revisión de los Conceptos de Frecuencia, Tiempo y Distancia	3-43
4.1.2 Líneas en PCBs	3-47
4.1.2.1 Líneas MicroStrip	3-47
4.1.2.2 Líneas Stripline	3-51
4.1.3 Terminaciones	3-52
4.1.3.1 Formas de Onda en una Línea no Terminada	3-53
4.1.3.2 Terminación en paralelo	3-54
4.1.3.3 Terminación en serie	3-56
4.2 Análisis Pre-Layout de la OMB	3-57
4.2.1 Líneas Diferenciales	3-58
4.2.2 Bus de Datos y Señales de Control	3-62
4.2.3 Relojes	3-66
4.2.4 Crosstalk	3-69
5. BIBLIOGRAFÍA	3-72

CAPÍTULO 4 VERIFICACIÓN EXPERIMENTAL Y RESULTADOS

1. INTRODUCCIÓN	4-1
2. DESCRIPCIÓN DE LAS PRUEBAS DEL SISTEMA ROD INICIAL Y RESULTADOS	4-1
2.1 Descripción del Test Beam de TileCal	4-1
2.1.1 Desarrollo Experimental	4-3
2.1.2 Medidas en el Haz del H8	4-3
2.1.3 Calibración y Desarrollo de los Módulos	4-4
2.2 Instalación del Sistema ROD Inicial en el Test Beam	4-5
2.3 Resultados del Sistema ROD Inicial en el Test Beam	4-7
3. DESCRIPCIÓN DE LAS PRUEBAS DEL SISTEMA ROD FINAL Y RESULTADOS	4-10
3.1 Descripción de los Tests Previos de la OMB	4-10
3.2 Instalación del Sistema ROD Final en el Banco de Test	4-13
3.2.1 Aplicaciones XTestROD y XFilar	4-14
3.3 Resultados del Sistema ROD Final en el Banco de Test	4-16
4. BIBLIOGRAFÍA	4-19

CAPÍTULO 5 CONCLUSIONES Y TRABAJO FUTURO

1. CONCLUSIONES	5-1
2. TRABAJO FUTURO	5-2
2.1 Tests Combinados del Sistema ROD Final	5-3
2.2 Tarjeta Multiplexora 9U	5-4

PRÓLOGO

El objeto de esta Tesis es el diseño, programación e integración del Módulo de Preprocesado de Datos del Sistema Read Out Driver (ROD) dentro de los trabajos que se están realizando en el Laboratorio Europeo de Física de Partículas (CERN). Se trata de un paso definitivo para la puesta en marcha de este sistema que se instalará en el detector ATLAS dentro del acelerador de partículas (LHC) que se está terminando de construir en el CERN.

Este Sistema ROD forma parte del sistema de adquisición de datos del subdetector TileCal incluido en ATLAS, en concreto del sistema de lectura y envío de datos a la frecuencia de cruce de haz que se produce en el acelerador (40 MHz). Incluido entre los niveles uno y dos de la arquitectura jerárquica de detección y estudio de eventos que se producen en las colisiones del LHC.

El trabajo se puede dividir en dos partes, una primera en la que se investigan y exponen las necesidades en materia de adquisición de datos del experimento ATLAS/LHC y una segunda parte centrada en la realización del Sistema ROD, desde su primer prototipo, su diseño hardware, el sistema final, los módulos de preprocesado que se han desarrollado, el software desarrollado en los dispositivos lógicos programables y por último las pruebas, integración y resultados obtenidos en el CERN.

La exposición de esta Tesis Doctoral se ha llevado a cabo con la siguiente estructura. En el primer capítulo se introduce el acelerador LHC para física de altas energías que se está construyendo en el CERN, así como el detector ATLAS incluido en el anillo del acelerador. Del experimento ATLAS se explica la arquitectura de adquisición de datos, y se estudia en función de la frecuencia de los sucesos y del tráfico generado la necesidad de incluir el prototipo ROD. También se estudia el subdetector TileCal incluido en ATLAS que es donde se colocarán los módulos del Sistema ROD dentro de la estructura de adquisición de datos. Se realiza una descripción del subdetector haciendo hincapié en los aspectos electrónicos y en la estructura final de detección de eventos.

En el capítulo 2 se trata el Sistema ROD Inicial, primer prototipo que sirve de base de estudio para su versión definitiva, se analiza su necesidad, sus características principales y la motherboard que se usa como elemento de soporte a la conexión VME con ATLAS y a las Unidades de Proceso. Así mismo, se define uno de los algoritmos de reconstrucción de la energía depositada en las celdas del calorímetro (Optimal Filtering). Así mismo, este capítulo se centra en el desarrollo del primer módulo de preprocesado de datos, el Módulo de Transición TM4Plus1. Inicialmente se describen los componentes que forman el módulo y la forma del flujo de datos que entran y salen del mismo. Del enlace con el subdetector, se analiza el Interface S-LINK usado

en la comunicación y su inclusión en el módulo. Los dos elementos principales del módulo son dos FPGAs (APEX 20K de Altera), de las cuales se realiza una descripción de su arquitectura interna y de la metodología usada para el desarrollo de las mismas.

Siguiendo con el mismo capítulo, se definen los diagramas de bloques de las dos FPGAs y se estudia en detalle cada uno de los elementos que lo constituyen. Se analizan los diferentes modos de operación y los algoritmos que se implementan. Como paso final se detalla el formato de datos de salida del ROD, datos que deben ser enviados al siguiente nivel de adquisición de datos.

El tercer capítulo de esta Tesis Doctoral nos muestra la evolución seguida por el Sistema ROD, y que conduce al Sistema ROD Final. Se analizan las partes más importantes del mismo y se expone en profundidad la Optical Multiplexer Board. Es en esta tarjeta donde se centra esta Tesis, desde sus inicios y su necesidad debida a la alta radiación que se producirá en TileCal hasta su desarrollo final, pasando por los estudios que de la misma se realizan.

Una novedad importante de este módulo es su realización completa dentro de los laboratorios de la Universidad de Valencia. Hasta estos momentos, este tipo de tarjetas se desarrollaban casi íntegramente en el CERN. Por primera vez, todo el desarrollo ha corrido a cargo nuestro. Se analizó con detalle todo el módulo y se realizaron estudios sobre integridad de la señal que nos permitieran realizar un trabajo correcto y estable.

En el capítulo 4, se describe el sistema Test Beam instalado en el CERN para poder realizar un análisis previo de todas las estructuras necesarias en ATLAS. Para ello se usa un pequeño acelerador con un haz suficiente para todos los trabajos previos antes de tener los sistemas finales que se instalarán en el detector. También se describe la integración y las pruebas realizadas con el Módulo de Transición y el Sistema ROD en el subdetector y los resultados obtenidos que verifican los teóricos expuestos anteriormente y demuestran la utilidad del módulo en el sistema de adquisición.

Así mismo, se describe el Banco de Test que se tuvo que montar en el CERN y en los laboratorios de la Universidad de Valencia para poder realizar un test correcto de los nuevos módulos del Sistema ROD. Es un test importante y definitivo que nos permitirá realizar una producción definitiva de todos los módulos necesarios para este proyecto.

En el último capítulo y en vista de los resultados obtenidos, se analiza en forma de conclusiones todo el trabajo desarrollado y se proponen una serie de trabajos futuros destinados a mejorar el diseño y adaptarlo para su utilización definitiva en el subdetector, así como a la realización de una tarjeta Optical Multiplexer Board de mayores dimensiones que nos permita recibir un número mayor de canales de datos en menor espacio. Esto nos permitiría reducir el número de RODs y presentar un preprocesado de datos para obtener una adquisición lo más fiable posible.

CAPÍTULO

1

EXPERIMENTO ATLAS/LHC Y EL SUBDETECTOR TILECAL DEL CERN

1. INTRODUCCIÓN	1
2. EXPERIMENTO LHC	1
2.1 Importancia del LHC	3
2.2 Descripción del acelerador	5
3. DETECTOR DE PARTÍCULAS ATLAS	6
3.1 Estructura del detector de partículas	7
3.2 Funcionamiento de ATLAS	8
4. TRIGGER Y ADQUISICIÓN DE DATOS	10
4.1 El primer nivel de trigger	10
4.2 El segundo nivel de trigger	11
4.3 El tercer nivel de trigger	12
5. CALORÍMETRO HADRÓNICO (TILECAL)	15
5.1 Descripción del calorímetro	15
5.2 Aspectos mecánicos, ópticos y electrónicos	17
6. BIBLIOGRAFÍA	22

1. INTRODUCCIÓN

El *Large Hadron Collider* (LHC) es el próximo instrumento de investigación en Física de Partículas que permitirá impulsar los avances realizados en los últimos 100 años en la comprensión de la estructura de la materia. El LHC es un acelerador de partículas que se está construyendo en el Laboratorio Europeo de Física de Partículas (CERN) en Ginebra.

En estos años nos enfrentamos a preguntas como, ¿qué es la masa?, ¿cuál es su origen?, ¿por qué hay en nuestro Universo más materia que antimateria? ¿son las aparentemente distintas fuerzas de la Naturaleza aspectos de una sola fuerza fundamental? El LHC está diseñado para buscar respuestas a estas preguntas y convertirse en una herramienta indispensable para poner sólidos cimientos científicos a este siglo.

Para el estudio de todas estas cuestiones que nacerán en el LHC se necesitan grandes detectores de partículas. Uno de ellos es el A Toroidal Lhc ApparatuS (ATLAS) y es en él donde centraremos esta Tesis Doctoral, concretamente en la adquisición de datos que se producen en una parte del detector.

A su vez, trataremos desde un punto de vista general, los distintos subdetectores que constituyen el detector ATLAS y sus principales características. A continuación, se darán a conocer los diferentes aspectos particulares del calorímetro hadrónico de tejas (TILECAL), desde el sistema de funcionamiento, la mecánica necesaria, la electrónica y los sistemas de lectura y adquisición de datos.

Este capítulo presenta una descripción del experimento ATLAS/LHC, de la arquitectura jerárquica para el tratamiento de los eventos producidos en el acelerador y del subdetector TileCal en el cual se ha trabajado.

2. EXPERIMENTO LHC

Actualmente todo el Universo, y las interacciones que en él se producen (fuerte, débil, electromagnética y gravitatoria), puede explicarse mediante dos familias de partículas: los fermiones y los bosones. Los fermiones, que son los constituyentes de la materia, son partículas con momento magnético o espín no entero, mientras que los bosones, responsables de las fuerzas, son aquéllas con momento entero.

Los fermiones se clasifican a su vez en quarks y leptones. Los primeros son partículas con carga eléctrica fraccionaria y que experimentan los cuatro tipos de interacción. Los leptones tienen carga eléctrica entera y no experimentan todas las fuerzas. La tabla 1.1 resume esta clasificación. De cada una de las partículas existe su antipartícula.

Leptones		
e (electrón)	μ (muón)	τ (tau)
ν_e (neutrino del electrón)	ν_μ (neutrino del muón)	ν_τ (neutrino del tau)
Quarks		
u (up)	c (charmed)	t (top)
d (down)	s (strange)	b (bottom)

TABLA 1.1 Clasificación de los fermiones.

Los cuatro tipos de interacción se explican a través de los bosones. El hecho de que una partícula sea responsable de una interacción no es una cosa fácil de entender. Por ello, se acude a menudo a símiles que se aproximan más o menos a la realidad únicamente explicable a través de las ecuaciones correspondientes. Uno de los símiles empleados más frecuentemente nos presenta una situación en la que estamos montados en una barca en un lago, y desde otra nos lanzan una pelota pesada. Al coger la pelota, nuestra barca se moverá. Un observador que no viese la pelota pensaría que entre las dos barcas ha aparecido una fuerza que las ha hecho separarse. En este caso, la pelota hace las veces del bosón, y las barcas juegan el papel de los fermiones que experimentan esa fuerza. El símil es limitado en cuanto que sólo nos ofrece una imagen de una fuerza de repulsión, y es difícil explicar por el mismo mecanismo una fuerza de atracción.

La tabla 1.2 muestra los diferentes bosones asociados a cada una de las fuerzas del Universo y el rango de distancias en el que actúan. La fuerza fuerte actúa entre los quarks, mientras que la débil es responsable de algunas formas de radioactividad. La fuerza gravitatoria es la responsable de la fuerza de atracción entre los cuerpos debido a su masa. Por último, la fuerza electromagnética es la responsable de los efectos de los campos magnéticos y eléctricos. De los bosones, sólo falta por descubrir el responsable de la fuerza gravitatoria.

Desde hace bastante tiempo, los físicos experimentales intentan probar que esas cuatro fuerzas no son sino diferentes manifestaciones de una misma. Actualmente ya se ha descubierto que las fuerzas electromagnética y débil son una misma que se manifiesta de forma diferente en función del rango de energías donde actúa.

Fuerza	Bosón	Rango
Fuerte	Gluón	-
Débil	W^+ , W^- , Z^0	
Gravitatoria	Gravitón (no descubierto aún)	
Electromagnética	γ (fotón)	+

TABLA 1.2 Clasificación de fuerzas y bosones.

Los experimentos de física de altas energías pretenden, en la mayoría de los casos, dar validez a una teoría enunciada con anterioridad, dicha teoría se funde en un Modelo Estándar que recoge todas estas partículas e intenta dar sentido al origen del Universo [1]. En general, la validez depende de la existencia, y con unas determinadas características, de una partícula que no suele presentarse de forma espontánea en la naturaleza.

El porqué de la no existencia lo hemos de buscar en el hecho de que la investigación no pretende otra cosa que profundizar en el conocimiento de la materia. Esta profundización conlleva un estudio hacia atrás en el tiempo para analizar qué es lo que pasó en los primeros instantes del Universo, apenas unos picosegundos después del Big Bang.

Los experimentos que se realizan se basan en la recreación de las condiciones existentes en esos primeros instantes del Universo, es decir, en producir una condensación de energía que dé lugar a la formación de materia. Para ello, se realiza el proceso inverso: se convierte materia en energía, con la ayuda de aceleradores como los instalados en el CERN, el Laboratorio Europeo de Física de Partículas, o en el FermiLab en EEUU.

En ellos, partículas como electrones (e^-), positrones (e^+), protones (p) o incluso núcleos atómicos, se aceleran a velocidades próximas a la de la luz para después hacerlos chocar contra blancos fijos o entre ellos (colisionadores de partículas). Para conseguir un número mayor de interacciones, las partículas circulan en paquetes o *bunches* de una determinada densidad. Al número de partículas por unidad de superficie y tiempo se le conoce como luminosidad.

Los aceleradores más potentes en cuanto a la energía que son capaces de generar son los colisionadores, como el LEP (Large Electron-positron Collider del CERN) [2], que se han mostrado eficaces en el mejor conocimiento de nueva física (producción masiva del Z^0 [3]) pero que se muestran insuficientes para lo que actualmente predice la teoría. Por ello, los proyectos del CERN van ahora encaminados a la construcción de un nuevo colisionador de partículas, el LHC (Large Hadron Collider) [4], con requerimientos muy exigentes en cuanto a frecuencia de colisión y luminosidad como veremos más adelante.

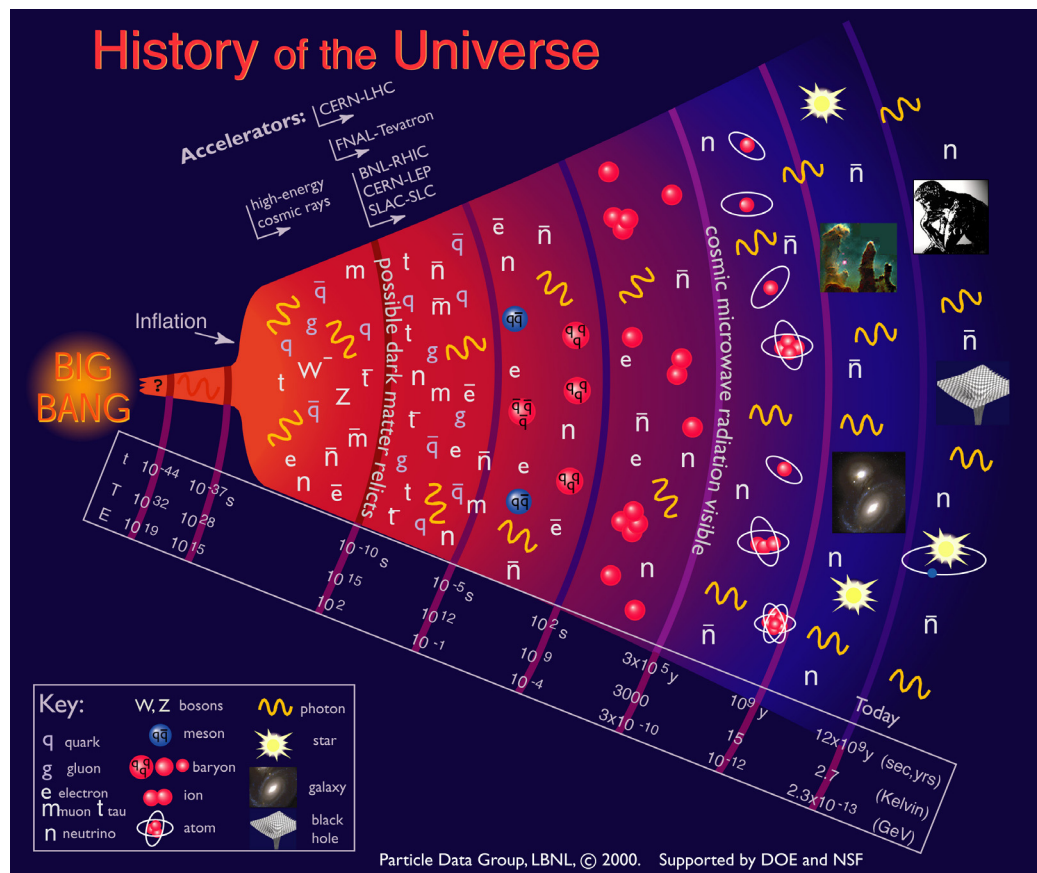


FIGURA 1.1 Historia del Universo.

2.1 IMPORTANCIA DEL LHC

LHC (Large Hadron Collider) es el nombre del colisionador de partículas que actualmente se encuentra en construcción en las instalaciones del CERN. Su estructura es la de un anillo de 27 Km de circunferencia por el que circularán y se harán colisionar protones entre sí con una luminosidad de $1,7 \cdot 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$, cada 25 ns, obteniéndose una energía de 17 TeV [5], casi un orden de magnitud superior a LEP (100-200 GeV). El colisionador se instalará en un túnel excavado a una

profundidad media de 100 m en la frontera entre Francia y Suiza bajo las instalaciones del CERN y que actualmente alberga el colisionador LEP.

La razón de la construcción de este nuevo colisionador, aprobada en diciembre de 1994 por el Consejo del CERN, es el estudio de ciertos fenómenos predichos por la teoría pero no comprobados hasta ahora. Entre ellos está el hecho de la existencia de tres bosones diferentes para la fuerza débil con masas diferentes (ver tabla 1.2); LHC intentará descubrir el motivo de esta diferencia que se atribuye a la existencia, en los primeros instantes del Big Bang, de un bosón de Higgs, el cual confirió a cada uno su masa particular. Puesto que se trata de un bosón, es una partícula responsable de una interacción, pero en este caso de la interacción de unos bosones (las de la fuerza débil) con un determinado campo de Higgs de forma que a mayor interacción mayor masa.

La explicación de cómo habría actuado este bosón la podemos explicar acudiendo también a un símil. Imaginemos una fiesta en la que hay un gran número de personas. De repente, entra en la sala un personaje muy conocido. La gente al verlo se aglutina alrededor de él y le impide el avance. El personaje ve frenada su marcha, es como si le costase más avanzar, como si tuviese "más masa" que mover. A lo largo de su avance por toda la sala nueva gente se situaría alrededor de él y otra se iría alejando cuando ya lo hubiese visto y/o hablado con él. El mismo efecto se produciría si, en vez de entrar una persona, lo que se difundiese por la sala fuera un rumor. La gente se reuniría para escucharlo y la cantidad de gente sería mayor cuanto más interesante y "morboso" fuera el rumor. Pues bien, el bosón de Higgs sería como ese rumor que se difunde por el espacio de la sala y que aglutinaría a más o menos gente; la cantidad de gente representaría la masa.

Los haces de protones colisionarán en cuatro puntos determinados de la circunferencia, donde se instalarán los cuatro detectores previstos: ATLAS (A Toroidal LHC Apparatus) [6], CMS (Compact Muon Solenoid) [7], ALICE (A Large Ion Collider Experiment) [8] y LHC-B (Large Hadron Collider Beauty experiment) [9].

En definitiva, el LHC será un instrumento clave en el estudio de partículas del próximo siglo, concretamente en los siguientes campos:

- Confirmación de la existencia o no del bosón de Higgs. De vital importancia para explicar el mecanismo por el cual las partículas adquieren su masa. La observación del Higgs, que de existir sabemos es una partícula muy pesada, requiere haces con energías muy altas. El LHC cubre las posibles bandas de energía en las que se podría encontrar.
- Partículas Supersimétricas. Los límites inferiores actuales en las masas de estas partículas son también muy elevados. Es necesario irse a grandes energías para probar un territorio nuevo. Si el esquema de supersimetría es válido habría más de una partícula Higgs.
- Estudio de violación CP. Esta asimetría de las Leyes Físicas podrían explicar porqué surgió del Big Bang la materia actual y no la antimateria. El Modelo Estándar predice a qué nivel deberá ocurrir la violación de CP con los mesones B que contienen a los pesados quarks. El LHC será una copiosa fuente de pares de mesones/antimesones B y someterá al Modelo Estándar a pruebas cruciales.
- Plasma de quarks y gluones. Además de colisionar protones, el LHC también podrá hacer colisionar núcleos atómicos pesados (iones). Estos grandes núcleos de muy alta energía chocarán produciendo "burbujas" de materia con densidades y temperaturas muy elevadas. La teoría de la fuerza fuerte predice que, en estas condiciones, las "bolsas de hadrones" de la materia corriente "estallarán" formando un plasma desconfinado de quarks y gluones. Esta nueva forma de materia podría haber existido en los primeros veinte microsegundos después del Big Bang y estar aún presente en los núcleos densos de las estrellas de neutrones.

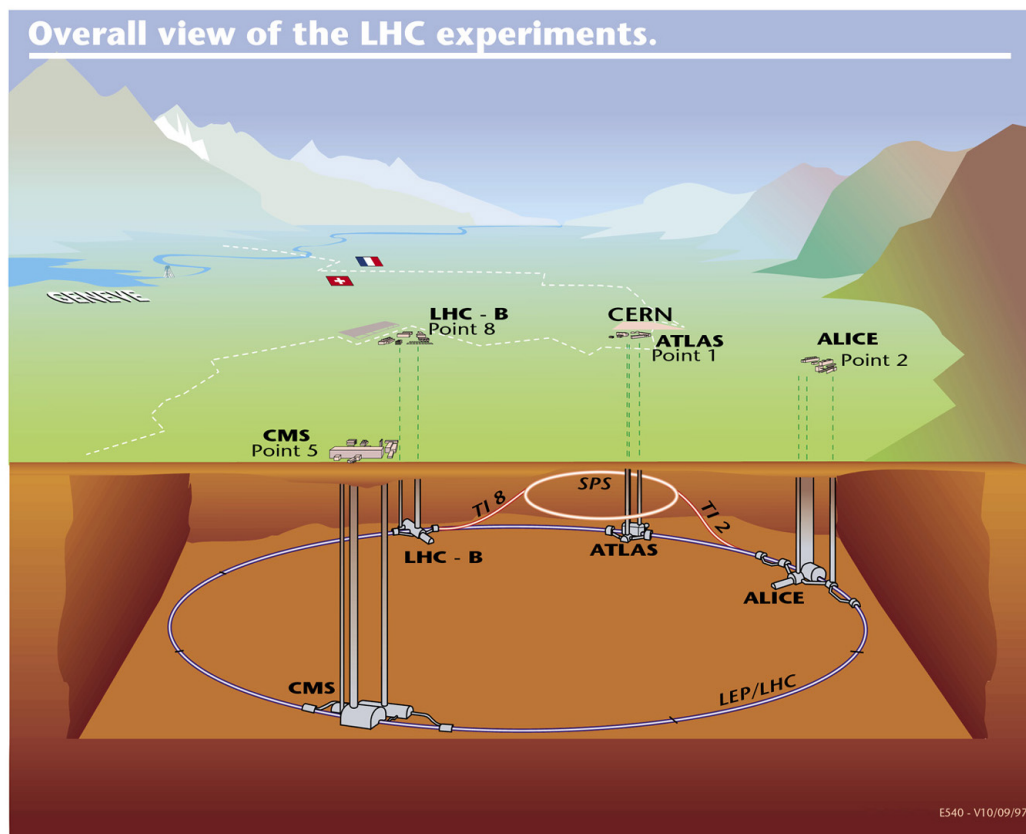


FIGURA 1.2 El experimento LHC.

2.2 DESCRIPCIÓN DEL ACELERADOR

El LHC será el primer acelerador donde colisionarán los quarks y los gluones, constituyentes de los protones. Para maximizar la probabilidad de observación de fenómenos poco comunes, las frecuencias de colisión serán cien veces más altas que los valores máximos actuales.

Al igual que los anteriores aceleradores, requiere una serie de componentes, como son los imanes dipolares para curvar la trayectoria de las partículas, imanes cuadrupolares para focalizar los haces y cavidades de radiofrecuencia para acelerar las partículas. En el LHC estos componentes serán superconductores, operando a la temperatura del helio superfluido (1,9 K).

A diferencia del anterior acelerador (LEP), donde las partículas circulando en direcciones opuestas tienen carga también opuesta (y por tanto, pueden mantenerse en idénticas órbitas con los imanes dipolares) en el LHC ambos haces son de protones. Esto requiere de dos “aceleradores” distintos, o para ser más precisos, de dos guías magnéticas distintas.

En el LHC los imanes dipolares tienen dos canales magnéticos, con los protones circulando en sendos tubos, compartiendo el mismo yugo magnético y criostato. Estos imanes, de los que se requieren 1296, cada uno de 13 metros de longitud, deben producir campos muy altos (8,65 Teslas), para mantener la curvatura de los protones de 7 TeV.

La aceleración la proporcionan las cavidades de radiofrecuencia que en el LHC operarán a 400 MHz. Los haces van en paquetes que se cruzan, en un punto dado,

cada 25 ns como ya se había comentado, es decir a una frecuencia de 40 MHz. El número de paquetes es de 2.835 en cada haz y la separación entre ellos de unos 7,5 m.

El gran número de paquetes junto con la intensidad de cada haz (cada paquete contendrá unos 10^{11} protones), y con la alta focalización en los puntos de cruce, hace que la luminosidad sea un factor diez veces más alta que cualquier otra máquina hecha hasta ahora. La frecuencia de sucesos dando lugar a un estado final determinado, viene dada por la sección eficaz de la reacción correspondiente multiplicada por la luminosidad del acelerador. La alta luminosidad permite por lo tanto obtener muestras estadísticamente muy precisas.

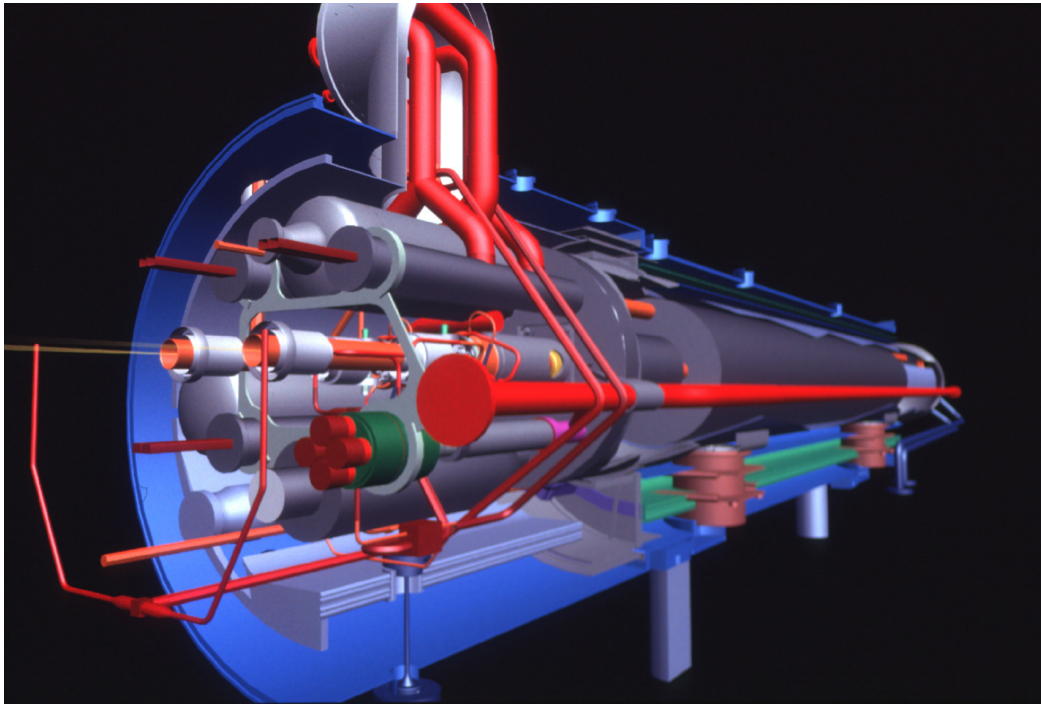


FIGURA 1.3 Interior del acelerador.

3. DETECTOR DE PARTÍCULAS ATLAS

La otra parte esencial del equipo experimental son los detectores, como ya hemos comentado anteriormente en el LHC se utilizarán cuatro detectores; dos para estudiar las interacciones protón-protón (ATLAS y CMS), uno para estudiar colisiones entre iones pesados (ALICE) y un detector especializado en el tema de la violación CP “o física del quark b” (LHCb), siendo el tema prioritario del LHC el estudio que realizarán ATLAS y CMS.

La complejidad de los detectores viene determinada porque a la máxima luminosidad del LHC, en un cruce de haces se producirán varias interacciones; es decir, varios protones de uno de los haces pasarán a una distancia lo suficientemente pequeña de protones del haz contrario como para que se produzcan varias interacciones fuertes y por lo tanto varios sucesos, unos veinte de promedio.

No obstante, los problemas de reconocimiento de trazas son mucho más graves y requieren que los detectores tengan una gran granularidad; es decir, puedan separar espacialmente las señales que dejan las partículas del estado final, aún cuando la densidad de éstas sea grande. La granularidad, unida al tamaño, hace que los

detectores del LHC tengan varios millones de canales electrónicos, en lugar del medio millón típico de los anteriores detectores.

Otra característica muy importante es la frecuencia a la que los haces se cruzan, cada 25 ns, esto provoca un “disparo” en el detector en ese tiempo y una lectura sólo si en ese cruce se produce un suceso de interés. En ese caso, todos los datos de los canales son almacenados para un posterior análisis. Evidentemente esta necesidad de almacenar la información introduce una complicación considerable en la electrónica, así como en toda la arquitectura lógica del sistema de adquisición de datos.

Cada detector se divide en diferentes subdetectores alrededor del punto de colisión en forma de anillos cilíndricos. Los extremos del cilindro están también cubiertos, al máximo posible, de planos de subdetectores, con objeto de evitar que las partículas que interactúan se “escapen” sin ser detectadas.

3.1 ESTRUCTURA DEL DETECTOR DE PARTÍCULAS

El detector ATLAS es un enorme cilindro con simetría dodecaédrica y dimensiones aproximadas de 22 m de longitud y 32 m de altura. Su peso total se estima en 6 KT y su precio alrededor de los 500 millones de francos suizos (unos 300 millones de euros).

Evidentemente, unas características de este calibre exigen que el proyecto se lleve a cabo de forma conjunta. La colaboración ATLAS, creada en 1994, está formada por más de 100 institutos europeos y americanos y más de 1000 personas. La Universitat de València, además de otras universidades y centros de investigación españoles, es una de las integrantes de esta colaboración.

La estructura de ATLAS, figura 1.4, es la de un gran toroide superconductor con núcleo de aire de 26 m de largo, con radios interno y externo de 5 y 10 m respectivamente, capaz de producir un campo magnético interno de 0,8 Teslas. Internamente, se sitúa un solenoide superconductor de 1,2 m de radio y 6,8 m de largo capaz de generar un campo magnético de 2 Teslas en el volumen central del detector.

ATLAS está compuesto básicamente de los siguiente subdetectores:

- Detectores de traza. Se sitúan más cerca del punto de colisión, en la parte más interna del detector. Su misión es determinar la trayectoria de las partículas resultantes del choque.
- Calorímetro electromagnético. Recubre a los detectores de traza y al solenoide interno. Mide la energía de los fotones o electrones resultantes de la colisión. La forma de realizar esta medida es "frenando" hasta parar la partícula y midiendo la energía producida en este proceso.
- Calorímetro hadrónico. Se sitúa alrededor del calorímetro electromagnético. Su misión es semejante a la de éste pero para las partículas pesadas.
- Cámaras de muones. Constituyen la parte más externa del detector. Miden el momento de los muones de mayor energía que no han podido ser frenados por los dos calorímetros.

Como se puede observar, cada subdetector se especializa en la medición de una determinada magnitud para unas determinadas partículas. Sin embargo, existen partículas que no son detectadas por ninguno de los subdetectores (como los neutrinos) y cuya presencia se deduce del desequilibrio momento-energía una vez reconstruido completamente el suceso. Por ello, el detector deberá ser hermético, es decir, sin agujeros a través de los cuales las partículas que interactúan puedan perderse sin ser detectadas, invalidando el cálculo de la energía que falta.

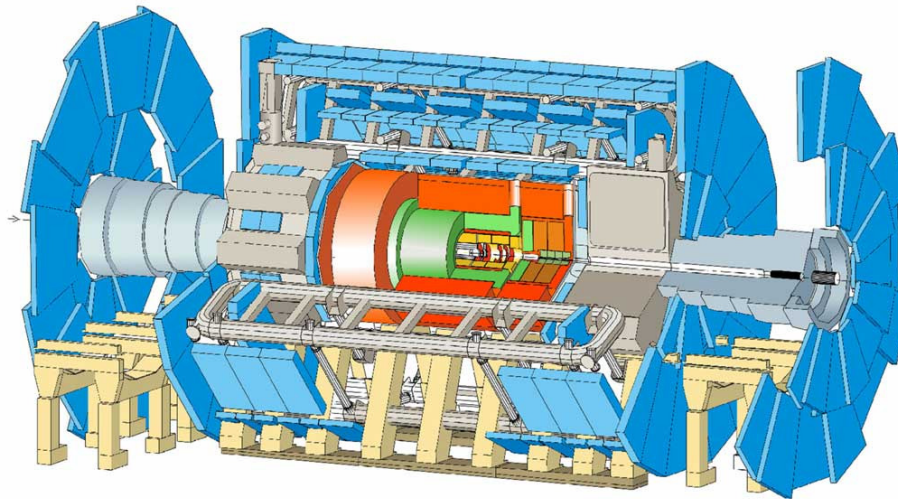


FIGURA 1.4 El detector ATLAS.

3.2 FUNCIONAMIENTO DE ATLAS

El sistema de adquisición del experimento ATLAS/LHC [6] es peculiar en varios aspectos. En primer lugar, se trata de un sistema para una red de sensores distribuidos compuesta de hasta centenares de millones de sensores. Puesto que el número de sensores es elevado, también lo es el número de canales electrónicos que hay que procesar. En principio, este tipo de experimentos necesita que se adquiera la información de cada sensor independientemente para luego ser procesada de manera adecuada. La tabla 1.3 ofrece los datos acerca del número de canales electrónicos a leer en cada una de las partes que componen cada subdetector del experimento.

Subdetector	Canales	Tamaño del suceso (KB)
Calorímetros		180
Electromagnético	224.100	
Hadrónico	10.000	
Trazas		900
Pixels	$140 \cdot 10^6$	
Silicon Strips	$2,9 \cdot 10^6$	
GaAs	$0,8 \cdot 10^6$	
TRT	$0,42 \cdot 10^6$	
Muones		200
MDT	300.000	
CSC	100.000	
RPC	400.000	
TGC	490.000	
TOTAL	147.144.100	1.280

TABLA 1.3 Canales electrónicos en cada subdetector [6].

En nuestro caso, las colisiones se producen cada 25 ns (es decir, con una frecuencia de 40 MHz). Como ya se apuntó, esto va a provocar que la tasa de datos sea excepcionalmente elevada, estimándose en unos 50 Tbytes/s. La capacidad de cálculo total necesaria para tratar el problema se estima en $5 \cdot 10^{10}$ MIPS (como referencia y para darse cuenta de la dimensión del problema, un procesador Pentium IV de última generación posee una capacidad de procesado de unos 10.000 MIPS, lo que implicaría que sería necesario un sistema con ¡5 millones de procesadores! para poder tratar el problema).

Como ya se ha visto, la implementación de un sistema paralelo resulta inabordable debido al gran número de procesadores necesarios. Por ello, tradicionalmente, en este tipo de experimentos se han empleado arquitecturas jerárquicas para la adquisición de datos por una razón básica: la sección eficaz de producción de las partículas buscadas es muy baja, del orden de 10^{-12} - 10^{-13} . Esto implica que la inmensa mayoría de los datos leídos no corresponderán con el suceso a estudiar y, por lo tanto, podrán ser desechados. En este caso, la solución jerárquica ofrece ventajas ya que el análisis total del suceso sólo se hará cuando ocurra uno de ellos y no siempre, como sucedería en un sistema paralelo.

Para el experimento ATLAS/LHC se ha propuesto un sistema de adquisición basado en tres niveles como se muestra en la figura 1.5. A cada uno de estos niveles se le conoce como *nivel de trigger*.

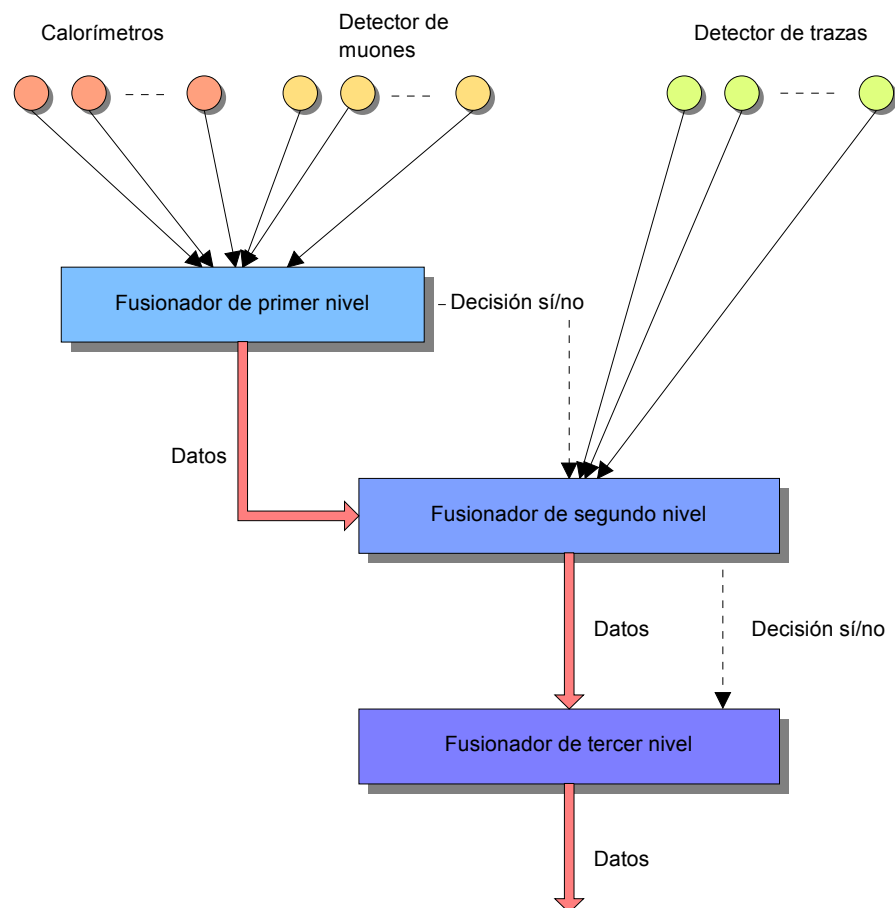


FIGURA 1.5 La arquitectura de selección de tres niveles de ATLAS/LHC.

4. TRIGGER Y ADQUISICIÓN DE DATOS

Los estudios realizados sobre el volumen de datos y las frecuencias de trigger que se van a obtener en el LHC presentan un desafío sin precedentes en materia de adquisición de datos. Como ya hemos comentado, la frecuencia de cruces de haz en ATLAS está estimada en 40 MHz, pudiendo llegar a producirse hasta 20 sucesos por cruce, lo que supone la lectura de miles de canales cada 25 ns.

En esta Tesis Doctoral se ha diseñado, estudiado y probado un componente fundamental dentro de la arquitectura jerárquica de la adquisición de datos en ATLAS. Antes de pasar a definir este componente, es necesario comprender de una forma precisa la estructura global del trigger y la adquisición de datos del detector ATLAS.

Las unidades globales de funcionamiento que se presentan a continuación fueron definidas por la colaboración ATLAS en abril de 1993. La arquitectura está definida en tres niveles jerárquicos diferentes, situados entre el corazón de los detectores donde se producen los cruces de haces y las estaciones de trabajo donde los físicos pueden estudiar los sucesos seleccionados.

Por tanto, el sistema de selección que permite la adquisición de un suceso o su veto se le denomina trigger. El sistema de trigger en ATLAS permite realizar una selección de sucesos válidos en tiempo real, y por lo tanto, reducir el volumen de información a almacenar. Esta organizado en tres niveles (LVL1, LVL2, LVL3).

4.1 EL PRIMER NIVEL DE TRIGGER

El primer nivel de trigger [10] aceptará los datos a la frecuencia de colisión del LHC (40 MHz) y realizará una decisión "sí/no" cada 2 μ s sobre un volumen de datos de unos 380 Kbytes correspondientes sólo al subdetector de muones y a los calorímetros (ver tabla 1.3) ya que el subdetector de traza, debido al gran número de canales, no interviene en esta decisión. Este volumen de datos implica unos requerimientos de ancho de banda de unos 14 Tbytes/s. Este nivel realiza básicamente operaciones sencillas tales como comparación con umbrales para desechar los canales no válidos, y hace uso intensivo de arquitecturas paralelas y encauzadas para poder realizar su tarea en el tiempo indicado. Las previsiones físicas indican que, con el procesado esperado en el primer nivel, se debe poder aceptar en media 1 de cada 400 sucesos. De esta manera, la frecuencia media de salida de datos hacia el segundo nivel será de 100 KHz, siendo la tasa de datos esperada de unos 100 GBytes/s. Se estima que por canal será necesario realizar un máximo de 80 operaciones durante los 2 μ s que dura la decisión. El número de canales es de aproximadamente 1.124.000 por lo que la capacidad de procesado necesaria en este nivel se estima en unos $45 \cdot 10^6$ MIPS.

La figura 1.6 muestra el diagrama de bloques del nivel 1. En este nivel sólo se realiza el procesado de los datos procedentes de los calorímetros y de la cámara de muones. Los procesadores de subtrigger se encargan de procesar la información de cada subdetector y trabajan en paralelo e independientemente en cada uno de ellos para obtener la información deseada. El procesador central de trigger (CTP) es el encargado de realizar la decisión "sí/no" del primer nivel a partir de los datos que los procesadores de subtrigger le pasarán. La decisión se distribuye a los buffers del front-end (la electrónica de lectura de los sensores) para que envíen los datos hacia el segundo nivel.

El procesador de nivel 1 se situará lo más cerca posible del detector para minimizar las latencias debidas a retrasos de propagación en el medio de transmisión. Parte del mismo se instalará en el mismo detector, mientras que el procesador central de trigger y el procesador de subtrigger del calorímetro lo harán en un recinto a unos 6 metros del detector.

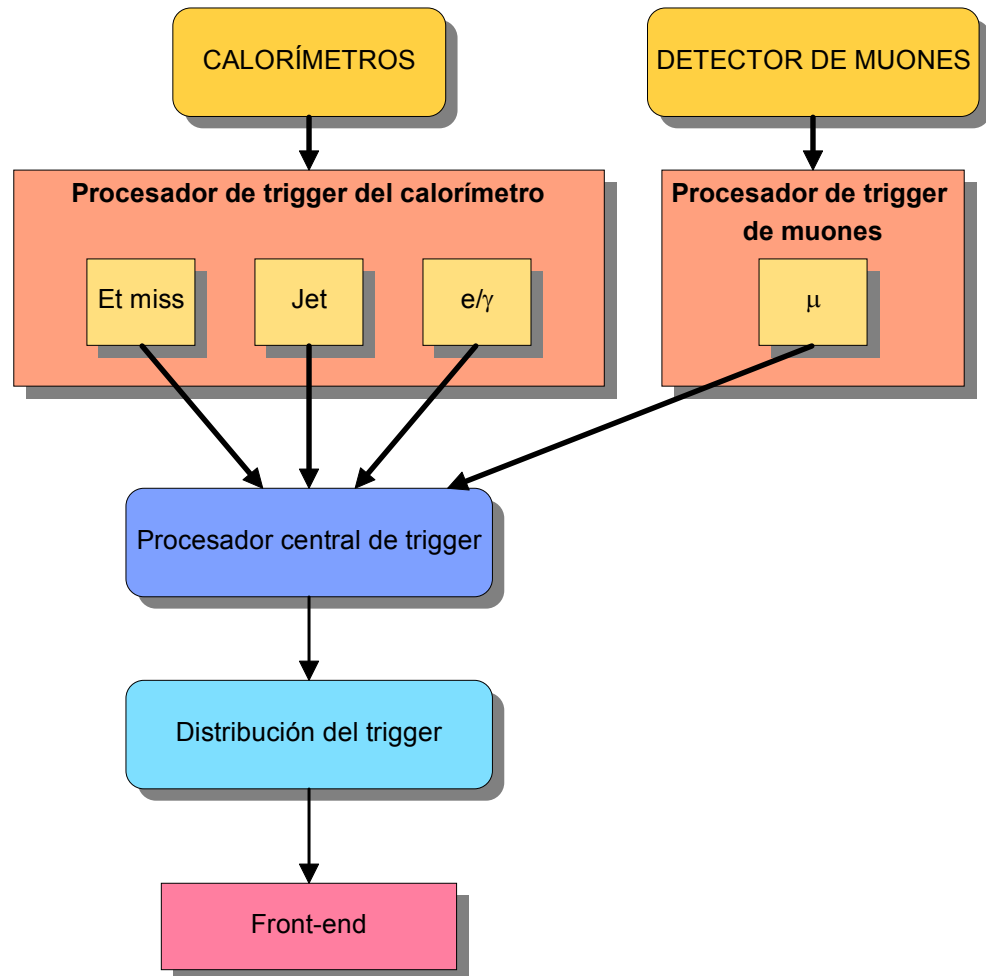


FIGURA 1.6 Diagrama de bloques del primer nivel de trigger.

4.2 EL SEGUNDO NIVEL DE TRIGGER

El segundo nivel trabaja ya con la información de todos los subdetectores y realiza un procesado más detallado sobre los mismos [11]. Este procesado implica la necesidad de correlacionar datos de diferentes subdetectores.

Si la decisión del primer nivel es afirmativa, el conjunto de datos del detector se envía hacia el segundo nivel. Cuando se realiza una decisión “no”, los datos se eliminan de las memorias de primer nivel. Puede ocurrir, sin embargo, que en dos colisiones sucesivas se produzcan triggers de interés, lo que produciría un pico en la frecuencia de salida de datos. Para absorber estos picos y mantener una frecuencia de salida de datos constante hacia el segundo nivel, se sitúan a la salida del primer nivel una serie de buffers desaleatorizadores. Estos buffers deben ser rápidos para poder

absorber los picos de datos y, por lo tanto, caros. Por ello se admite una pérdida de datos de un 1% por buffer lleno para lograr un compromiso entre tamaño y precio.

Los datos de los buffers se envían al segundo nivel a través de fibra óptica a 1 Gbit/s. Los datos son almacenados en las memorias digitales del segundo nivel (*ReadOut Buffers* o *ROB*) donde permanecen durante el tiempo que dura la decisión (10 ms).

La tasa de entrada de datos es de 100 Gbytes/s y la capacidad de cálculo necesaria se estima en unos $32 \cdot 10^6$ MIPS [12]. Si la decisión es positiva, los datos almacenados en los ROBs, se transfieren al tercer nivel. El factor de rechazo esperado en este nivel es del 99%, de modo que la frecuencia de salida de datos será 1KHz, dando una tasa de datos de salida entre 1 y 10 GBytes/s.

4.3 EL TERCER NIVEL DE TRIGGER

El tercer nivel trabaja con la información completa de todo el detector, si la decisión del segundo nivel es afirmativa. Las decisiones se realizan cada 10 ms y se debe conseguir una tasa de salida de 10 a 100 MBytes/s a partir de una de entrada de 1 a 10 Gbytes/s. Los datos que pasan este tercer nivel, uno de cada diez según las simulaciones de física, se almacenan de manera permanente para su análisis posterior.

La estructura general de este nivel puede observarse en la figura 1.7. Los datos del segundo nivel serán rutados a través del denominado *event builder*, una red de conmutación de alta velocidad, hacia los procesadores correspondientes del tercer nivel. Estos procesadores ejecutarán algoritmos complejos como correlaciones entre los valores de los diferentes canales o filtrado de los datos a través de algoritmos más sofisticados con el fin de identificar las partículas a partir de las informaciones de todos los subdetectores y poder seleccionar los sucesos de acuerdo con su significado físico. La tasa total de datos a almacenar será reducida mediante una combinación de compresión de datos y selección de sucesos.

El control de los datos será supervisado por un manejador de flujo de datos (*Data Flow Manager* o *DFM*). Este dispositivo recibirá información de las unidades de procesado del tercer nivel en términos de su estado (libres u ocupadas) y será el responsable de la asignación de destinos para los sucesos a reconstruir y de mantener la carga tanto en la red como en los procesadores.

Las simulaciones realizadas para obtener la potencia de cálculo de este nivel muestran que ésta debe situarse sobre los 1000 MIPS x s/suceso [12]. Ya que se espera una tasa de 1000 sucesos/s, la potencia de procesado del tercer nivel está próxima a 10^6 MIPS.

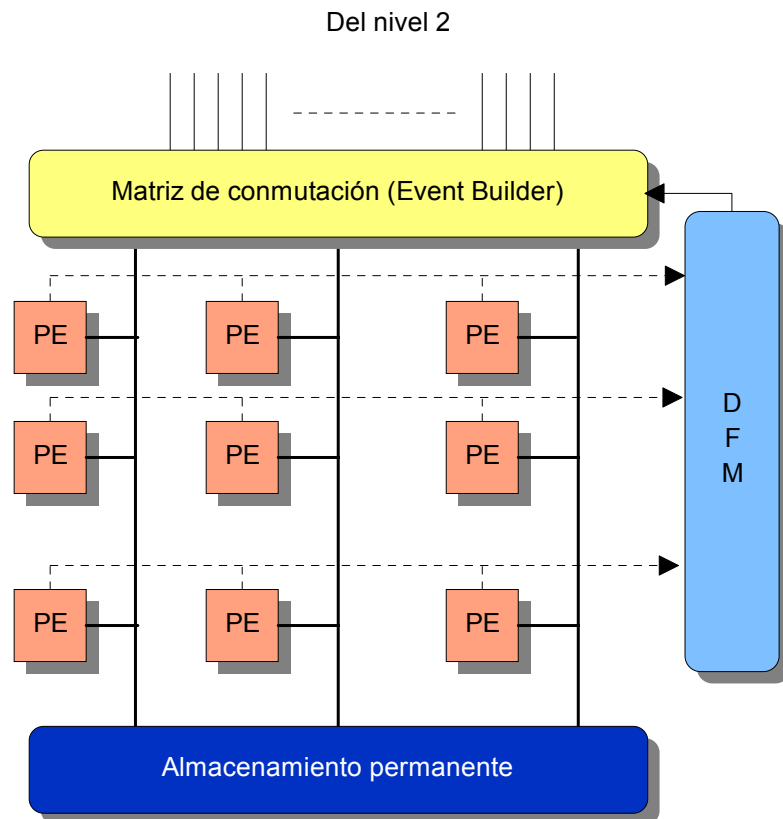


FIGURA 1.7 Diagrama de bloques del tercer nivel de trigger.

El sistema de procesado de este nivel, debido a la gran capacidad de cálculo requerida, se organizará como un conjunto de máquinas procesadoras cada una de las cuales trabajará sobre un suceso completo. Cada una de las máquinas, modelada en la figura 1.8, constará de un segmento de entrada que recibe los datos de cada suceso completo y proporciona la información al DFM, un elemento de procesado que ejecuta los algoritmos de análisis y un segmento de salida que recibe los sucesos seleccionados y comprimidos del elemento de procesado y los envía hacia el sistema de almacenamiento permanente.

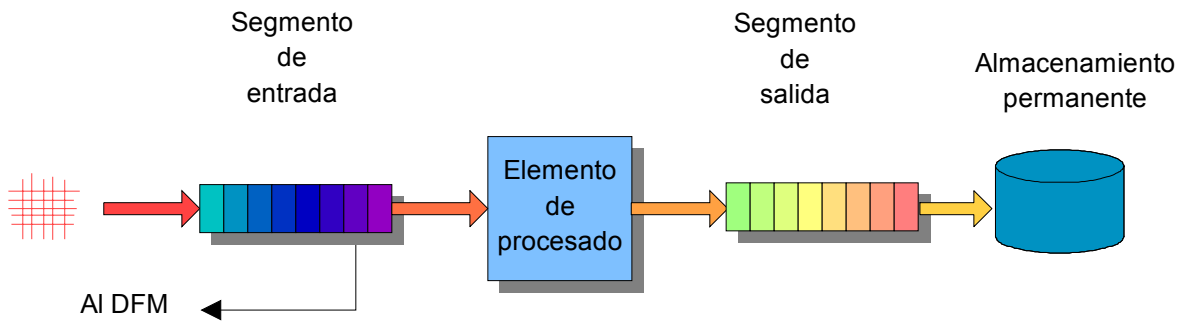


FIGURA 1.8 Esquema de cada una de las máquinas procesadoras del tercer nivel.

En definitiva, tenemos un primer nivel de trigger (LVL1) que acepta datos a una frecuencia de 40 MHz, almacenando los datos en memorias, con un tiempo de decisión de 2 μ s y una tasa máxima de salida limitada a 100 KHz hacia el segundo nivel de trigger.

El segundo nivel de trigger (LVL2) reduce la tasa de 100 KHz y de unos 150 GB/s de datos hasta 1 KHz como máximo y con un margen de entre 1 y 10 GB/s. En este nivel la información que recibe es utilizada para identificar las regiones del detector que contienen características interesantes del suceso (RoI).

Después de que un suceso es aceptado por el segundo nivel de trigger, se envían los datos al tercer nivel de trigger mediante el constructor de sucesos (Event Builder, EB). La reconstrucción total de un suceso, y la decisión aceptado/rechazado se ha de realizar en el tercer nivel en un tiempo no mayor de 1 s. Las tasas son reducidas a 10-100 Hz y 10-100 MB/s.

Hay que destacar las impresionantes cifras que ATLAS consigue tratar. Estamos hablando de procesar 50 TB/s y reducirlo a niveles finales de almacenamiento de tan sólo 10-100 MB/s.

La estructura general de este sistema de adquisición puede observarse en la figura 1.9.

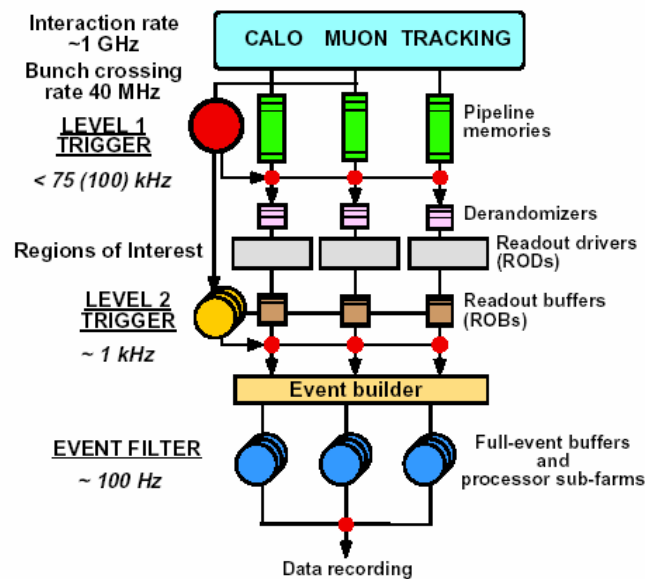


FIGURA 1.9 Sistema de adquisición de datos en ATLAS.

5. CALORÍMETRO HADRÓNICO (TILECAL)

El Calorímetro Hadrónico (TileCal) es uno de los subdetectores del detector ATLAS; se trata de un gran calorímetro de muestras hadrónicas, el cual usa el hierro como material absorbente y medio pasivo, y con platinas centelleadoras como medio activo [13]. El nuevo aspecto de este diseño es la orientación de las tejas centelleadoras las cuales son colocadas en planos perpendiculares a la colisión del haz.

TileCal presenta una buena homogeneidad en las muestras que recoge cuando el calorímetro se coloca detrás de un compartimiento electromagnético y una bobina equivalente para un total de dos longitudes de onda de material.

Consiste en una estructura cilíndrica subdividida en tejas y en una parte central y dos extendidas a los lados de la misma. A continuación se detallan los aspectos más importantes de este subdetector.

En la figura 1.10 podemos ver cómo quedan dispuestos los módulos en TileCal.

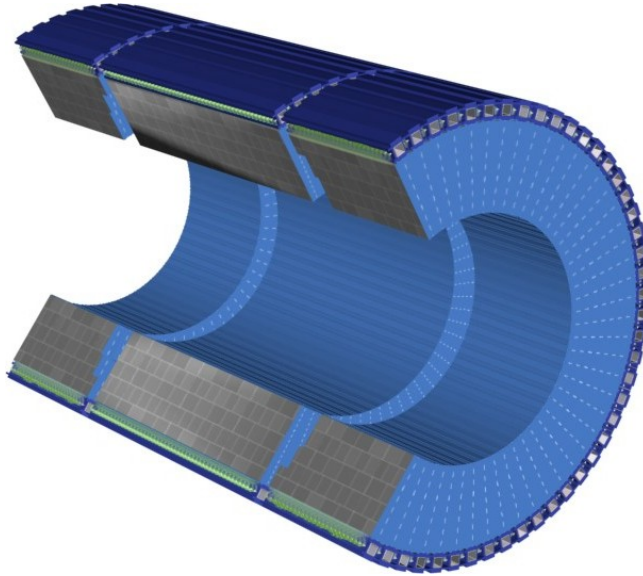


FIGURA 1.10 Estructura del calorímetro.

5.1 DESCRIPCIÓN DEL CALORÍMETRO

El Tile Calorímetro consiste en una estructura cilíndrica con un radio interior de 2.280 mm y con un radio exterior de 4.230 mm. Está subdividido en un barril central de 5640 mm de longitud y dos barriles extendidos de 2.910 mm. Cada cilindro del detector está construido en base a 64 módulos a través de la dirección azimutal. Entre los barriles existe un gap de 600 mm, el cual es necesario para los cables del Inner Detector y del Liquid Argon Detector, la electrónica y diferentes servicios.

El barril central cubre la región de $-1,0 < \eta < 1,0$, y los barriles extendidos cubren la región $0,8 < |\eta| < 1,7$. Parte del gap contiene una extensión del barril extendido, el Calorímetro de Tejas Intermedio (ITC), que mediante una estructura escalonada maximiza el volumen del material activo en esa región. La siguiente figura nos ofrece una sección del calorímetro y sus partes.

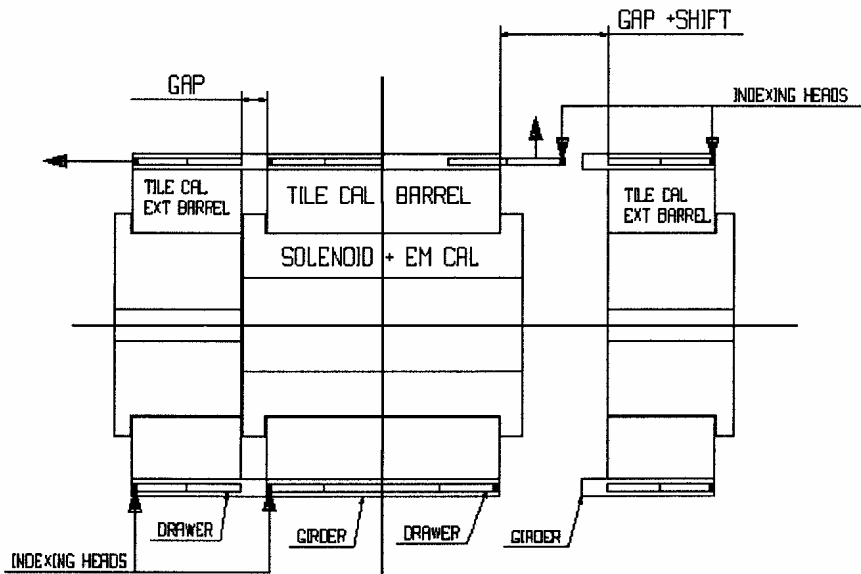


FIGURA 1.11 Sección de TileCal.

La estructura absorbente consiste en un laminado de planchas de hierro de diferentes dimensiones sujetas a un elemento principal denominado *girder*. La periodicidad de todo el sistema permite la construcción de un gran detector mediante la unión de los diferentes módulos.

Las tejas centelleadoras están dispuestas en el plano $r-\theta$ y su anchura coincide con la del módulo en la dirección θ . Las fibras colocadas radialmente son las que recogen la luz de las tejas. Estas fibras se agrupan definiendo celdas de lectura que posteriormente son leídas por fotomultiplicadores (PMT).

El uso de fibras como elemento de lectura permite definir una celda tridimensional, creando una geometría proyectiva para la reconstrucción de la energía. Un sistema de lectura se inserta en el *girder* de cada módulo, permitiendo una primera adquisición de datos. El principio fundamental del diseño del calorímetro se define claramente en la siguiente figura.

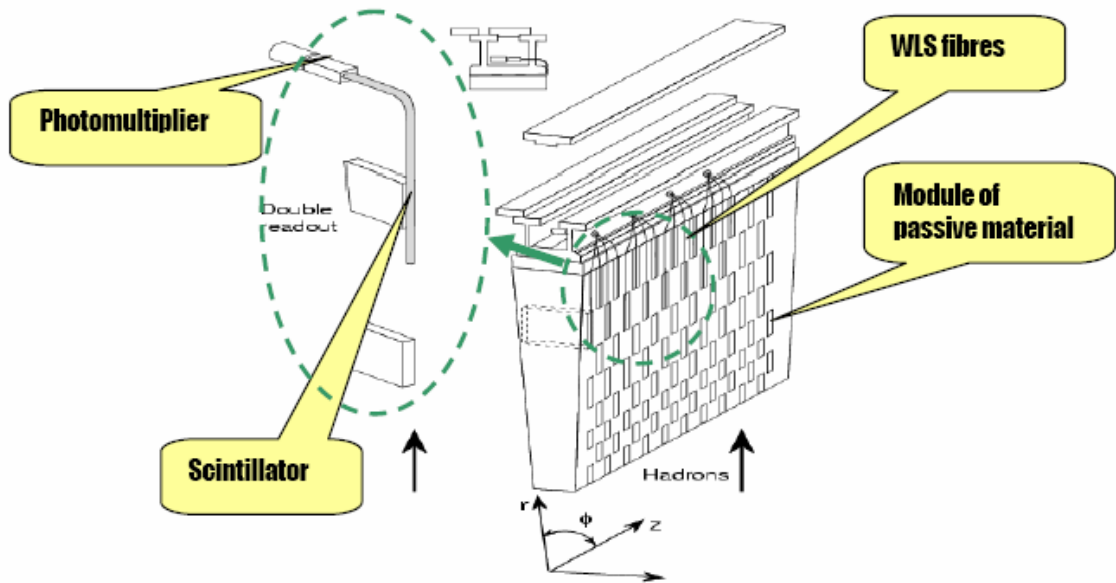


FIGURA 1.12 Sistema de lectura de un módulo en TileCal.

En resumen, las consideraciones básicas para el diseño del detector se pueden resumir en: una alta resolución en energías para todo el intervalo de η , una buena linealidad en respuesta al rango de unos pocos GeV hasta TeV, una excelente uniformidad en las direcciones η y θ , y por último una buena hermeticidad para evitar zonas muertas en el detector.

5.2 ASPECTOS MECÁNICOS, ÓPTICOS Y ELECTRÓNICOS

El desarrollo de un subdetector como TileCal ofrece una serie de características que lo definen por completo. En primer lugar hablaremos de los aspectos mecánicos que presenta en calorímetro.

La estructura de cada uno de los módulos o sectores de TileCal consiste en un apilamiento de elementos de forma repetida (periodo). Cada periodo está compuesto por láminas de hierro de forma trapezoidal (*master plates*) de 5 mm de espesor, que cubren radialmente toda la dimensión del módulo. En la segunda y cuarta capa, pequeñas láminas de hierro también de forma trapezoidal (*spacer plates*) se alternan con tejas centelleadoras a lo largo de la dirección radial.



FIGURA 1.13 Montaje del periodo de un submódulo del calorímetro.

Los *spacer plates* son de 4 mm de espesor y de 11 tamaños diferentes según su posición. Las tejas centelleadoras de 3 mm de espesor se insertan en los módulos después del proceso de unión de las láminas de hierro. Una guía rígida en la parte externa y una lámina metálica de 10 mm de espesor en el radio interno, proporcionan una estructura compacta y permiten la unión de unos módulos sobre otros; asegurando la estabilidad y la rigidez mecánica.

La división en sectores del calorímetro da como resultado la segmentación en θ . Radialmente cada sector está formado por tres capas denominadas *samplings*, correspondiéndole a cada uno de estos *samplings* un número diferente de celdas de lectura.

El primer *sampling* (*A*) es el más interno, contiene las tejas centelleadoras más pequeñas, las denominadas 1, 2, 3 de menor a mayor. Una fibra lee las tejas 1 y 3, la otra fibra lee la teja 2. El radio medio es de 2,4 m y consta de 10 celdas de lectura por módulo para $\eta > 0$, análogamente para $\eta < 0$.

El segundo *sampling* (*BC*) está compuesto por dos secciones, de esta forma se aumenta la proyectividad en η . La primera sección es leída por dos fibras y contiene las tejas de clase 4, 5, 6; una fibra lee la teja 5, otra las tejas 4 y 6. La segunda sección contiene las tejas 7, 8, 9; y del mismo modo, una fibra lee las tejas 7 y 9 y la otra la teja 8. Los radios medios son de 2,8 m y 3,2 m respectivamente. Hay 9 celdas de lectura por módulo para $\eta > 0$ y otras tantas para $\eta < 0$.

El tercer *sampling* (*D*) es el más externo y está formado por las tejas 10, 11; cada una de ellas leída por una fibra. Posee 4 celdas de lectura para $\eta > 0$ y otras 4 celdas para $\eta < 0$.

En la figura siguiente podemos observar como están distribuidos estos *samplings*.

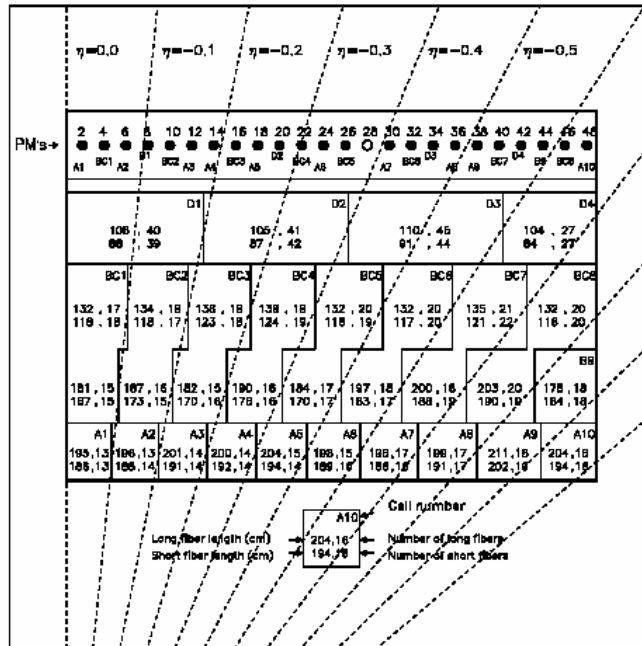


FIGURA 1.14 Segmentación de las celdas de lectura en TileCal.

Una vez descrita la parte mecánica del detector, los siguientes elementos a tener en cuenta son los ópticos y los electrónicos. En cuanto a la parte óptica, hay que tener en cuenta que la luz producida por las tejas centelleadoras de 3 mm de espesor es transmitida por las fibras WLS (*Wavelength Shifting Fibres*) de 1 mm de diámetro situadas a ambos lados del centelleador.

Como se ha comentado antes, las fibras se agrupan formando mazos. Antes de llegar a esos mazos, éstas se insertan en unos envoltorios (*profiles*) que cumplen la función de reunir a las fibras en un mismo grupo y de contribuir al aumento de la resistencia mecánica. Estos *profiles* son opacos para evitar fugas de luz y transmisión de unos a otros.



FIGURA 1.15 Agrupamiento de las fibras ubicadas en el interior de la viga.

Posteriormente, la luz llega a los fotomultiplicadores (PMTs) que finalmente convertirán la señal luminosa en un pulso eléctrico. Estos PMTs junto con la electrónica asociada están a su vez insertados en un sistema móvil o cajón (*drawer*), incluido dentro de la propia viga. Esto permite acceder fácilmente a la electrónica sin afectar al resto del detector. En total tendremos 512 drawers idénticos.

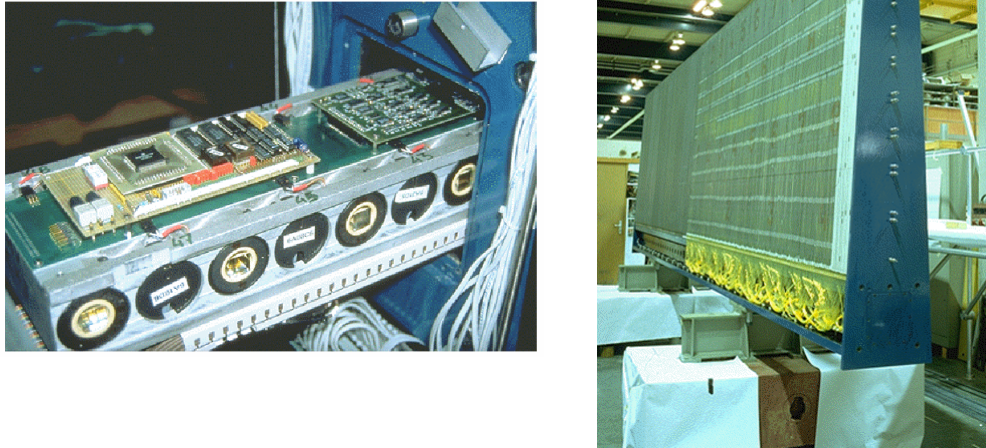


FIGURA 1.16 Viga y *drawer* insertado en el *girder* como sistema de lectura.

La unidad del PMT contiene un difusor de luz (*light mixer*), el fototubo con su electrónica y una envoltura magnética de μ -metal que lo protege del campo magnético del solenoide central. A continuación, la base del PMT con el correspondiente divisor de tensión y la tarjeta “3 en 1”, que se describirá más tarde. En principio, la alta tensión se suministra a cada uno de los *drawers* y después, dentro de los *drawers*, existen unos controladores que distribuyen individualmente la tensión a cada PMT.

Finalmente se debe describir la electrónica que se usa en el calorímetro, parte en la cual se ha desarrollado esta Tesis Doctoral. La electrónica está diseñada para sacar la mayor ventaja posible a la alta luminosidad de lectura, con el mínimo ruido posible limitando el número de canales a leer.

El diseño es simple y compacto, todo el *front-end* (primera parte de lectura) y la electrónica digitalizadora está situada en los *drawers* comentados anteriormente y directamente instalados en los submódulos del detector. La información que se digitaliza en estos *drawers* se envía directamente y por fibra óptica al sistema Read Out Driver (ROD, centro de estudio y diseño de esta Tesis). En el siguiente punto se detallará exactamente todo el sistema ROD.

Cada celda del Calorímetro es leída por dos PMTs para tener una redundancia en los datos y mejorar la uniformidad espacial. Es importante resaltar que tenemos dos fibras con la misma información para evitar que en el proceso de envío de los datos al ROD, dicha información resulte corrupta por la radiación existente en el calorímetro.

El número total de canales es aproximadamente de 10.310, en la siguiente tabla se puede comprobar cómo está dividido el calorímetro. Hay que tener en cuenta que un *súper-drawer* físicamente consiste en dos *drawers*.

	Número de celdas en el Calorímetro	Número de canales de lectura (Read-Out)	Número de Súper- Drawers
Sección de Barril	46	92	2
<i>Total de Barril</i>	2990	5980	130
Sección de Barril Extendido	14	28	1
<i>Total de Barril Extendido</i>	1820	3640	130
Sección de ITC	3	3	
<i>Total de ITC</i>	390	390	

TABLA 1.4 Clasificación de los canales de lectura en el TileCalorímetro.

El diseño general de la lectura electrónica se basa en dos subsistemas modulares, los bloques PMTs y los súper-drawers. El bloque de PMTs es un ensamblaje de tres partes principales (un mezclador luminoso y un PMT, un divisor y el ya mencionado “3 en 1”). El “3 en 1” consiste en un shaper dedicado al análisis del pulso, un integrador que mide la corriente de las calibraciones de Cesio y los eventos minimum bias (se explican con mayor detalle en el capítulo 4) y un control lógico.

Toda esta parte es lo que conocemos como Primer Nivel de Trigger dentro del sistema de adquisición de datos de TileCal. Tenemos en total 256 súper-drawers en ATLAS, con aproximadamente 24 bloques PMT en cada drawer.

El siguiente paso dentro de la electrónica es el Read Out Driver (ROD), encargado de recoger la información del primer nivel de trigger. Su misión es en general, darle formato a los datos, realizar un preprocesado de los mismos y enviarlos a los Read Out Buffers (ROBs). Más tarde estos llegarán al Segundo Nivel de Trigger. Veremos en el siguiente capítulo y con mucho más detalle en qué consisten los RODs y las funciones que realizan.

6. BIBLIOGRAFÍA

- [1] Whyte, H. E., *Introducción a la física atómica y molecular*, Ed. Alhambra, 1970.
- [2] Myers, S., *The LEP collider, from design to approval and commissioning*, CERN Yellow Report 91-08, 1991.
- [3] DELPHI collaboration, *Measurement of the mass and width of the Z^0 particle form multihadronical final states produced in $e^+ e^-$ annihilations*, Physics Letters B, Vol. 231, Núm. 4, 1989.
- [4] Evans, L. R., *The Large Hadron Collider*, 1995 Particle Accelerator Conference, Dallas, EEUU, 1995.
- [5] The LHC Study Group, *The Large Hadron Collider Accelerator project*, CERN AC/93-03-(LHC), 1993.
- [6] ATLAS Collaboration, *ATLAS Technical Proposal*, CERN/LHCC/94-43, 1994.
- [7] CMS Collaboration, *CMS Technical Proposal*, CERN/LHCC/94-38, 1994.
- [8] ALICE Collaboration, *Letter of Intent for a Large Ion Collider Experiment*, CERN/LHCC/93-16, 1993.
- [9] LHC-B Collaboration, *A dedicated LHC collider beauty experiment for precision measurement of CP violation*, CERN/LHCC 95-5, 1995.
- [10] Brawn, I. P. et al., *The level-1 calorimeter trigger system for ATLAS*, ATLAS-DAQ-NO-30, 1995.
- [11] Bock, R. et al., *More readout specifications (LVL2 demonstrator program)*, Addendum a la nota interna DAQ-NO-53, 1996.
- [12] Clarke, P. E. L. et al., *Test of components for an asynchronous level-2 trigger for ATLAS*, ATLAS/DAQ-NO-36, 1994.
- [13] TileCal Collaboration, *TileCal Technical Design Report*, CERN/LHCC 96-42, 1996.

CAPÍTULO

2

PROTOTIPO DEL SISTEMA ROD Y DESARROLLO DEL MÓDULO DE TRANSICIÓN TM4PLUS1

1. INTRODUCCIÓN	1
2. PROTOTIPO DEL SISTEMA READ OUT DRIVER (ROD)	1
2.1 Descripción del sistema ROD	1
2.2 ROD Motherboard	4
2.3 DSP Processing Units	6
2.4 Algoritmo de Filtrado Óptimo	7
3. MÓDULO DE TRANSICIÓN (TM4PLUS1)	10
3.1 Descripción del Módulo de Transición	10
3.2 Interface S-LINK	12
3.3 Dispositivos Lógicos Programables	14
3.4 Metodología Diseño FPGAs	18
3.5 Reformatting Altera FPGA	19
3.5.1 Bloque FIFO	19
3.5.2 Bloque connect FIFO	20
3.5.3 Bloque control S-LINK	23
3.5.4 Bloque data multiplexing, control & filtering unit	23
3.5.5 Bloque auxiliary FPGA	26
3.5.6 Bloque reformatting FPGA control & test	27
3.6 Auxiliary Altera FPGA	27
3.6.1 Bloque S-LINK	27
3.6.2 Bloque reformatting FPGA	28
3.6.3 Bloque J2B connection	28
3.6.4 Bloque data control unit	28
3.6.5 Bloque auxiliary FPGA clock, control & test	28
3.6.6 Bloque ODIN LSC (G-LINK)	28
4. FORMATO DE DATOS DEL ROD	30
4.1 Estructura de los Datos de Salida	30
5. BIBLIOGRAFÍA	33

1. INTRODUCCIÓN

En el capítulo anterior se ha considerado, desde un punto de vista general, el experimento LHC del CERN y el detector ATLAS con el subdetector TileCal donde se centra esta Tesis. A continuación, vamos a dar a conocer los diferentes aspectos particulares del sistema de adquisición de ATLAS, el cual se divide en tres niveles. El sistema Read Out Driver (ROD) es un elemento imprescindible dentro de esta adquisición, que se sitúa entre el primer nivel de trigger y el segundo nivel de trigger. Daremos una descripción del mismo tanto a nivel hardware como a nivel de las necesidades que debe cumplir.

Después de encuadrar el ROD en la estructura de adquisición de datos de ATLAS y de ver las características de dos de los elementos fundamentales del mismo (Motherboard y PUs), en este capítulo también se va a estudiar en profundidad el Módulo de Transición Activo (Tm4Plus1).

Veremos una descripción detallada de sus características principales, los elementos que lo componen, un estudio de los dispositivos programables incluidos en él, los protocolos de comunicación implementados en las FPGAs y los desarrollos realizados para tratar y dar formato a los datos que recibe del primer nivel de trigger.

Se ha trabajado tanto en su diseño como en su test, su programación y su puesta en marcha en el Test Beam del CERN. Trabajando así mismo en colaboración directa con el desarrollo del RODDemo y del Sistema ROD final.

2. PROTOTIPO DEL SISTEMA READ OUT DRIVER (ROD)

El Read Out Driver [1] de TileCal es un sistema electrónico formado por una motherboard y un módulo de transición. Su función principal es la de leer y computar los 9856 canales del calorímetro cada 10 μ s, y tiene que ser capaz de distribuir una gran cantidad de datos en tiempo real.

Los datos recogidos del detector, ya digitalizados, son enviados a los RODs mediante enlaces ópticos de alta velocidad. Cada módulo ROD se encarga de procesar dichos datos y de enviarlos hacia el siguiente nivel de adquisición (ROBs) también mediante enlaces ópticos.

2.1 DESCRIPCIÓN DEL SISTEMA ROD

El sistema ROD, está construido en base a dos tarjetas VME independientes (motherboard y módulo de transición) [2], capaces de analizar hasta 2Gbytes/s de información. Para la lectura de todos los canales que presenta el calorímetro TileCal, se está trabajando en el uso de 32 módulos ROD. Cada uno de ellos procesará unos 300 canales y se tratarán sólo los datos que pasaron el primer nivel de filtrado de datos.

En la figura 2.1 se puede observar con claridad cómo está desarrollado el sistema de adquisición de datos hasta el módulo ROD. Cada uno de los barriles de TileCal (barril central dividido en dos partes y dos extendidos) presenta 64 módulos, cada uno de ellos con un drawer (en total 256). Cada drawer lee 45 canales si se trata del barril central y 32 canales si hablamos de los extendidos; por tanto estamos analizando los 9856 canales comentados con anterioridad.

De cada canal tenemos dos fibras ópticas que envían los datos al ROD (recordemos de nuevo la redundancia de datos para evitar la pérdida de los mismos por radiación). A cada ROD le llegan 8 fibras ópticas de 4 canales de lectura, con lo cual necesitaremos de 32 RODs para analizar todo el calorímetro [3].

Cada ROD está instalado en crates VME, necesitando cuatro crates para instalar los 32 RODs necesarios. De cada ROD parten 2 fibras ópticas con los datos ya

analizados hacia el siguiente nivel de adquisición, sumando 64 fibras que se distribuyen en otros tantos ROB's.

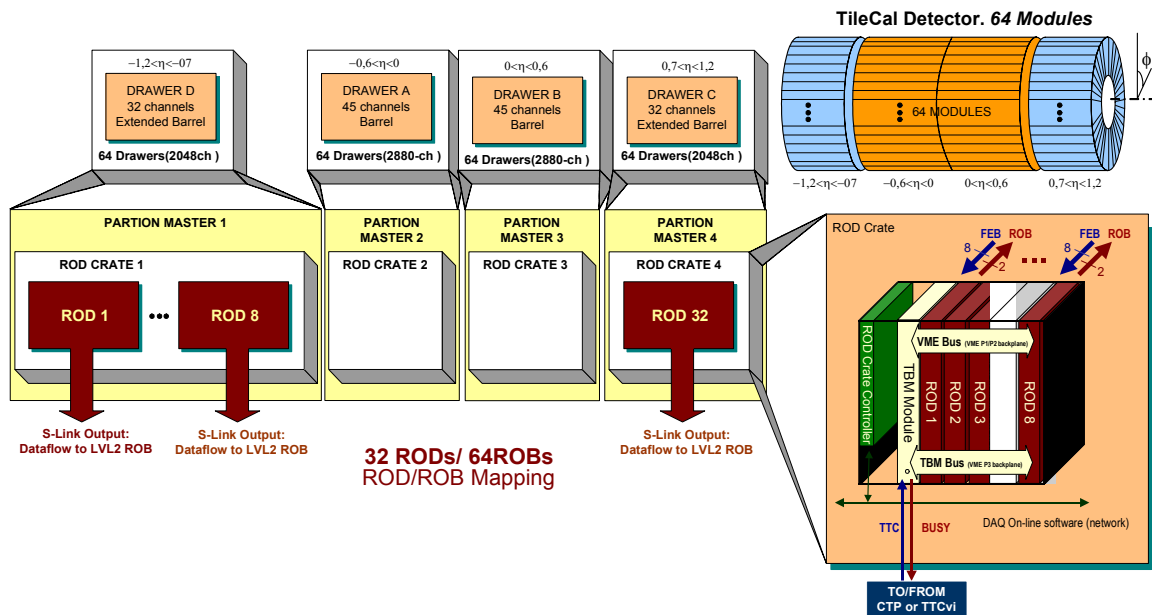


FIGURA 2.1 Sistema ROD de TileCal.

Las especificaciones generales del sistema ROD se pueden observar en la siguiente tabla y se pueden resumir diciendo que además de recibir los datos y volverlos a enviar, para cada celda del calorímetro y a partir de 7 muestras de 10 bits el ROD calculará la energía depositada en cada celda, el tiempo de llegada y una estimación de la forma del pulso. Así mismo, está encargado de comprobar el inicio y fin de cada paquete enviado y de que todo el sistema está actuando en tiempo real, sin ningún tipo de retardo adicional al propio de la tecnología usada.

Número de canales	9856
Número de enlaces fibra óptica	256 (512 con redundancia datos)
Ancho de banda de entrada total (Gbytes/s)	14,03 (24,06 con redundancia datos)
Número de enlaces por módulo ROD	8 (16 con redundancia datos)
Número de RODs	32
Tiempo de proceso con trigger de 100 KHz en tiempo real	10 μ s
Capacidad de procesado en MIPS	68.992
Número de enlaces ópticos de salida	64
Ancho de banda de salida total (Gbytes/s)	6,7

TABLA 2.1 Características generales del sistema ROD.

Además de las funciones principales comentadas en el párrafo anterior, el sistema ROD debe estar en continua comunicación para poder monitorizar y controlar todos los módulos ROD. Para ello cada crate debe incluir, tal y como se puede observar en la figura 2.1, un ROD Controller (se trata de un Single Board Computer) que es una CPU Master que controla los módulos ROD (dispositivos Slave). Así mismo, debe incluir un Trigger and Busy Module (TBM) que es responsable de la recepción de la información del Timing, Trigger and Control system (TTC) en la ROD crate y de distribuirlo a los módulos ROD [4].

Con ello tenemos comunicación directa con el sistema general de ATLAS para saber cuándo tenemos eventos, también nos permite en caso de error generar una señal ROD Busy que para la adquisición del Nivel 1 de Trigger. Por tanto, tenemos una comunicación bidireccional con el Central Trigger Processor y con el ATLAS Data Adquisition Control (DAQ). Toda la adquisición de datos en ATLAS está comunicada vía Online Software para monitorizar y controlar en todo momento el proceso de los datos analizados. En esta figura se puede observar la comunicación continua que tiene el ROD con el ATLAS DAQ.

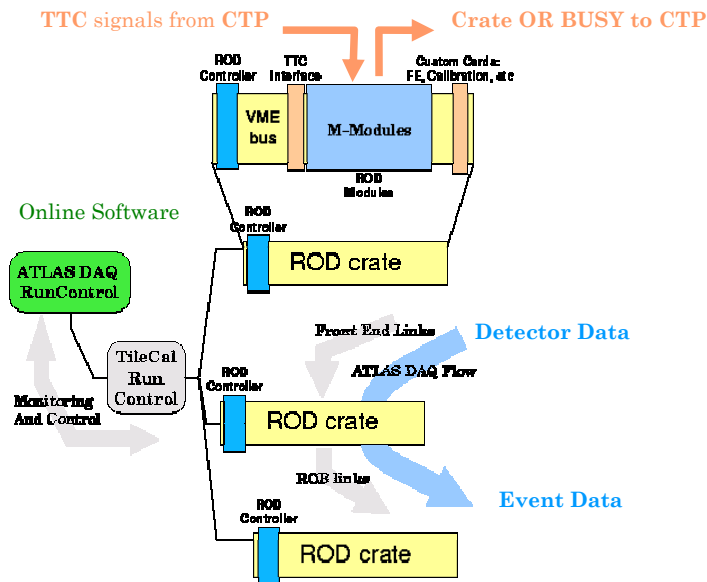


FIGURA 2.2 Conexión con ATLAS DAQ Control.

Resumiendo, las funciones básicas que realiza el ROD son:

- Procesado de datos. Encargado de compactar la enorme cantidad de datos que provienen del Primer Nivel de Trigger.
 - Proporciona una estimación de la energía, el tiempo y la forma del pulso de cada evento recogido mediante algoritmos de Filtrado Óptimo.
 - Posibilidad de no realizar procesado alguno en caso de apilamiento de eventos o alta energía en los mismos.
- Trigger. Las señales del Timing, Trigger and Control system (TTC) están presentes (con una latencia de 2 μ s después del L1Accept) en cada módulo ROD, proporcionando señales como ROD L1ID (identificador del primer nivel de trigger) y ROD BCID (identificador del cruce de haz). Ambas señales nos identifican correctamente el evento producido.
- Detección de errores. Encargado de sincronizar las tareas del trigger, debe chequear que el L1ID y el BCID enviados por el TTC coinciden con los recibidos del Front-End (datos que vienen directamente del calorímetro).

Si detecta cualquier error o no coincidencia, debe enviar una señal de error.

- Enlaces de lectura. Los datos ya procesados por el ROD deben ser enviados al siguiente nivel de procesamiento a través de unos estándares de ATLAS y con un estándar de datos DAQ-1 (estándar del CERN para el *Data Acquisition*) a la frecuencia del L1Accept (100 KHz).
- Generación de una señal de ocupado. Capaz de producir una señal ROD Busy en caso de estar ocupado o sin capacidad de proceso, para detener la adquisición de datos del L1.
- Monitorización local. Acceso por VME [5] a los datos durante una secuencia sin introducir tiempos muertos o latencias en el DAQ. Cada motherboard del ROD es un Módulo VME Slave reconocido por el ROD Controller.

A continuación, y siguiendo un orden de definición de mayor a menor, pasamos a conocer los dos elementos que componen el sistema ROD (motherboard y módulo de transición).

2.2 ROD MOTHERBOARD

El sistema ROD que actualmente tenemos es un prototipo del sistema final y nos debe servir como paso intermedio para la construcción del módulo ROD completo para el calorímetro hadrónico. En este apartado haremos una descripción detallada de la motherboard del RODDemo [3] y en el siguiente capítulo de esta Tesis Doctoral se mostrarán los nuevos diseños del ROD final.

El ROD consta de una tarjeta VME 9U dividida en dos partes, una motherboard con las conexiones de entrada/salida, controles e interfaz VME y cuatro DSP Processing Units (PUs) montadas encima de la motherboard donde se realiza el procesamiento de los datos.

Las principales características de esta tarjeta RODDemo son:

Ítem	
Tipo de Crate	VME64x standard, con adaptador P3 back-plane
Tipo de tarjeta	9U
Dimensiones de la tarjeta	366,7 x 400 mm
DSP Processing Units	4
Dimensiones tarjeta PU	85 x 185 mm
Número de canales por tarjeta	256
Número de canales por PU	64
FEB links	2 (32 bits @ 40 MHz)

TABLA 2.2 Características generales de la motherboard del ROD.

Desde el punto de vista de la motherboard, las PUs son vistas como FIFOs donde los datos están continuamente entrando según el reloj. Los algoritmos de Filtrado Óptimo se implementarán en estos DSPs.

Los datos los recoge primero el módulo de transición que después de procesarlos y adecuarlos (lo veremos con mayor detalle en puntos posteriores), los envía a la motherboard. En las PUs se filtran y se adaptan para volver de nuevo al módulo de transición que se encargará de enviarlos hacia el siguiente nivel de trigger.

En este diagrama de bloques lo podremos ver con mayor claridad.

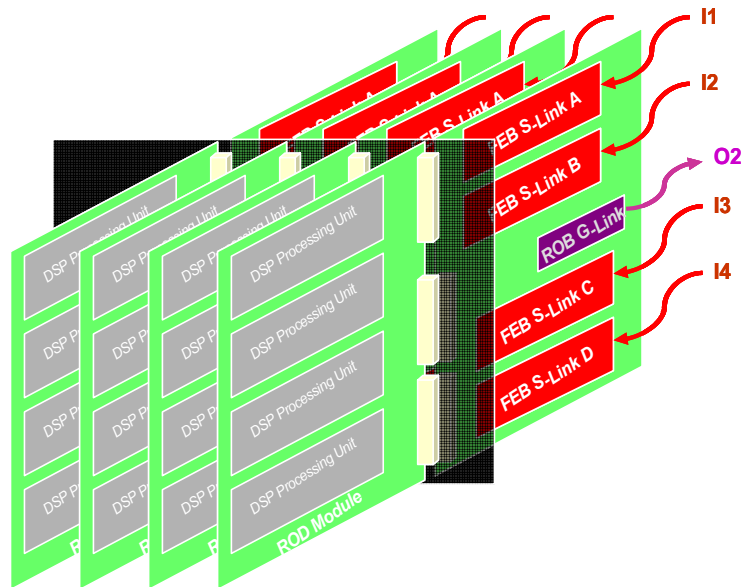


FIGURA 2.3 Diagrama de bloques del sistema RODDemo.

Centrándonos en la motherboard, los elementos que presenta son:

- Entradas y salidas de datos. Todas las entradas y salidas de datos de la motherboard se producen por los conectores VME P2 y P3 que están conectados al módulo de transición.
- TTC FPGA. Recoge la información que envía por medio de VME el TTC Controller y lo distribuye en la motherboard y en las PUs.
- Output Controller. Recibe y guarda la información del TTC la cual se envía por el backplane vía VME. A su vez, lee los eventos de las cuatro PUs dándoles formato para obtener los fragmentos completos del ROD que se enviarán al módulo de transición para su posterior salida al ROB.
- Interfaz VME. En general, el módulo ROD se considera un módulo esclavo. Todas las acciones y comandos son controlados desde una CPU local. Mediante este interfaz podemos cargar y configurar las motherboard y los componentes de las PUs, realizar el proceso de calibración de los pesos incluidos en el algoritmo de filtrado, monitorizar la información del ROD y enviar al ATLAS DAQ Control las señales de error que genere el sistema.

La figura 2.4 nos muestra cómo están distribuidos estos elementos dentro de la motherboard y nos ofrece una primera visión de los componentes del módulo de transición. Así mismo, podemos observar el camino que siguen los datos (líneas en color rojo), el camino que siguen los comandos del TTC (líneas en azul) y la conexión VME.

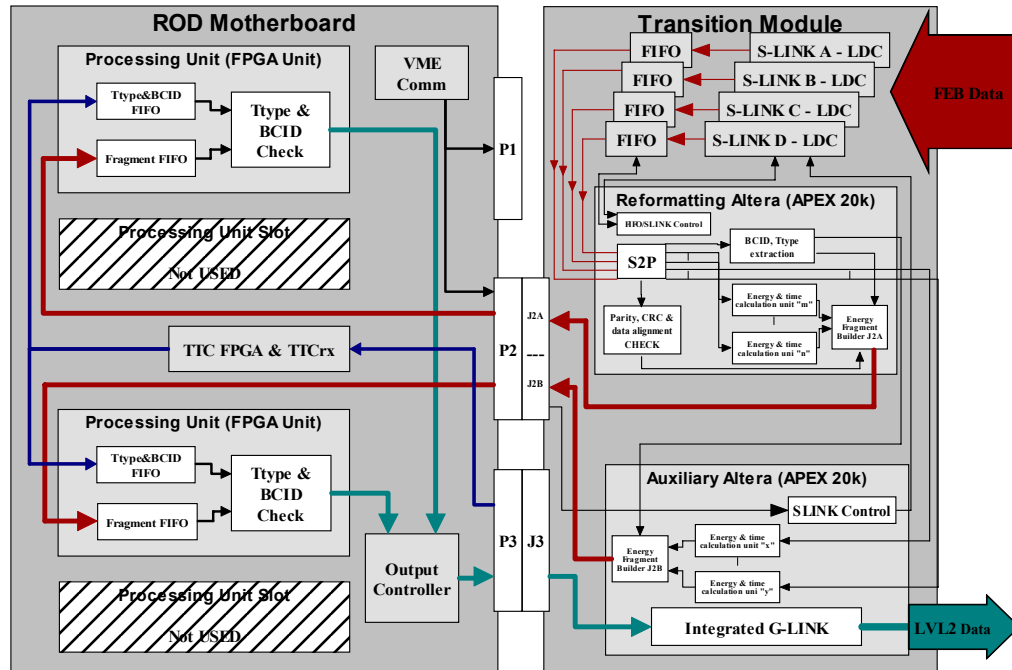


FIGURA 2.4 Componentes del sistema ROD.

2.3 DSP PROCESSING UNITS

El siguiente elemento que vamos a analizar y describir son las Unidades de Proceso [3] donde están ubicados los DSPs. Podemos observar en el diagrama de bloques anterior que, de las cuatro PUs que podemos conectar en la motherboard, sólo se usarán dos de ellas al tratarse de un módulo inicial de test. Pasamos a conocer con mayor detalle las características principales de las DSP Processing Units.

Los requerimientos que deben cumplir las PUs son:

- Función de buffer de los datos de entrada y de salida. Absorbe las fluctuaciones dadas por los diferentes eventos y algoritmos usados, se estima que puede retener unos 100 eventos.
- Implementación de los algoritmos de Filtrado Óptimo. Mediante los cuales podemos evaluar la energía, el tiempo y la calidad del pulso para todos los canales.
- Realiza el primer paso en el análisis de los datos de calibración. Además ha de ser capaz de producir la señal de Busy cuando los buffers estén casi llenos.

Como se puede ver en la figura 2.5, estas PUs contienen cuatro elementos principales:

- Input FPGA. Capaz de leer y chequear la entrada de datos antes de enviárselos al DSP mediante un conversor serie-paralelo (S2P) y con lectura directa de la información del TTC.
- El DSP con memoria local. Se trata de un TMS320C6202 a 250 MHz donde se implementarán los algoritmos ya comentados y que veremos con más detalle un poco más adelante.
- Dual Port Memory. Donde se almacenan los eventos antes de ser leídos por el DSP, a su vez también sirve para inicializar el programa interno del DSP y la memoria de datos.
- Output FPGA. Una vez el evento ha sido procesado, el DSP escribe los datos de salida (con el formato adecuado) en la FIFO de salida. Está preparado para ser leídos desde el Output Controller de la motherboard.

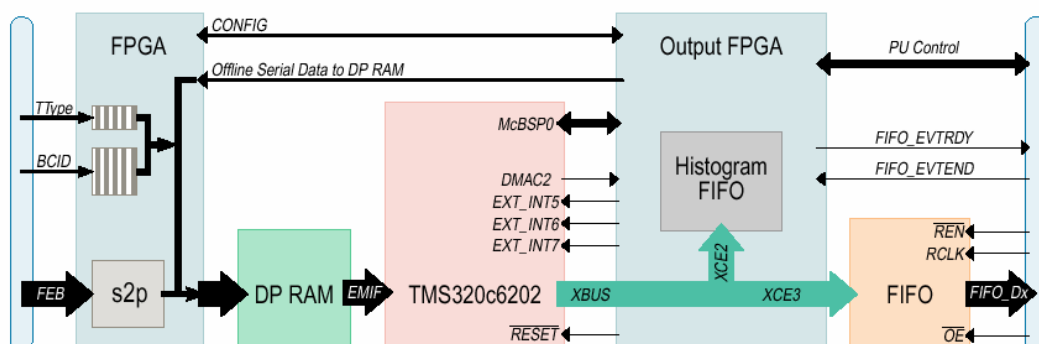


FIGURA 2.5 Estructura interna de las PUs.

La implementación exacta de estos componentes depende del DSP que finalmente se elija y del tipo de necesidades que tengamos para el tratamiento de los datos. El diseño de la motherboard se ha realizado de forma que sea lo más flexible posible con el fin de poder acoplar diferentes PUs según la evolución del mercado.

2.4 ALGORITMO DE FILTRADO ÓPTIMO

Como paso final en la explicación de la motherboard y una vez conocidos los detalles de la misma y de las PUs, pasamos a definir uno de los algoritmos que se implementarán en los DSPs. Este algoritmo es el de Filtrado Óptimo (Optimal Filtering) [6].

Uno de los puntos clave de la calorimetría es la transformación de la señal obtenida de las unidades de lectura en energía depositada en las mismas. El caso que nos ocupa en esta Tesis Doctoral, el del calorímetro TileCal, trata la señal de los fotomultiplicadores digitalizándola en muestras de 25 ns (como ya hemos comentado en puntos anteriores se trata del tiempo transcurrido entre dos colisiones del LHC).

El algoritmo de Filtrado Óptimo transforma dichas muestras digitales en Energía (la depositada en su respectiva celda), el Tiempo de llegada de la señal y la forma del Pulso generado. A su vez, minimiza el ruido producido por los componentes electrónicos y el ruido físico que proviene de los sucesos de Minimum Bias (conjunto

de colisiones inelásticas entre protones causante de depositar energías no legibles en el detector).

El algoritmo, forzosamente sencillo para cumplir con los requisitos de rapidez, se basa en ponderar las muestras digitales, con pesos previamente calculados, a fin de dar mayor importancia a aquellas muestras cuya relación señal ruido es mayor. El siguiente diagrama de bloques nos muestra su forma.

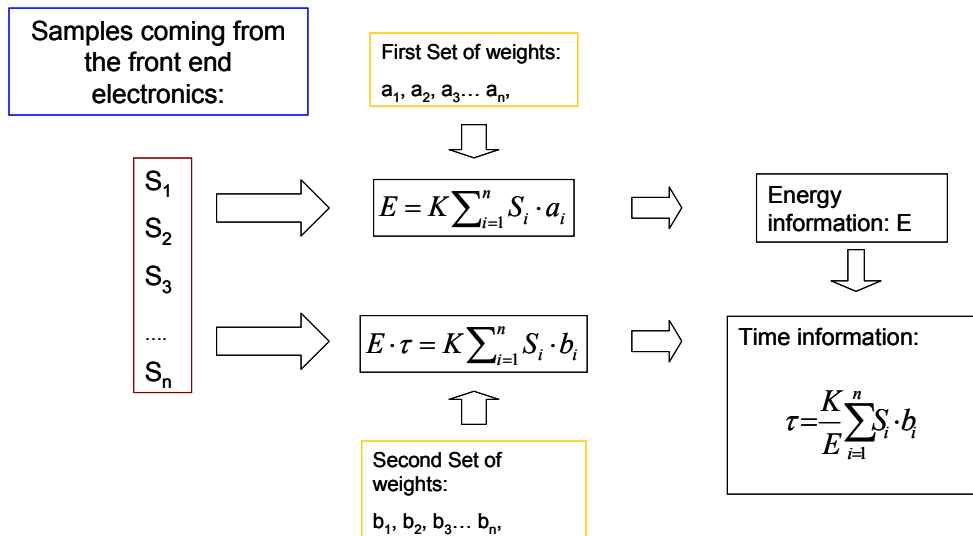


FIGURA 2.6 Algoritmo de Filtrado Óptimo.

Es importante comprobar el rendimiento del algoritmo y para ello se ha usado un algoritmo de reconstrucción como Flat Filtering (FF), consistente en la simplificación de igualar los pesos a la unidad. Los datos para la reconstrucción están tomados de uno de los módulos de TileCal sometido al haz de pruebas proporcionado por el acelerador SPS del CERN (lo veremos con mayor detalle en el capítulo 4 donde mostraremos el sistema de calibración, el Test Beam montado en Ginebra y los resultados obtenidos en él) consistente en muones, electrones y piones a altas energías.

No obstante, se eligen sucesos con baja energía depositada, del orden de unos pocos GeVs. Es ahí donde la eficacia del algoritmo se hace patente ya que la resolución a bajas energías viene dominada por el ruido electrónico.

En las siguientes figuras podemos observar cómo el algoritmo de filtrado óptimo (OF) reduce en un factor dos la sigma del ruido electrónico frente al otro algoritmo (FF) [7].

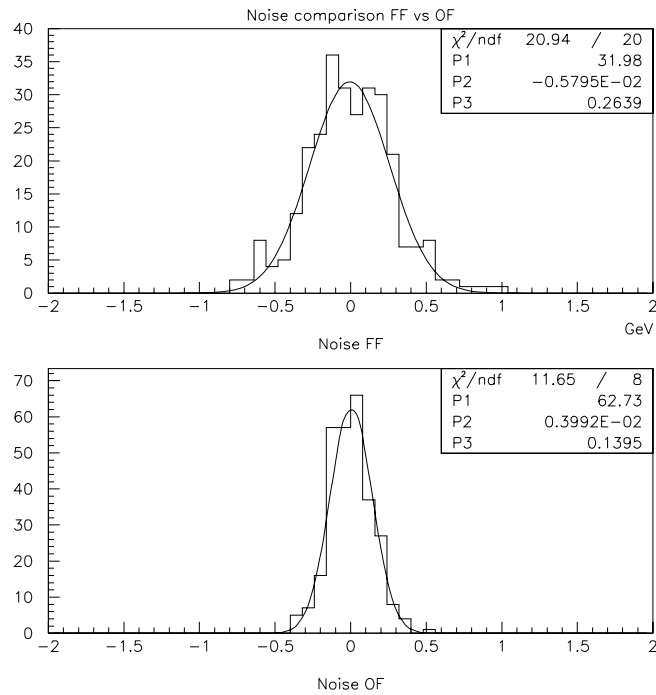


FIGURA 2.7 Reconstrucción de ruido con ambos algoritmos.

En cuanto a su eficacia en sucesos de física, se ve visiblemente reducida por efecto de las fluctuaciones estadísticas de los fenómenos físicos envueltos. En la siguiente figura se puede ver, sin embargo, que la mejora es apreciable por la buena reconstrucción de la distribución de Landau (convolucionada con una Gaussiana) que caracteriza la energía depositada en estos sucesos.

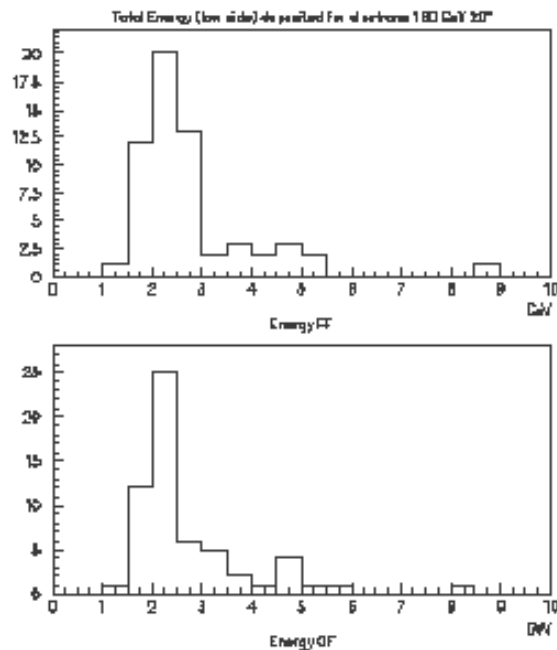


FIGURA 2.8 Reconstrucción de sucesos de baja energía.

3. MÓDULO DE TRANSICIÓN (TM4PLUS1)

El Módulo de Transición [8] es un interfaz entre el front-end de TileCal, la motherboard del ROD y los links de salida de este sistema hacia el ROB. Es un paso intermedio en la adecuación de los datos que se reciben y los algoritmos de filtrado que se implementan en las PUs de la Motherboard. Así mismo, está encargado de rutar los datos al siguiente nivel de trigger ya con el formato del ROD.

Es un módulo que se inserta en la parte trasera del ROD, conectado a él por VME y que tiene la capacidad de incorporar cuatro tarjetas S-LINK [9] (veremos más adelante en que consisten estas tarjetas) para la entrada de datos y una tarjeta ODIN integrada para la salida de datos. Mediante las dos FPGAs incluidas en el módulo, podemos realizar operaciones de filtrado, multiplexado y comprobación de los datos que entran al ROD así como la implementación de protocolos de comunicación entre los diferentes módulos presentes en él.

3.1 DESCRIPCIÓN DEL MÓDULO DE TRANSICIÓN

El módulo activo Tm4Plus1 [10] es un sistema VME64x insertado en una crate 9U que presenta slots de conexión para cuatro tarjetas S-LINK y que tiene integrada una tarjeta ODIN. En los slots S-LINK se puede conectar cualquier modelo de tarjeta LDC (link destination card). El módulo se ajusta en la parte trasera de las nuevas extensiones de las crates VME64 y puede usar cualquier entrada/salida de las conexiones con la motherboard para la transferencia de datos.

Es capaz de leer 8 x 32 bits de datos a 40 MHz, cada tarjeta LDC conectada al módulo permite la conexión de dos fibras ópticas provenientes del detector. Así pues, nos permite tratar 4 canales con redundancia en los datos. Sus características principales se detallan en la siguiente tabla.

Ítem	
Tipo de Crate	VME64x standard, con adaptador P3 back-plane
Tipo de tarjeta	9U
Slots entrada datos	4 (a 4 FIFOs de 3 Kwords)
Número de fibras ópticas entrada	8 (32 bits @ 40 MHz)
Número de LDCs conectadas	4 (conector doble)
Número de fibras ópticas salida	2 (32 bits @ 40 MHz)
Número de FPGAs (APEX 20K)	2 (BGA 252 pines)
Tipo de Alimentación	5 V – 3.3 V (tomados VME)

TABLA 2.3 Características generales del Módulo de Transición.

Por tanto, el Módulo de Transición presenta cuatro conectores para las tarjetas LDC, cuatro FIFOs para la lectura y almacenamiento de datos, dos FPGAs para el tratamiento de los mismos, una tarjeta ODIN integrada para la salida de datos, dos conversores electrónico-ópticos con salida a dos fibras ópticas, un conversor para obtener la alimentación de 3,3 V necesaria, dos conectores JTAG y dos EPROMs para la programación de las FPGAs, dos osciladores para generar los relojes de 40 MHz de la placa, un buffer para su distribución y dos conectores (J2 y J3) mediante los cuales se conecta a la crate.

En estas figuras podemos ver el diagrama de bloques del Módulo de Transición con el flujo que siguen los datos y una fotografía del mismo.

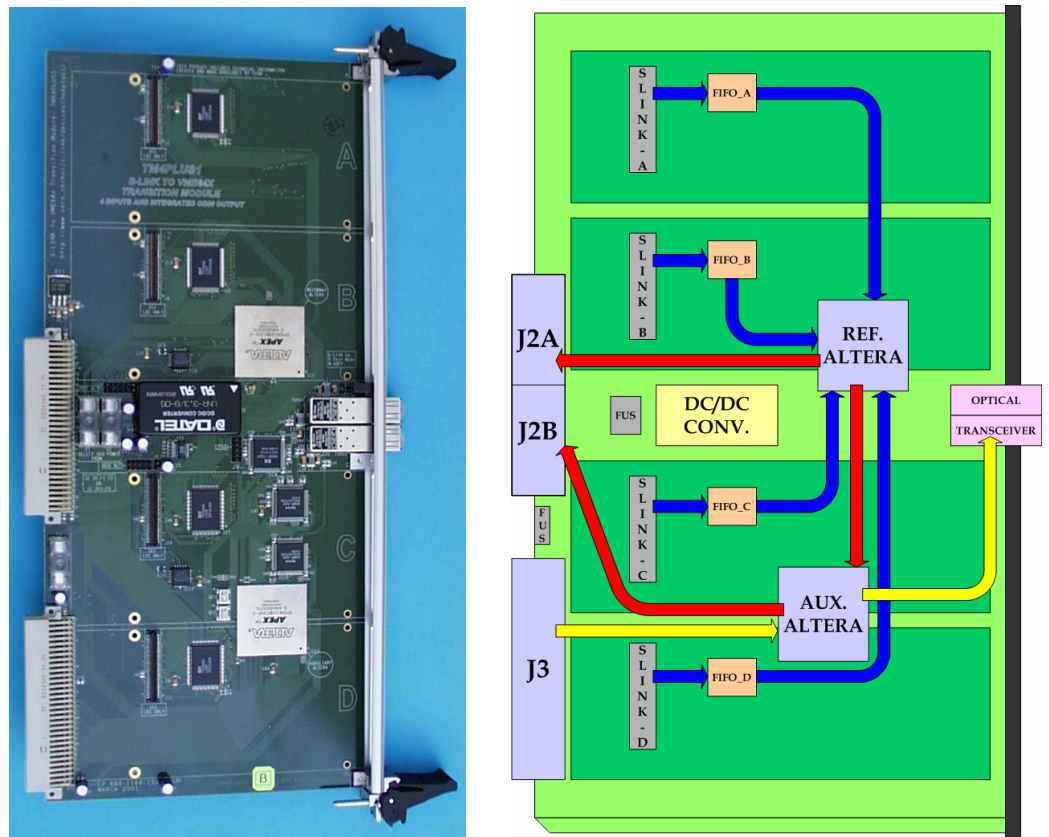


FIGURA 2.9 Fotografía y diagrama de bloques del Módulo de Transición.

Como se puede observar en el diagrama de bloques anterior, el flujo que siguen los datos (se explicará con mayor detalle en apartados posteriores) comienza por la recepción de los datos en las cuatro tarjetas LDC conectadas a los slots (SLINK A B C D) su posterior envío a las FIFOS (A B C D) y su rutado a la Reformatting Altera FPGA.

En dicha FPGA se reciben los 8 canales de datos de 32 bits a 40 MHz, se sincronizan los datos al reloj del módulo, se comprueba si el CRC es correcto, se realiza un primer filtrado (este caso dependerá del funcionamiento que se le esté dando al ROD) y se multiplexan los datos en dos bloques de 16 bits para ser enviados a las PUs de la motherboard. Este envío se realiza por los conectores J2A y J2B. Los datos de los slots A y B se envían directamente desde la Reformatting Altera FPGA y los datos de los slots C y D pasan primero por la Auxiliary Altera FPGA para ser rutados hacia J2B (por una cuestión de número de pines).

Una vez realizado el procesado en la motherboard, los datos son enviados al Módulo de Transición por J3 y de aquí a la Auxiliary Altera FPGA. Dichos datos ya viene con el formato del ROD en forma de palabras de 32 bits a 40 MHz, conteniendo la información de la energía el tiempo y la forma del pulso. La Auxiliary FPGA multiplexa estos datos en dos bloques de 16 bits a 40 MHz para enviarlos a los transmisores de la tarjeta ODIN integrada en el módulo que serán los encargados de enviarlos por fibra óptica al siguiente nivel de trigger (ROB).

En este otro diagrama de bloques se puede ver con mayor detalle todo el ciclo que siguen los datos desde su salida por fibra óptica del detector hasta su envío al ROB.

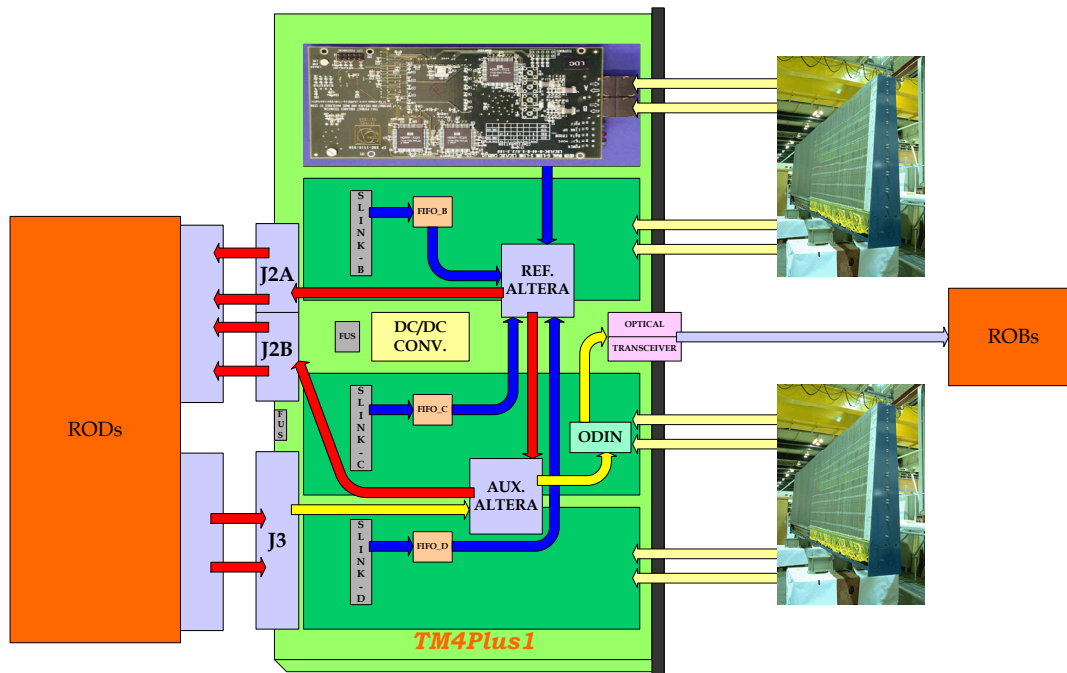


FIGURA 2.10 Flujo de los datos de TileCal.

Una vez tenemos estructurado y definido el sistema de adquisición de datos de TileCal sobre el cual hemos trabajado, vamos a ver en los siguientes apartados los protocolos usados para el envío y recepción de los datos, las implementaciones y características de las FPGAs y el formato de los datos que estudiamos.

3.2 INTERFACE S-LINK

El desarrollo de sistemas de comunicación donde trabajan un gran número de tarjetas requiere de protocolos de comunicación simples a la vez que muy fiables. Las conexiones de las fibras ópticas deben funcionar perfectamente, con el mínimo retardo posible y la mayor eficiencia. Para este fin, y después de estudiar distintos sistemas de comunicación, nació el protocolo S-LINK [10] (Simple LINK interface). Se usa para conectar cualquier capa de la electrónica del Front-End con la siguiente capa de la electrónica del Read-Out [11].

Nos ofrece unas especificaciones para el envío de los datos de un punto a otro, ofreciendo además:

- Una detección de errores (línea LDERR).
- Un control de los datos (línea UCTRL).
- La posibilidad de trabajar con diferentes tamaños de palabra (8, 16 y 32 bits).
- Un canal de retorno para controlar el flujo de los datos (línea UXOFF).
- Una función de autotest (línea UTEST) para comprobar si el enlace es correcto.
- Una función de reset para sincronizar correctamente las dos tarjetas y que el envío sea correcto (línea URESET).

En nuestro caso, se usa doblemente. Existe un primer enlace entre el Front-End y el Módulo de Transición y un segundo enlace entre Módulo de Transición y ROB. Para poder usar este interface se han diseñado en el CERN dos tipos de tarjetas, las Link Source Cards (LSC) y las Link Destination Cards (LDC) [12]. La primera de ellas encargada del envío de los datos y la segunda de la recepción de los mismos. Una imagen de este enlace es la siguiente figura.

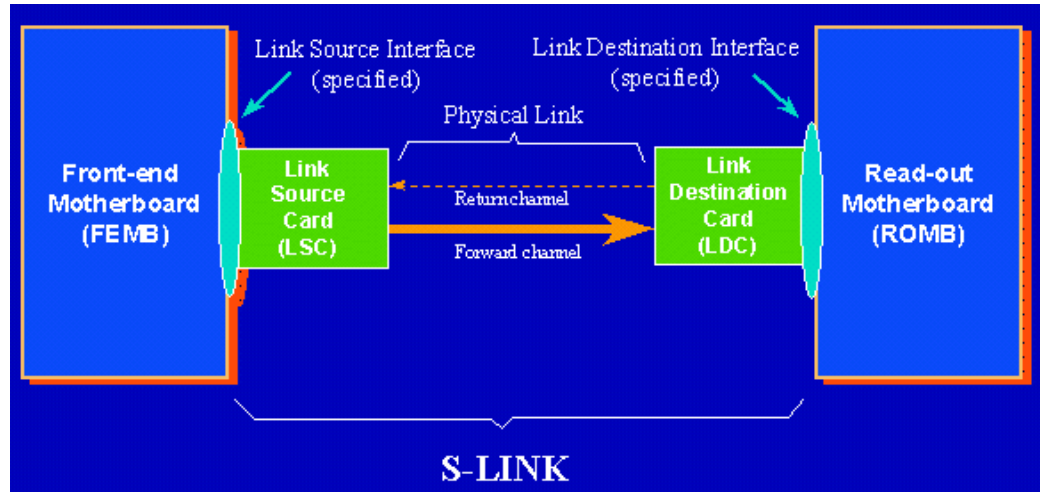


FIGURA 2.11 Protocolo de comunicación S-LINK.

Como acabamos de comentar, S-LINK necesita de dos componentes, las tarjetas LSC y las tarjetas LDC. Su comunicación se basa en las siguientes líneas, las cuales realizan las funciones definidas antes, durante y después de la recepción o envío de los datos.

Pin Symbol	Pin Name	I/O	Description
UD[31..0]	User Data input lines	Input to S-LINK	Data on these lines is transferred to the LSC on a low-to-high transition of UCLK when UWEN is low. UD[3..0] are ignored if UCTRL# is low. Synchronous with UCLK.
LFF#	Link Full Flag	Output from S-LINK	Data <i>shall</i> only be written to the S-LINK when this line is high. After it goes low, up to two more words <i>may</i> be written. Functional when S-LINK is in test mode. Undefined when URESET# is low. Synchronous with UCLK.
URESET#	User Reset line	Input to S-LINK	When low initiates a reset cycle. Asynchronous.
UTEST#	User Test line	Input to S-LINK	When low switches the LSC to test mode. Causes LDOWN# to go low. Asynchronous.
UDW[1..0]	User Data Width lines	Input to S-LINK	Define the data width the S-LINK is to be operated in. Data width codes at LSC and LDC must be the same. Sampled after a reset cycle.
UCTRL#	User Control line	Input to S-LINK	When low indicates that the data to be transmitted is a control word. Causes UD[3..0] to be ignored. Synchronous with UCLK.
UWEN#	User Write Enable	Input to S-LINK	When low enables data to be transferred to the S-LINK on the low-to-high transition of UCLK. Synchronous with UCLK.
UCLK	User Clock	Input to S-LINK	Data is transferred to the S-LINK on the low-to-high transitions of UCLK when UWEN# is low. This is a free running clock.
LRL[3..0]	Link Return Lines	Output from S-LINK	These lines reflect the state of URL[3..0] at the LDC. In the simplex version these lines are not functional. A simplex FEMB <i>shall</i> leave these lines unconnected. A simplex LSC <i>shall</i> connect these lines to GND. Asynchronous, even relative to each other.
LDOWN#	Link Down	Output from S-LINK	When low indicates that the S-LINK is not operational. Asynchronous. Can go low due to: - S-LINK failure; then LDOWN# is latched low until cleared by a reset cycle. - S-LINK undergoing reset cycle, then LDOWN# goes high when reset cycle is complete. - S-LINK in test mode, then LDOWN# goes high when test mode is ended. The FEMB <i>shall</i> pull LDOWN# low if the LSC is not present or not powered up.

TABLA 2.4 Descripción de las señales de una tarjeta LSC [10].

Para una tarjeta LDC las líneas que utiliza para su reconocimiento y funcionalidad son las siguientes.

Pin Symbol	Pin Name	I/O	Description
LD[31..0]	Link Data output lines	Output from S-LINK	Data present on these lines may be latched on the low-to-high transition of LCLK when LWEN# is low. LD[3..0] have special meanings when LCTRL# is low. Synchronous with LCLK.
UXOFF#	User Transmit Off	Input to S-LINK	When low, signals the LSC to stop transmitting data. Functions in test mode if UTDO# is low but does not stop transmission of test pattern if UTDO# is high. Asynchronous. In the simplex version this pin is not functional. A simplex ROMB <i>shall</i> pull-up this line to Vcc. A simplex LDC <i>shall</i> leave this line unconnected.
URESET#	User Reset line	Input to S-LINK	When low initiates a reset cycle. Asynchronous.
UTDO#	User Test Data Out	Input to S-LINK	When low, the received test pattern is transferred from the LDC to the ROMB during test mode. Sampled after a reset cycle.
UDW[1..0]	User Data Width lines	Input to S-LINK	Define the data width the S-LINK is to be operated in. Data width codes at LSC and LDC must be the same. Sampled after a reset cycle.
LCTRL#	Link Control line	Output from S-LINK	When low indicates that a control word is being transferred. Causes LD[3..0] to have a special meaning. Synchronous with LCLK.
LWEN#	Link Write Enable line	Output from S-LINK	When low indicates that valid data will be transferred to the ROMB on the low-to-high transition of LCLK. Synchronous to LCLK.
LCLK	Link Clock	Output from S-LINK	Data is transferred to the ROMB on each low-to-high transition of LCLK when LWEN# is low. This is a free-running clock if LDOWN# is high. If LDOWN# is low, this signal is undefined.
LDERR#	Link Data Error	Output from S-LINK	When low indicates that a data transmission error has occurred or that a pattern error has occurred during test mode. With word-by-word error reporting, LDERR# goes low with the word in error and with the control word for that block. With block basis error reporting, LDERR# goes low only with the control word for the block containing the word with error.
URL[3..0]	User Return Lines	Input to S-LINK	The state of these lines is sampled, transmitted back to the LSC and presented on LRL[3..0]. Asynchronous, even relative to each other. In the simplex version these pins are not functional. A simplex ROMB <i>shall</i> connect these lines to GND. A simplex LDC <i>shall</i> leave these lines unconnected.
LDOWN#	Link Down	Output from S-LINK	When low indicates that the S-LINK is not operational. Asynchronous. Can go low due to: - S-LINK failure; then LDOWN# is latched low until cleared by a reset cycle. - S-LINK undergoing reset cycle, then LDOWN# goes high when reset cycle is complete. - S-LINK in test mode, then LDOWN# goes high when test mode is ended. The ROMB <i>shall</i> pull LDOWN# low if the LDC is not present or not powered up.

TABLA 2.5 Descripción de las señales de una tarjeta LDC [10].

Una vez definido el enlace entre el detector y el Módulo de Transición, los datos usando este interface se almacenan en las FIFOs presentes en el módulo. El siguiente paso es la lectura de esas FIFOs mediante la Reformatting Altera FPGA y su adecuación a nuestras especificaciones.

3.3 DISPOSITIVOS LÓGICOS PROGRAMABLES

Dentro del Módulo de Transición podemos decir que los elementos principales del mismo son dos FPGAs. Gracias a ellas hemos podido diseñar un sistema compacto para la recepción, tratamiento, distribución y salida de los datos.

Las FPGAs (Field Programmable Gate Arrays) son una clase de Dispositivo Lógico Programable [13], es decir, dispositivos digitales que contiene estructuras lógicas compuestas de puertas lógicas, recursos de interconexión programables, estructuras de entrada/salida flexibles y estructuras de memoria interna.

Permiten al usuario configurar la implementación hardware de una determinada máquina algorítmica definida en un lenguaje de descripción de alto nivel, que va a ser sintetizado en función de la tecnología y el dispositivo en cuestión. Se ha usado para

nuestro sistema dos FPGAs de Altera, el software QUARTUS II [14] y el lenguaje VHDL. La siguiente figura muestra la estructura genérica de una FPGA.

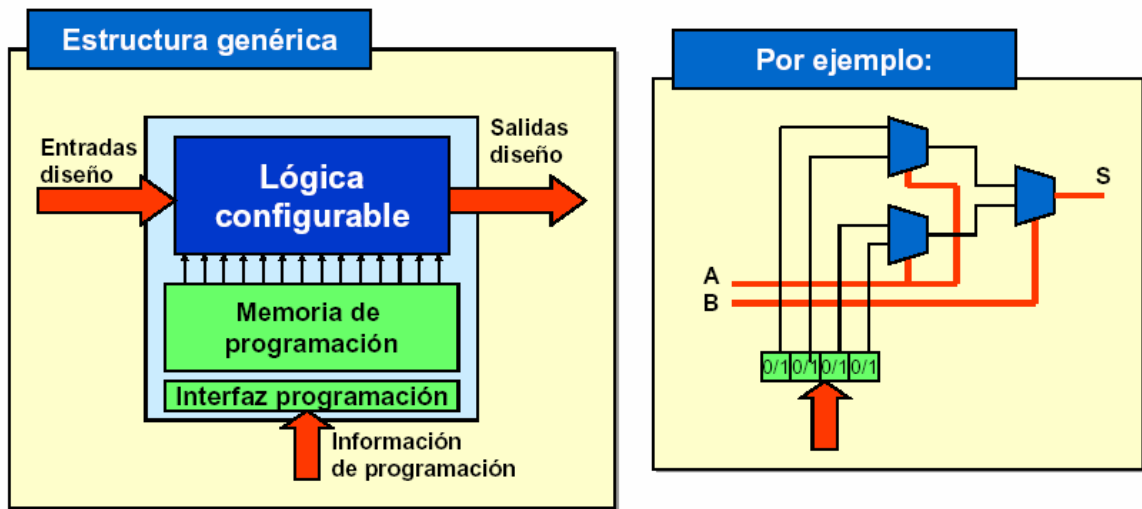


FIGURA 2.12 Estructura genérica de las FPGAs.

Las FPGAs pueden considerarse como una evolución de los Dispositivos Lógicos Programables (PLDs), aunque el incremento de complejidad ha supuesto cambios estructurales. Una característica importante de estos sistemas, que ayudó a su inserción en el Módulo de Transición, es la autonomía del diseñador respecto al sistema en sí. Permite una simulación completa del dispositivo sin necesidad de trabajar directamente en él.

Respecto a la evolución tecnológica de los últimos años, los dispositivos reconfigurables han ido progresivamente comiendo terreno a soluciones tipo ASIC (Application Specific Integrated Circuit), predominantes en la década anterior. Ello ha sido debido a su mejor adaptación a la constante evolución tecnológica.

Recordemos, en cualquier caso, que la diferencia fundamental entre las soluciones de implementación ASIC y FPGA reside en el nivel de personalización de la aplicación. Mientras que al nivel de ASIC el diseñador puede interaccionar con diferentes etapas del proceso tecnológico al nivel de FPGA únicamente se configura el sistema, es decir, se asignan los valores a las funciones lógicas, las interconexiones activas entre dispositivos y la funcionalidad de las celdas de entrada/salida del circuito integrado.

Como consecuencia, todo el proceso de concepción, desde la descripción hasta la implementación hardware puede ser realizado en “casa” del diseñador (“Field Programmable”) evitándose tiempos de espera debidos a procesos de fabricación. Esto tiene una segunda lectura: cualquier rediseño requerido por nuevas especificaciones o por detección de errores no deseados se puede realizar de una forma muy rápida al evitar la interacción con el fabricante. La clave está en saber escoger un dispositivo que se adecue a nuestras especificaciones tanto actuales como futuras.

Un ejemplo del interior de una FPGA puede observarse en el siguiente diagrama de bloques.

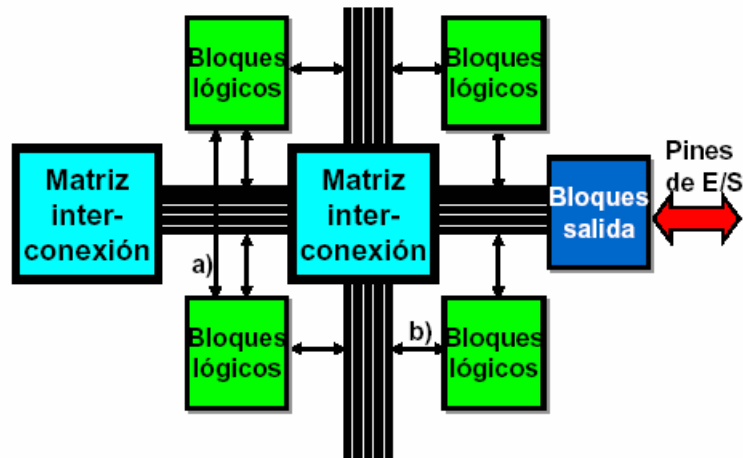


FIGURA 2.13 Estructura interna de las FPGAs.

En contrapartida, las prestaciones equivalentes en superficie de silicio y tecnología utilizada de los dispositivos reconfigurables son menores que las de los dispositivos ASIC. Esta relación suele cifrarse en un factor 5 respecto a densidad de integración (medido como número de puertas por unidad de superficie) y en un factor 3 en términos de velocidad (medido en velocidad máxima de reloj).

Uno de los argumentos del auge de las FPGAs es la sencillez de las herramientas, aunque los procesos involucrados en el diseño de las mismas son muy similares a los correspondientes al diseño con ASICs. Para la programación de las FPGAs del Módulo de Transición se usó el Software QUARTUS II de Altera, tal y como habíamos comentado anteriormente, mediante el cual se realiza tanto la Síntesis de Alto Nivel (comportamental y lógico) como la Síntesis de Nivel Físico (floorplan, placement y routing).

Las FPGAs que contiene el módulo son de la familia APEX 20K de Altera, a continuación veremos sus características fundamentales.

Se trata del primer dispositivo lógico programable que incorpora la integración de un SOPC (System On a Programmable Chip) [14]. Presenta una arquitectura MultiCore integrada por lógica Look Up Table (LUT), lógica producto-término y memoria embebida. Su nombre viene de Advance Programmable Element matrix (APEX) usando tecnología SRAM (2,5 V y 0,25/0,2 μ).

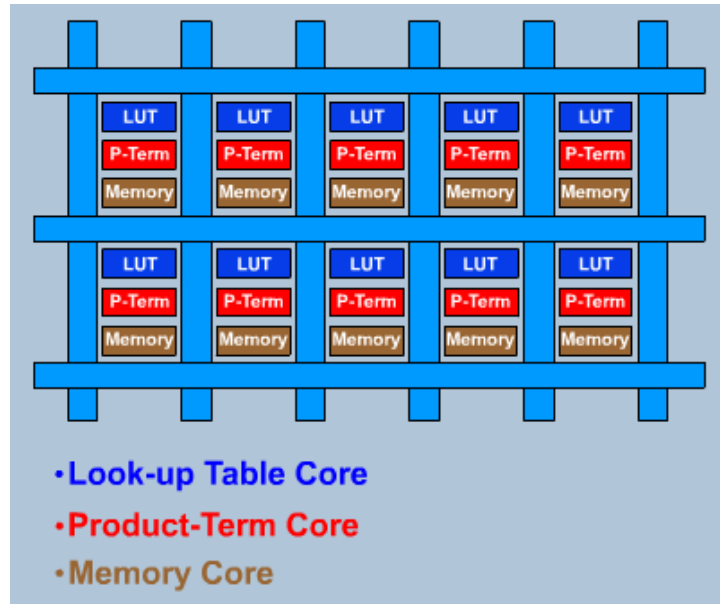


FIGURA 2.14 Arquitectura principal de la APEX 20K.

El bloque computacional básico de las FPGAs esta formado por Elementos Lógicos (LE), Logic Array Blocks (LAB) que tienen un total de 10 LEs, Embedded System Block (ESB) que son memorias o bloques lógicos, MegaLABs formadas por n-LEs + ESB, FastTrack o interconexiones locales, Entradas/Salidas y Relojes específicos.

Las MegaLABs se conectan con interconexiones Fila o Columna y los elementos de E/S se sitúan al final de las filas o las columnas. La siguiente figura muestra esta disposición.

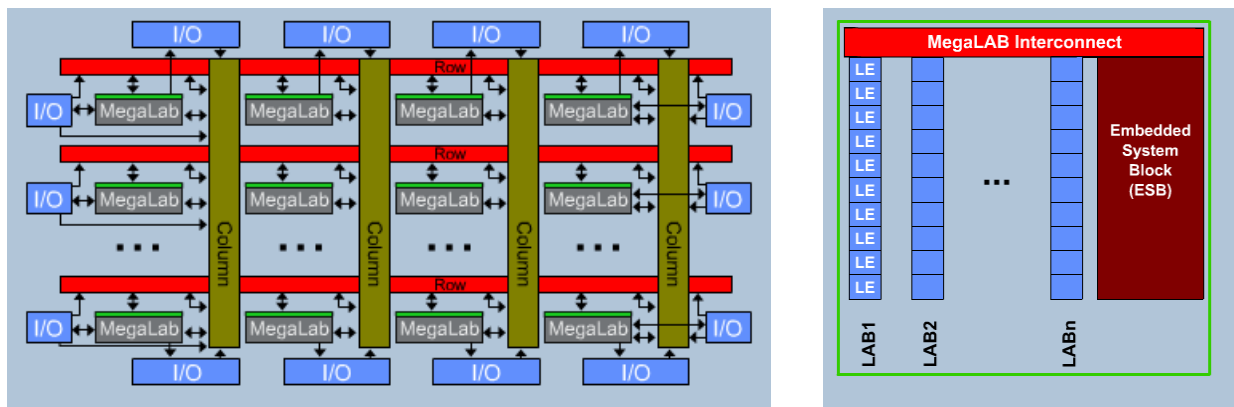


FIGURA 2.15 Interior de una APEX FPGA.

3.4 METODOLOGÍA DISEÑO FPGAs

La metodología de diseño top-down usada en esta Tesis se puede resumir en estos pasos:

- Diagrama de bloques. Primera idea mental de los bloques que se desean implementar.
- Captura del diseño. Realización del diagrama de bloques en la aplicación del fabricante, introducción de la programación en VHDL.
- Compilación. Comprobación de la no existencia de errores funcionales.
- Análisis de tiempos. Confirmación de que los requerimientos de tiempo se cumplen sin presentar retardos.
- Simulación. Creación de los vectores de test y verificación de tener una salida correcta.
- Programación. Implementación de nuestro diseño en la FPGA.

Para la realización de todos estos pasos se ha usado la herramienta QUARTUS II que nos ofrece el mismo fabricante de las FPGAs. Se trata de un software de desarrollo que nos ofrece un entorno de diseño completo para los SOPC. Tiene una total integración con múltiples herramientas EDA (Electronic Design Automation).

En los siguientes diagramas de bloques se puede observar el camino que se debe seguir para una implementación completa del diseño.

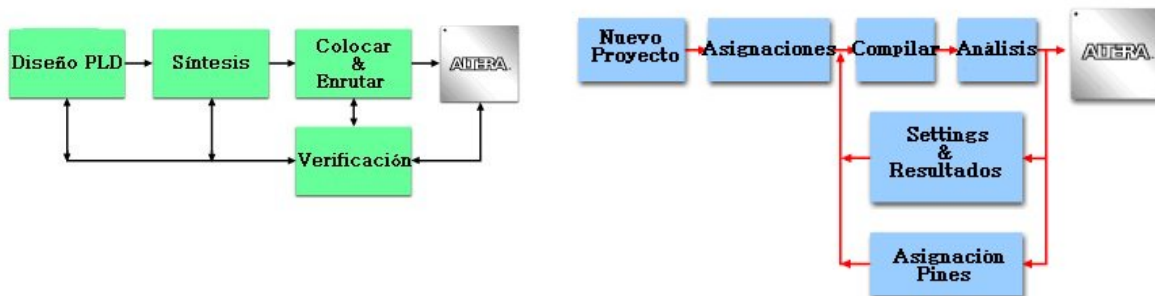


FIGURA 2.16 Metodología de desarrollo.

3.5 REFORMATTING ALTERA FPGA

Siguiendo la metodología de diseño expuesta en el apartado anterior, la implementación de nuestras especificaciones de trabajo [15] para un entorno de funcionamiento hardware en tiempo real se basa en el siguiente diagrama de bloques. Será a partir de él desde donde explicaremos bloque a bloque el funcionamiento del sistema.

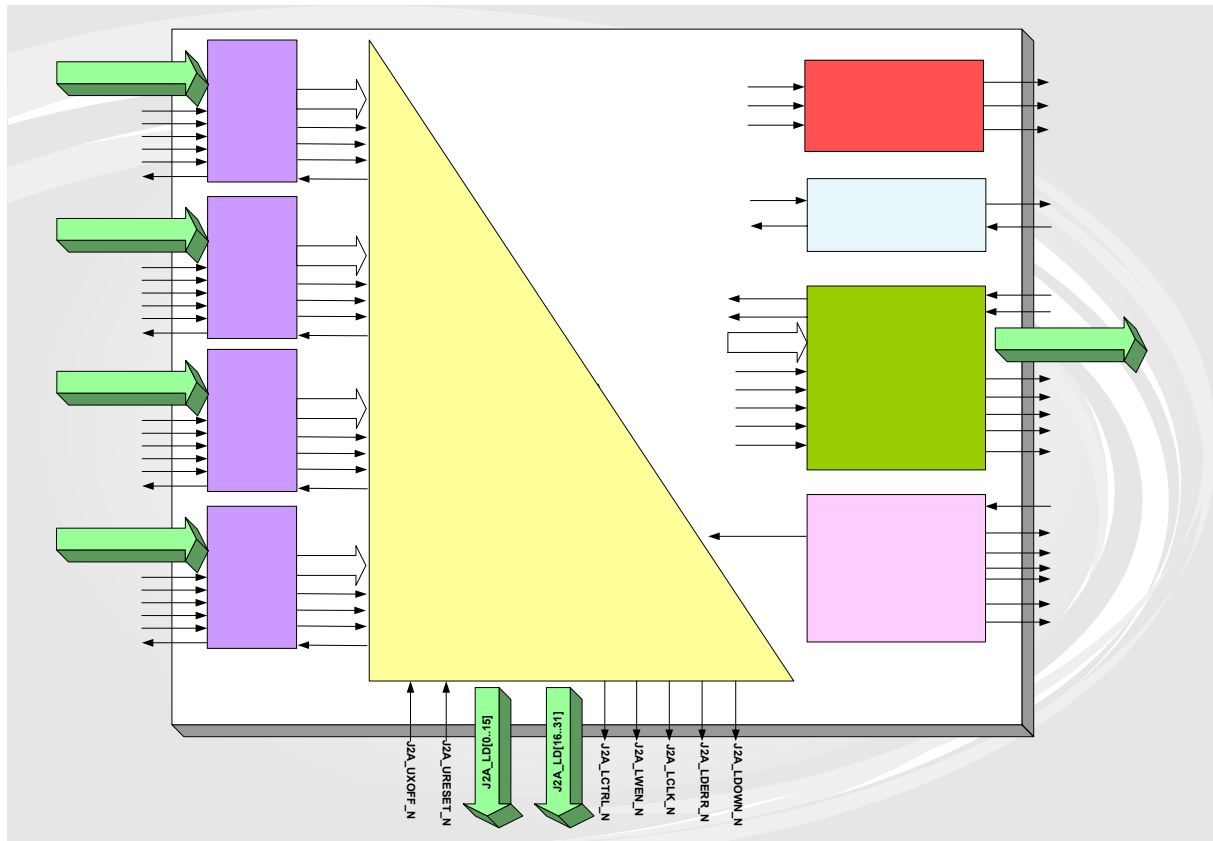


FIGURA 2.17 Diagrama de bloques Reformatting Altera FPGA.

3.5.1 Bloque FIFO

En estos bloques se realiza la lectura de los datos que provienen del front-end y que mediante el protocolo S-LINK pasan al Módulo de Transición y se almacenan temporalmente en las FIFOs que incorpora el mismo. Se realiza la lectura de las cuatro tarjetas LDC conectadas y de los ocho canales de datos de entrada. Se trata de cuatro IDT 72V3660 FIFOs [16] que presentan la siguiente estructura interna.

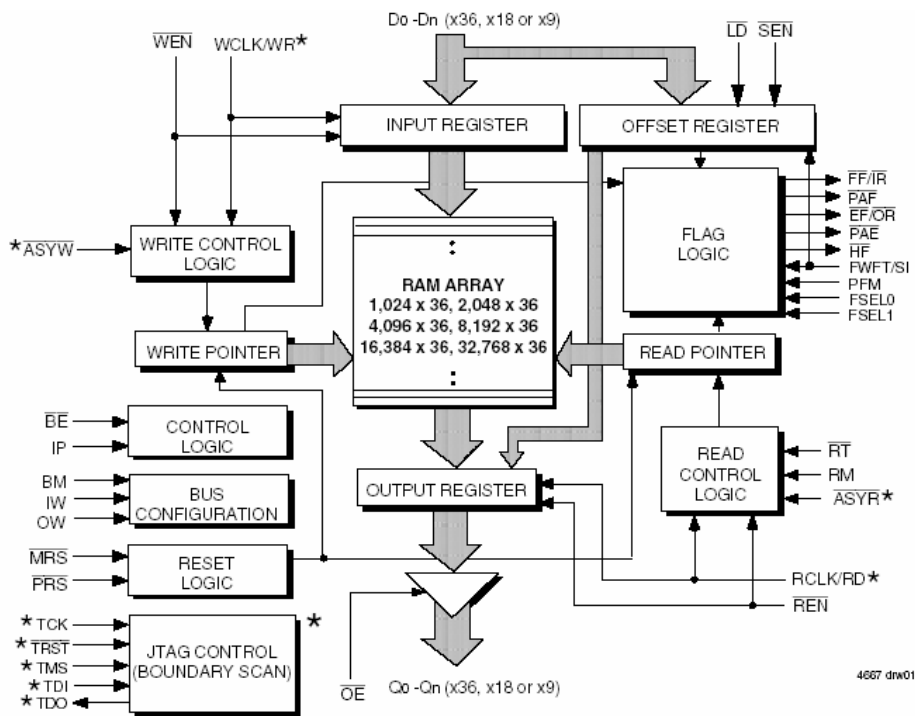


FIGURA 2.18 Diagrama de bloques interior FIFO.

Para la lectura de los datos (AF_LD[31..0]), se envía una señal READEN que permite el inicio de la toma de datos. Como se puede observar en el diagrama de bloques, en este bloque se realiza la lectura de las señales de control de S-LINK (LCTRL, LDERR, UXOFF) y las señales de control de la FIFO (EMPTY, FULL). Los datos están formados por palabras de 32 bits @ 40 MHz que entran en paralelo en la FPGA.

Este bloque se encarga por tanto de sincronizar dichos datos a la frecuencia de reloj del módulo, permitiendo así un flujo de datos continuo y sin retardos. En caso de que la FIFO esté vacía, el proceso de lectura se realiza (envío de la señal READEN) hasta que se produzca un flag de FIFO llena y se pare el proceso. Las señales de S-LINK se envían al bloque central para su posterior tratamiento.

3.5.2 Bloque connect FIFO

Encargado de inicializar el modo de operación de las FIFOs, necesario para seleccionar el IDT Standard Mode Timing (modo de operación definido por las FIFOs donde sólo se envía el dato cuando mandamos una señal de lectura) [16]. Mediante la señal MRESET se realiza un master reset que permite inicializar las funciones de la FIFO. La señal FWFT selecciona el modo de operación comentado anteriormente y la señal BUSM la anchura del bus de datos a transmitir.

Las siguientes figuras muestran los timings de reset, lectura y escritura.

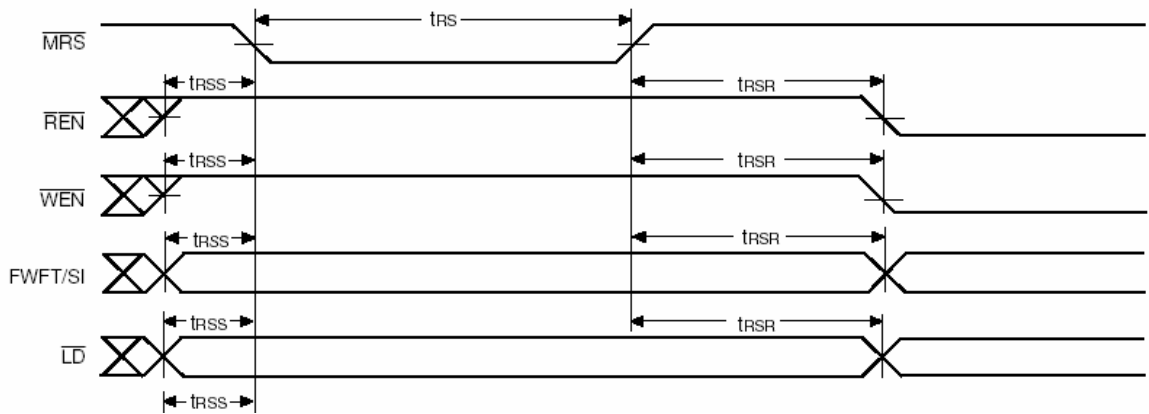


FIGURA 2.19 Timing Master Reset.

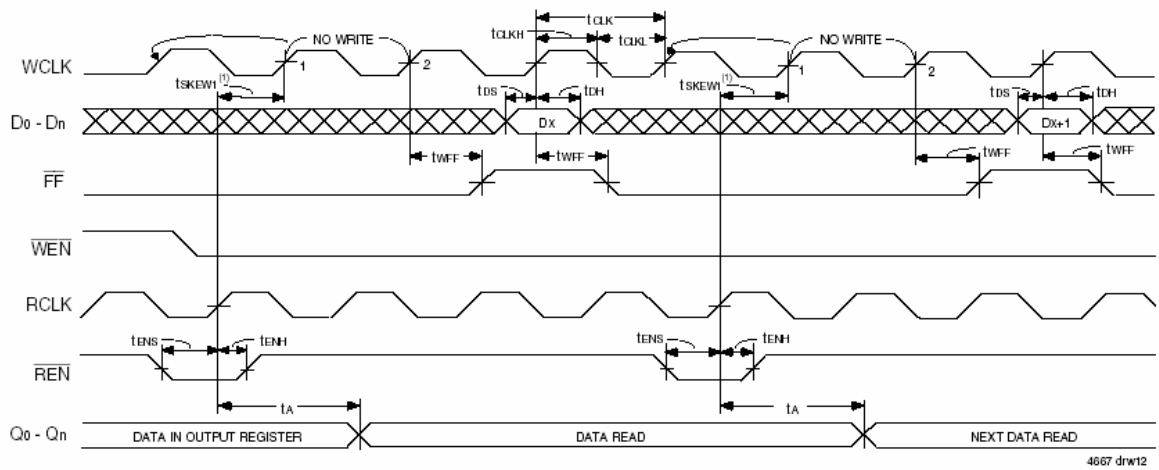


FIGURA 2.20 Write Cycle Timing.

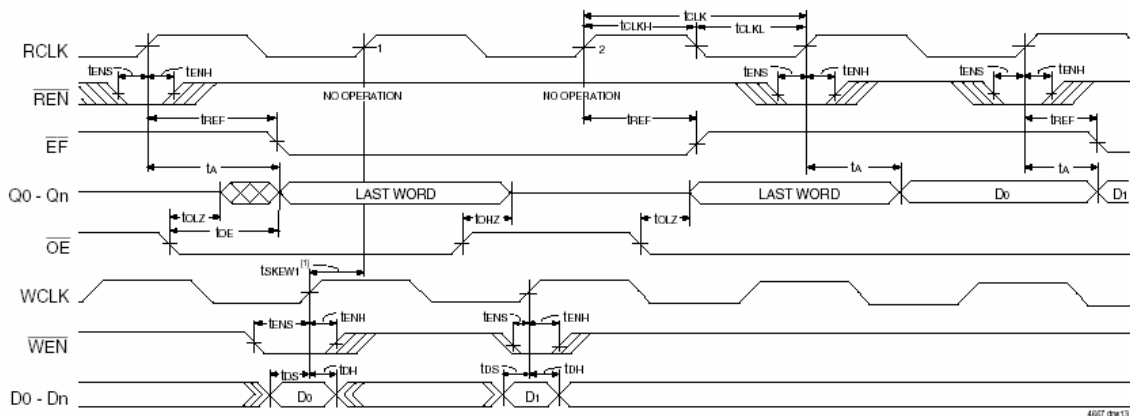


FIGURA 2.21 Read Cycle Timing.

La siguiente tabla muestra todas las líneas implicadas en el proceso de lectura/escritura de la FIFO [9] y su descripción.

Name	Symbol (Schematics)	Symbol (FIFO)	I/O, Pins	Description
Data Inputs	A_LD[0..31]	D	I-34	From the S-LINK (AF_LD, AF_LCTRL#, AF_LDERR#)
Data in (pins 18, 36)		D	I-2	Unused data input pins (total inputs: 36; used: 34)
Master Reset	F_MRESET#	MRS#	I-1	The only reset feature implemented
Partial Reset		PRS#	I-1	Disabled: only master reset is used
Retransmit		RT#	I-1	Useful only if data needs to be read again but not sent again (no use in the design)
First Word Fall Through/ Serial In	F_FWFT#	FWFT/SI	I-1	During master reset, selects First Word Fall Through or IDT Standard mode.
Output width		OW	I-1	Selects the bus width of the read port (18 or 36-bit)
Input Width		IW	I-1	Selects the bus width of the write port (fixed to 36-bit)
Bus-matching	F_BUSM#	BM#	I-1	Selects the bus width of the read/write port
Big Endian/ Little Endian		BE#	I-1	Always set to Little-Endian format
Retransmit timing mode		RM	I-1	Set to Normal Latency Mode
Programmable Flag Mode		PFM	I-1	Set to Synchronous Programmable Flag Timing Mode
Interspersed Parity		IP	I-1	Set to Non- Interspersed Parity Mode
Flag Select Bit0		FSEL0	I-1	During master reset, selects (with FSEL1 and LOAD) the default offset values for PAF (fixed to 1k)
Flag Select Bit1		FSEL1	I-1	During masterresesses (with FSEL0 and LOAD) the default offset values for PAF (fixed to 1k)
Write Clock	A_LCLK	WCLK	I-1	Clock from the S-LINK (LCLK); write clock for the FIFO
Write Enable	A_LWEN#	WEN#	I-1	Signal from the S-LINK (LWEN)
Read Clock	BUFFERCLK_A	RCLK	I-1	Clock from the Buffer; read clock for the FIFO
Read Enable	AF_READEN#	REN#	I-1	Signal from the Altera (READEN)
Output Enable		OE#	I-1	Controls the output impedance of Q
Serial Enable		SEN#	I-1	Parallel loading of programmable flag offsets. Feature not available since it works on WCLK, not available in the Altera.
Load	LOAD#	LD#	I-1	During master reset, selects (with FSEL0 and FSEL1) the default offset values for PAF (fixed to 1k)
Full Flag/ Input Ready	AF_FULL#	FF/IR#	O-1	Indicates whether or not the FIFO memory is full
Empty Flag/ Output Ready	AF_EMPTY#	EF/OR#	O-1	Indicates whether or not the FIFO memory is empty
Programmable Almost-Full Flag	A_UXOFF#	PAF#	O-1	Goes HIGH if the number of free locations in the FIFO memory is more than offset m (LOW if less or equal to m); m=1024
Programmable Almost-Empty Flag		PAE#	O-1	LOW if the number of words in the FIFO memory is less than offset n (HIGH if more or equal). Not used
Half-Full Flag		HF#	O-1	Not used
Data Output	AF_LD[0..31]	Q	O-34	To the Altera (AF_LD, AF_LCTRL#, AF_LDERR#)
Data out, pins 18, 36		Q	O-2	Unused data output pins (total inputs: 36; used: 34)

TABLA 2.6 Descripción de las señales de en las FIFOs.

3.5.3 Bloque control S-LINK

Mediante la línea URESET conseguimos inicializar la máquina de estados de la tarjeta LDC conectada al módulo. A su vez, y mediante la línea LDOWN, sabemos si el enlace es correcto y si vamos a tener datos de entrada. Estas líneas cumplen el protocolo S-LINK comentado con anterioridad, no pasan por las FIFOs sino que van directas a la tarjeta LDC.

El conjunto de todas las líneas [9] de estos tres bloques se puede observar con claridad en la siguiente figura.

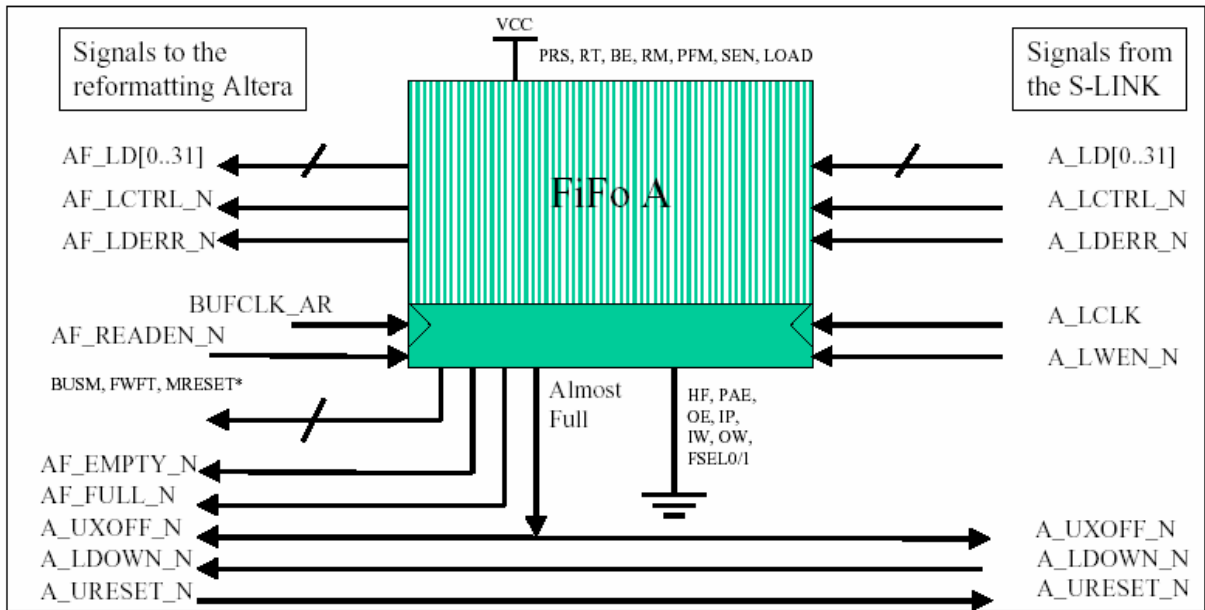


FIGURA 2.22 Señales entrada/salida LDC.

3.5.4 Bloque data multiplexing, control & filtering unit

Nos encontramos ante el bloque principal del diseño de esta FPGA; una vez los datos han entrado a la FPGA y han sido sincronizados, es aquí donde se realiza un control del CRC (Cyclic Redundancy Checks), el Flat Filtering y el Multiplexado de los datos para su posterior envío hacia la motherboard del ROD (conector J2A) o hacia la Auxiliary Altera FPGA que los enviará luego a la motherboard (conector J2B).

Lo primero que se produce es la comprobación del CRC, esto significa que del formato de datos de entrada tenemos una palabra de 32 bits que incluye un código representativo de cada TileDMU (Tile Data Management Unit) que debe coincidir con la que se genera internamente en la FPGA. Si se detecta cualquier error, se debe enviar un flag CRC ERROR WORD.

En el Anexo III están desarrollados en Código VHDL todos los algoritmos de programación vistos hasta ahora, así como todo el desarrollo del proyecto.

El formato de datos de TileCal [8] viene determinado por la siguiente figura.

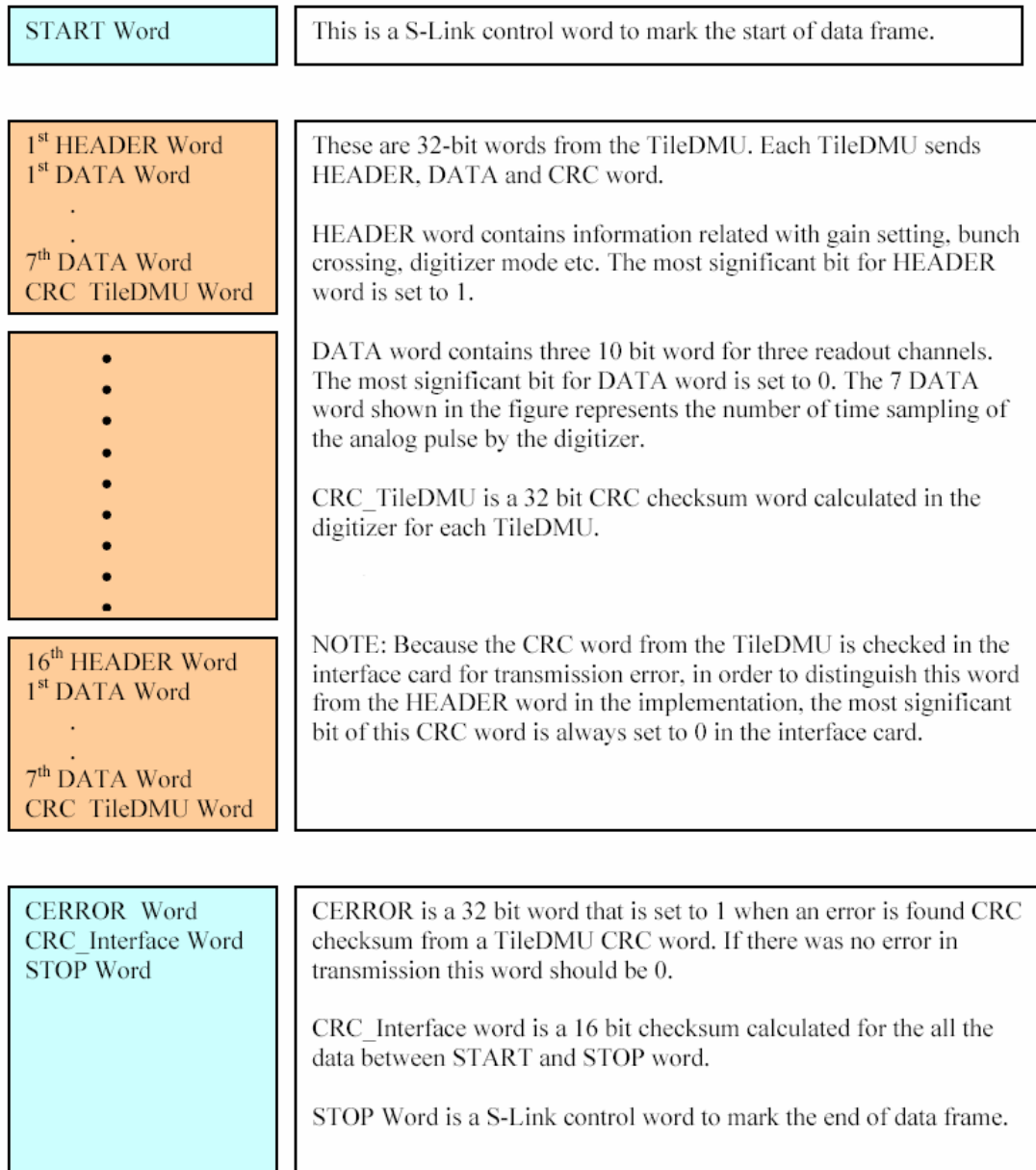


FIGURA 2.23 Formato de datos entrada TileCal.

Una vez comprobado el CRC, los datos son filtrados mediante el algoritmo Flat Filtering, una versión más sencilla del algoritmo Optimal Filtering que se implementa en las PUs del ROD.

Cada palabra de 32 bits de datos contiene tres canales de datos que debemos separar para sumar a las siete palabras. Con esos datos sumados tendremos la energía depositada en cada celda del detector, en total 3 celdas por cada evento recibido.

Las siguientes figuras representan el formato de un evento de TileCal y el formato de cada una de las palabras de datos.

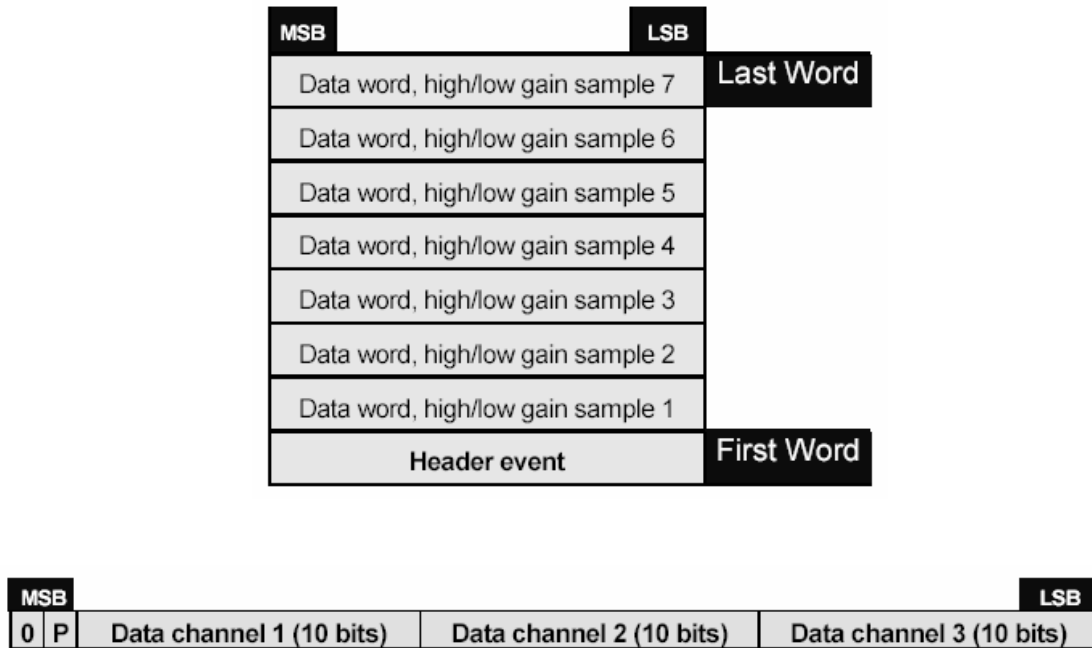


FIGURA 2.24 Formato de un fragmento de un evento y de tres canales de datos [12].

El algoritmo Flat Filtering sólo calcula la energía depositada, no tiene en cuenta el tiempo ni la forma del pulso. Para ello, antes del cálculo debemos normalizar los datos recibidos eliminando el pedestal mediante una sencilla fórmula. Con esos datos y con una constante de calibración (K_{FF}) obtendremos el valor de la energía.

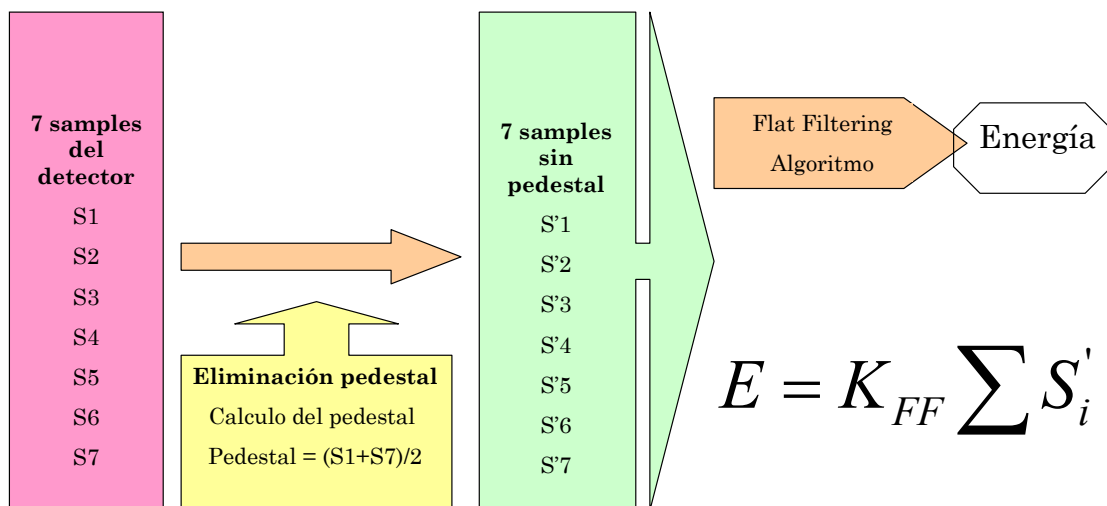


FIGURA 2.25 Algoritmo Flat Filtering.

Como paso final en el tratamiento de la información enviada por el detector debemos multiplexar los datos que recibimos en formato de 32 bits en dos palabras de 16 bits. Esto se debe a las especificaciones de entrada de las PUs del ROD.

El proceso de multiplexado se basa en el siguiente diagrama de bloques.

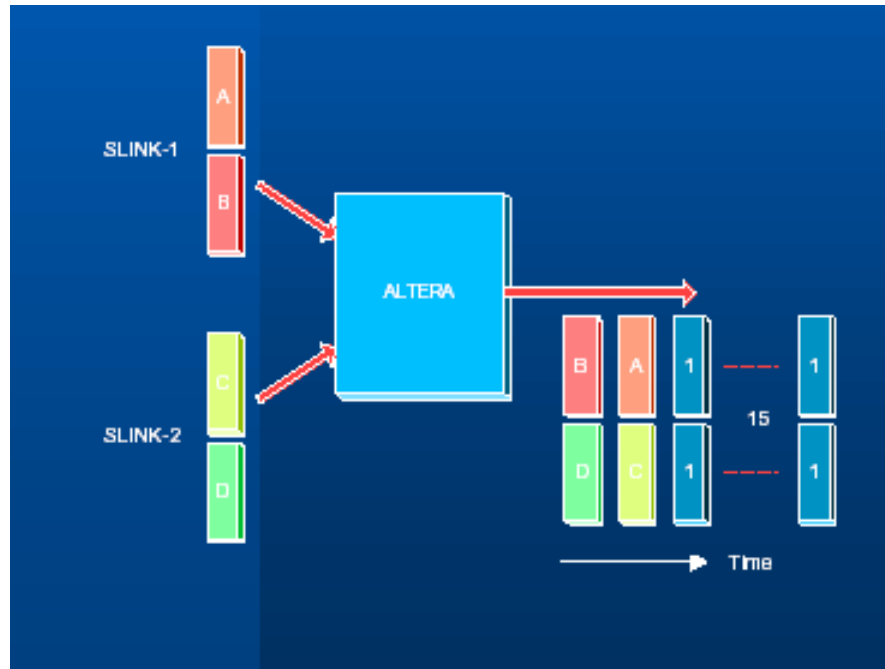


FIGURA 2.26 Multiplexado de datos.

Se puede observar como los 32 bits que se reciben en paralelo en la FPGA de cada uno de los conectores S-LINK, se convierten en bloques de 16 bits que se mandan sincronizados hacia la motherboard del ROD.

Una característica fundamental en el desarrollo del Módulo de Transición es la posibilidad de trabajar de forma independiente a la motherboard. Este modo de funcionamiento permite realizar todo el proceso en las FPGAs y enviar los datos directamente al siguiente nivel de trigger sin pasar por el proceso de la motherboard.

No se trata del modo correcto de funcionamiento, pero si de una forma real de trabajo que en caso de necesidades extremas siempre tendremos en consideración. Este modo ha sido usado dentro del proceso de test que se comentará en el siguiente capítulo en situaciones de fallo de la motherboard.

3.5.5 Bloque auxiliary FPGA

Es en este bloque donde se realiza el rutado de los datos recibidos en los conectores C y D del módulo hacia la Auxiliary Altera FPGA. Así mismo, nos puede servir para enviar todos los datos a dicha FPGA en el caso de estar funcionando de forma independiente a la motherboard.

Se trata de una conexión de gran utilidad, tanto para el test del módulo como para su funcionamiento real. No sólo presenta líneas de datos, también tenemos las líneas de control del protocolo S-LINK; pudiendo mantener la escalabilidad y determinación de errores tanto en funcionamiento como en desarrollo del proceso.

3.6.2 Bloque reformatting FPGA

Bloque encargado de la recepción de los datos que provienen de la Reformatting Altera FPGA. Capaz de generar señales de control y error en caso de fallo en la transmisión. Según el modo de funcionamiento en el que nos encontremos recibirá datos de los cuatro enlaces ópticos o de sólo dos de ellos.

3.6.3 Bloque J2B connection

Conexión directa de esta FPGA con la motherboard del ROD. Dicho enlace se produce por el conector J2B, enviando los datos en dos bloques de 16 bits hacia las PUs. No se realiza ninguna función de multiplexado, al haberse realizado previamente en la Reformatting Altera FPGA.

3.6.4 Bloque data control unit

Este bloque central realiza el rutado y la sincronización de los datos que se reciben en esta FPGA, tanto desde la motherboard como desde la otra FPGA. A su vez, se encarga de actuar sobre el protocolo S-LINK y de controlar todo el sistema.

3.6.5 Bloque auxiliary FPGA clock, control & test

Realiza la función de recepción y rutado del reloj, de control y test de las líneas dedicadas a tal fin y de encender los LEDs que indican buena comunicación con el siguiente nivel de trigger, fallo en la comunicación o modo test.

3.6.6 Bloque ODIN LSC (G-LINK)

Bloque dedicado a implementar las funciones de una tarjeta física ODIN LSC [17] basada en el protocolo S-LINK pero con chipset G-LINK de bajo consumo.

La tarjeta ODIN S-LINK es una tarjeta estándar con doble canal de salida de datos, en nuestro caso, se trata de una tarjeta integrada por completo en el Módulo de Transición y sólo testada por nosotros. Se realizó el diseño de esta manera por la necesidad de tener cuatro conectores para la entrada y no poder colocar tarjeta alguna para la salida.

Usa el chipset G-LINK HDMP-1032/34 y trabaja a una frecuencia de 40 MHz. Presenta una tasa de transferencia máxima de 160 Mbytes/s. Este dispositivo necesita de un oscilador en la placa y de un reloj de referencia para enganchar el flujo de datos que se producen.

En este apartado describiremos cómo se mapean los 32 bits que se reciben desde la motherboard y cómo se usan los comandos internos para la comunicación.

El protocolo G-LINK está definido en base al envío de palabras para encontrar la comunicación correcta entre el emisor y el receptor. Los 32 bits de entrada se almacenan en una FIFO de entrada cada flanco positivo del reloj UCLK. La lógica de rutado lee los datos en cada flanco positivo del reloj XCLK para ser comprobados los bits de paridad.

El siguiente paso es multiplexar los datos recibidos en bloques de 16 bits (del mismo modo que se hacía en la Reformatting Altera FPGA para los datos que se recibían y se enviaban hacia la motherboard).

Una vez realizados todos esos pasos, se codifican los datos con un CRC para ser enviados a través de los transmisores al siguiente nivel de procesado de datos.

El protocolo que se ha diseñado en QUARTUS II e implementado en esta FPGA sigue el siguiente diagrama de bloques [18].

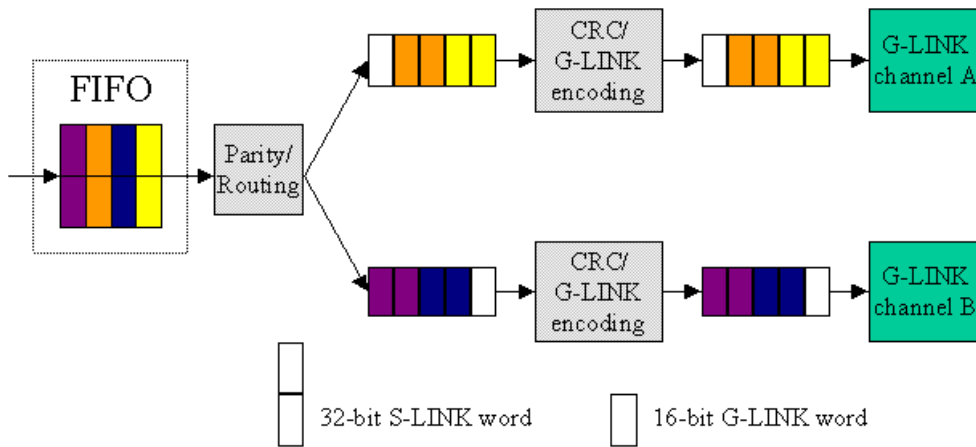


FIGURA 2.28 Diagrama de bloques del protocolo G-LINK.

Como se ha comentado, la comunicación se realiza gracias a un primer protocolo de reset que se realiza entre emisor y receptor, dicho protocolo nos sirve para encontrar errores, si los hay, entre ambos elementos. La máquina de estados de reset se puede ver en la figura 2.29. Así mismo, podemos observar cómo se produce el cambio de estados que definen correctamente el protocolo que hemos definido.

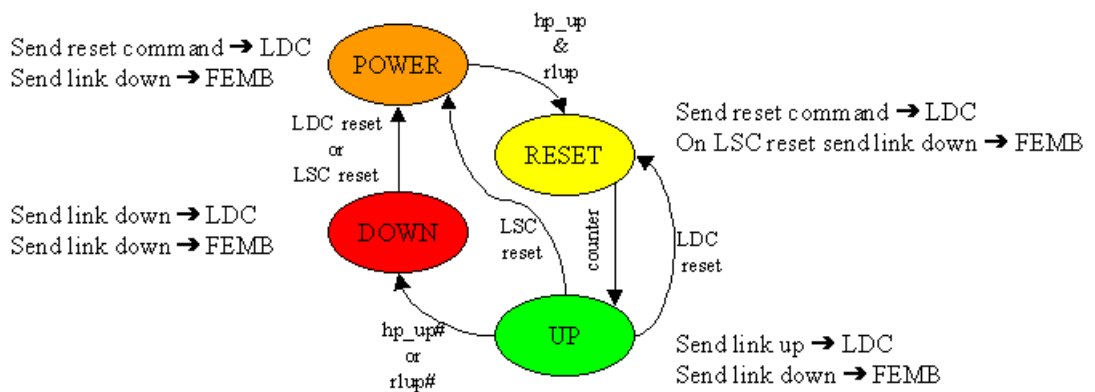


FIGURA 2.29 Máquina de estados del funcionamiento de G-LINK.

Cuando conectamos la alimentación, inmediatamente nos encontramos en el estado POWER. Si las fibras no estuvieran bien conectadas o algún reloj fallara y no se pudieran enganchar el transmisor y el receptor, pasaríamos al estado DOWN.

Si todo es correcto, se envía un comando de reset que comprueba que el enlace se está produciendo satisfactoriamente. En ese momento, pasamos al estado UP en el cual se permanece hasta el final de la comunicación o al aparecer un error.

La secuencia correcta de lo que acabamos de explicar viene definida en la figura 2.30.

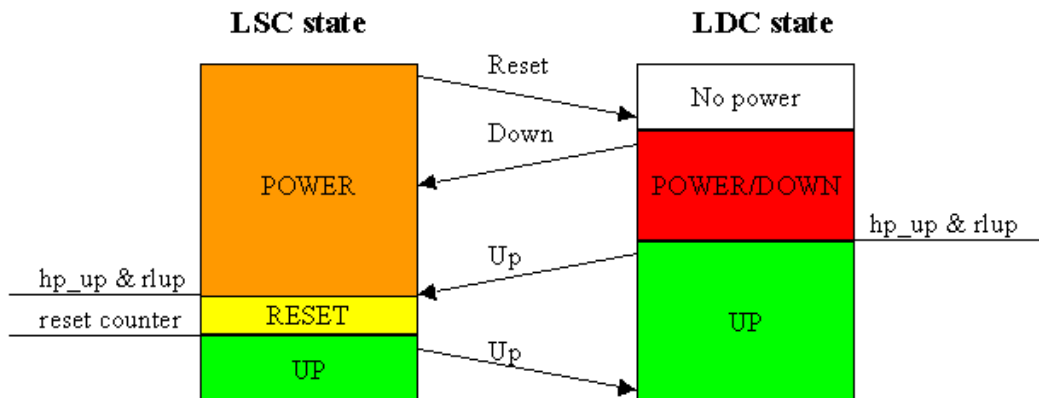


FIGURA 2.30 Secuencia POWER-UP de funcionamiento.

Con este bloque quedan definidas todas las operaciones que se realizan internamente en la Auxiliary Altera FPGA. El siguiente apartado nos definirá y mostrará el formato de los datos que se tratan en la salida del Módulo de Transición.

4. FORMATO DE DATOS DEL ROD

El formato de datos que presenta el ROD de TileCal [19] viene determinado por múltiples discusiones con la comunidad del detector. Se han tenido en cuenta consideraciones para decrementar al máximo el ancho de banda de entrada y salida de los enlaces ópticos, siempre teniendo en mente perder la mínima cantidad de información y la máxima flexibilidad para futuros cambios en la estructura hardware.

Una vez descrita la forma con la cual se reciben y se envían los datos en el ROD, vamos a establecer en este apartado el formato exacto que le daremos internamente a dichos datos y el cuál deberá ser interpretado correctamente por el siguiente nivel de adquisición.

4.1 ESTRUCTURA DE LOS DATOS DE SALIDA

La motherboard presenta cuatro PUs que procesan un determinado número de canales, dependiendo de la recepción que se realice en el Módulo de Transición. Cada PU tiene información sobre su propio canal, por lo que necesitamos dos formatos para los datos: un formato de dato individual para cada PU y un formato de dato global para la motherboard y todas las PUs.

El Formato de Datos del ROD será la unión entre los datos de las cuatro PUs y el formato del evento del DAQ-1. El Formato de Datos Intermedio será la salida de cada PU.

El Formato de Datos de Salida completo del ROD se puede observar en la siguiente figura.

		Nb. Words	31	0	
Header		0	Beginning of fragment (0x0000B0F)		
		1	Start of header marker (0xEEEEEEEE)		
		1	Header size (0x20 in bytes)		
		1	Format version number		
		1	Source Identifier		
		1	Level 1 ID		
		1	Bunch crossing ID ¹¹		
		1	Level 1 Trigger Type		
		1	Detector Event Type		
	Detector Data	Status word	1	Nb of PU	PU Mask
Processing Unit 1		Nb. #0	Variable Block #0		
		Nb. #1	Variable Block #1		
		Nb. #2	Variable Block #2		
		Nb. #3	Variable Block #3		
		Nb. #4	Variable Block #4		
Processing Unit 2		Nb. #0	Variable Block #0		
		Nb. #1	Variable Block #1		
		Nb. #2	Variable Block #2		
		Nb. #3	Variable Block #3		
		Nb. #4	Variable Block #4		
Processing Unit 3		Nb. #0	Variable Block #0		
		Nb. #1	Variable Block #1		
		Nb. #2	Variable Block #2		
		Nb. #3	Variable Block #3		
		Nb. #4	Variable Block #4		
Processing Unit 4		Nb. #0	Variable Block #0		
		Nb. #1	Variable Block #1		
		Nb. #2	Variable Block #2		
		Nb. #3	Variable Block #3		
	Nb. #4	Variable Block #4			
	Status Word	1	Status flag from Output Controller		
Trailer		1	Number of Status Elements		
		1	Number of Data Elements		
		1	Status Block Position		
		0	End of Fragment (0x0E0F)		

FIGURA 2.31 Formato de los Datos de Salida del ROD.

Pasamos a continuación a definir uno por uno todos los elementos presentes en el formato de datos expuesto anteriormente:

- Los bloques de las **Processing Units** están activados sólo si la PU correspondiente no está enmascarada por la lectura de los datos.
- El **Format Version Number** es un número entero de 32 bits que se carga vía VME al mismo tiempo que la configuración. Define la versión del ROD data fragment header, no la versión del formato de los datos del detector.
- El **Source Identifier** es una palabra que define el fragmento, queda subdividida de la siguiente forma.

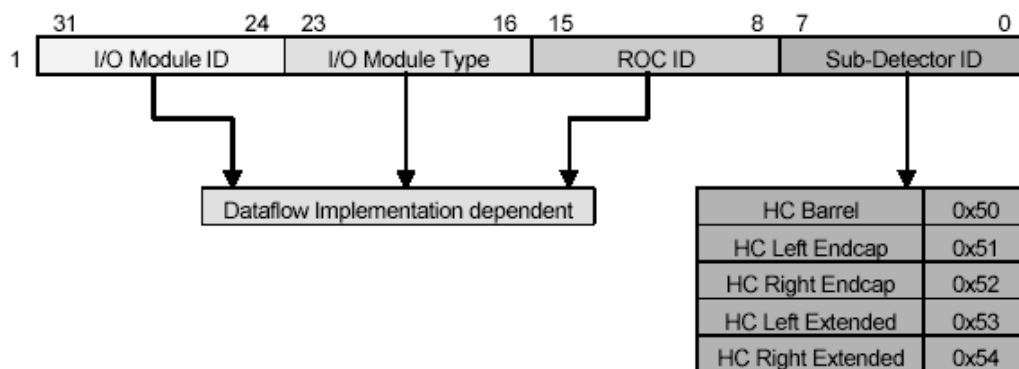


FIGURA 2.32 Source Identifier Word.

La implementación exacta de estas palabras dependerá de la organización de lectura/escritura del detector.

- El **I/O Module ID** y el **I/O Module Type** deben definirse para la implementación hardware final, hay suficientes bits para concretar todas las posibilidades que tengamos.
- El **Level 1 ID** es el identificador del evento generado por el sistema de trigger del nivel 1 (24 bits).
- El **Bunch Crossing ID** define el cruce de haz y se genera en el sistema de trigger del nivel 1 (12 bits).
- **Detector Event Type**. Este elemento identifica un evento el cual puede haber sido generado por un subdetector independiente de los otros subdetectores y de los sistemas de trigger de ATLAS.
- **PU Mask**. Se trata de un patrón de 8 bits que define que PU del ROD está leyendo los datos.
- **Detector Format Version Number**. Indica la versión del formato para el bloque de datos del detector.
- **Number of Data Elements**. Nos da la suma de los bloques del detector.
- **Status Block Position**. Se pone a cero para indicar que el Status Block precede a los datos.
- El **Status Flag from Output Controller**. Usado para marcar eventos en los cuales la lectura por medio de las PUs no ha sido correcta.

Con este apartado queda definida toda la estructura de los datos que se usan en el proceso de lectura/escritura en el Módulo de Transición. Así mismo, queda completada la distribución, diseño e implementación del dispositivo aquí estudiado.

5. BIBLIOGRAFÍA

- [1] Riu, I., *The Readout System of the ATLAS Liquid Argon Calorimeters*, CERN/LHCC 2002-34, 2002.
- [2] Castelo, J., *ROD General Requeriments and Present Hardware Solution*, General TileCal Meeting, <http://agenda.cern.ch/fullAgenda.php?ida=a02281>, 2002.
- [3] ATLAS Collaboration, *The ROD Demonstrator for the Largon Calorimeter*, ATL-COM-LARG-99-011, 1998.
- [4] ATLAS TDAQ/DCS. Global Issues Working Group, *TDAQ Run and States*, ATLAS TDAQ-2002, 2002.
- [5] LiAR Collaboration, *VME and BUSY FPGA for the ROD motherboard*, Atlas Internal Note, 2002.
- [6] Fullana, E., *Algoritmo de Filtrado Óptimo para la Reconstrucción de la Energía y el Tiempo en el Calorímetro Hadrónico de ATLAS*, XXIX Reunión Bienal de la Real Sociedad Española de Física, ISBN: 84-688-2573-5, 2003.
- [7] Cleland, W.E., *Signal processing considerations for liquid ionization calorimeters in a high rate environment*, NIMM A338, 1994.
- [8] Matricon, P., *The TM Module of the ATLAS Lar Calorimeter ROD System*, Atlas Internal Note, 2002.
- [9] Van der Bij, E., McLaren, R., *The S-LINK Interface Specification*, ECP Division. CERN, 1997.
- [10] ATLAS Collaboration, *TM4Plus1 Active S-LINK to VME64x Transition Module*, ECP Division. CERN, <http://hsi.web.cern.ch/HSI/s-link/devices/tm4plus1/>, 2000.
- [11] ATLAS Collaboration, *S-LINK Overview*, ECP Division. CERN, <http://hsi.web.cern.ch/HSI/s-link/introduccion/overview.htm>, 1995.
- [12] ATLAS Collaboration, *ATLAS Tile Calorimeter S-Link Interface Card*, Atlas Internal Note, 2001.
- [13] Torroja, Y., *Diseño Lógico mediante FPGAs*, Universidad Politécnica Madrid, 2001.
- [14] Torres, J., González, V., *QUARTUS II: Una Herramienta para el Diseño Digital Avanzado*, Apuntes Curso Extensión Universitaria, 2002.
- [15] Castelo, J., Torres, J., González, V., *On the Developments of the Read Out Driver for the ATLAS Tile Calorimeter*, CERN/LHCC 2001-34, 2001.
- [16] Integrated Device Technology, *IDT72V3660 Datasheet*, FIFO Applications Guide, <http://www.idt.com/products/pages/FIFOs-72V3660.html>, 2003.
- [17] ATLAS Collaboration, *ODIN LSC*, ECP Division. CERN, <http://hsi.web.cern.ch/HSI/s-link/devices/odin/>, 2001.
- [18] ATLAS Collaboration, *ODIN S-LINK Hardware Specification*, ECP Division. CERN, <http://hsi.web.cern.ch/HSI/s-link/devices/odin/hwspec.html>, 2001.
- [19] Castelo, J., *The I/O Dataformat for the Tilecal Read Out System*, ROD TileCal Group Internal Note, 2003.

CAPÍTULO

3

SISTEMA ROD FINAL Y DESARROLLO DE LA OPTICAL MULTIPLEXER BOARD

1. INTRODUCCIÓN	1
2. SISTEMA READ OUT DRIVER (ROD) FINAL	1
2.1 Descripción de la ROD Motherboard	1
2.2 Descripción de la Staging FPGA	4
3. OPTICAL MULTIPLEXER BOARD (OMB)	6
3.1 Estudios de Radiación en el Front End de TileCal	6
3.2 Descripción de la Optical Multiplexer Board	8
3.2.1 Descripción del Conector de Fibra Óptica de Entrada/Salida	11
3.2.2 Descripción del Deserializador	11
3.2.3 Descripción de la FPGA de CRC	14
3.2.4 Descripción de la FPGA de VME	15
3.2.5 Descripción del Serializador	17
3.2.6 Generación y distribución de los relojes en la OMB	19
3.2.7 Distribución de los datos en la OMB	21
3.2.8 Distribución de la alimentación en la OMB	22
3.2.9 Conectores J1/P1 y J2/P2	22
3.2.10 Señales del Panel Frontal	23
3.3 Desarrollo y Costes de la OMB	24
3.3.1 Realización de los esquemáticos	24
3.3.2 Diseño del circuito impreso	27
3.3.3 Rutado del circuito impreso	29
3.3.4 Costes de la OMB	32
3.4 La OMB como Multiplexora de Datos	33
3.4.1 Bloque Reception Fiber A-B	33
3.4.2 Bloque CRC Check, Data Multiplexing & Control Unit	34
3.4.3 Bloque Transmission	36
3.4.4 Bloque VME FPGA	37
3.4.5 Bloque CRC Channel 2 FPGA	37
3.4.6 Bloque CRC FPGA Clock and Test	37

3.4.7 Bloque Connector Test	37
3.5 La OMB como Inyectora de Datos	38
3.5.1 Bloque External Trigger & Busy	38
3.5.2 Bloque Clock Divider	38
3.5.3 Bloque Event Generator	39
3.5.4 Bloque Transmission	39
3.5.5 Bloque Injector Test	39
3.5.6 Bloque Connector Test	39
3.6 Protocolo VME en la OMB	40
3.6.1 Bloque Prerod_Genadd	40
3.6.2 Bloque CR_CSR	41
3.6.3 Bloque VME_Registers	41
4. ESTUDIOS SOBRE INTEGRIDAD DE LA SEÑAL	42
4.1 Fundamentos del Diseño Digital para Alta Velocidad	42
4.1.1 Revisión de los Conceptos de Frecuencia, Tiempo y Distancia	43
4.1.2 Líneas en PCBs	47
4.1.2.1 Líneas MicroStrip	47
4.1.2.2 Líneas Stripline	51
4.1.3 Terminaciones	52
4.1.3.1 Formas de Onda en una Línea no Terminada	53
4.1.3.2 Terminación en paralelo	54
4.1.3.3 Terminación en serie	56
4.2 Análisis Pre-Layout de la OMB	57
4.2.1 Líneas Diferenciales	58
4.2.2 Bus de Datos y Señales de Control	62
4.2.3 Relojes	66
4.2.4 Crosstalk	69
5. BIBLIOGRAFÍA	72

1. INTRODUCCIÓN

Después de encuadrar el Prototipo del Sistema ROD en la estructura de adquisición de datos de ATLAS y de ver las características de los elementos fundamentales del mismo (Motherboard, PUs y Módulo de Transición Activo), en este capítulo se va a estudiar el diseño del Sistema ROD Final.

Este Sistema ROD difiere en algunos aspectos al del Prototipo comentado en el capítulo anterior, todo ello como resultado de la experiencia adquirida con el mismo. Se realizará una nueva descripción del mismo, haciendo hincapié en las novedades que presenta.

Así mismo, desarrollaremos en profundidad la nueva tarjeta que se ha incorporado al Sistema ROD. Estamos hablando de la Optical Multiplexer Board (OMB), un nuevo módulo que ha surgido desde la necesidad producida por la gran cantidad de radiación que presentará TileCal en el momento de su funcionamiento.

Veremos los estudios preliminares que dan origen a su nacimiento, una descripción detallada de sus características principales, el desarrollo a nivel estructural de la misma, los elementos que la componen, un estudio de los distintos modos de funcionamiento que se han introducido y los algoritmos usados para el análisis de los datos.

Es en este módulo donde se ha centrado esta Tesis Doctoral, desde su idea hasta su testeo tanto en los laboratorios de la Universidad y del IFIC en Valencia como en los laboratorios del CERN, pasando por su desarrollo, su necesidad, los estudios de integridad de la señal, su programación y su puesta en funcionamiento. Trabajando así mismo en colaboración directa con el desarrollo del Sistema ROD Final.

2. SISTEMA READ OUT DRIVER (ROD) FINAL

El Sistema Read Out Driver (ROD) Final [1], tal y como se ha comentado en el capítulo anterior, es un enlace intermedio de la cadena formada por la electrónica del front-end y el sistema de adquisición de datos general del detector ATLAS (TDAQ). Representa la parte final de la electrónica del subdetector TileCal.

El Sistema ROD Final de TileCal presenta como elementos principales una Motherboard y la Optical Multiplexer Board, la Motherboard ya incluida en el Sistema RODDemo y una nueva etapa en la adquisición de datos desarrollada en la Optical Multiplexer Board. Entre la nueva Motherboard y la del antiguo Sistema existen diferencias estructurales importantes que nos dan una mayor flexibilidad en la adquisición de datos y que nos permiten adaptar la Optical Multiplexer Board al Sistema ROD.

2.1 DESCRIPCIÓN DE LA ROD MOTHERBOARD

Esta Motherboard [2], al igual que la anterior, es un módulo 9U VME 64x que presenta cuatro tarjetas con unidades de proceso como tarjetas mezzanine montadas encima de ella. Recibe los datos de la Optical Multiplexer Board a través de ocho fibras ópticas y envía los resultados de los cálculos realizados con el algoritmo de Filtrado Óptimo hacia un Módulo de Transición Pasivo (TMP) instalado en la parte trasera de la crate de VME.

Una lista básica de las funcionalidades incluidas en este módulo son:

- Procesa y chequea la consistencia de los datos enviados por la OMB.
- Calcula la energía y el tiempo a través del Algoritmo de Filtrado Óptimo implementado en los DSPs de las Unidades de Proceso (PUs).

- Los resultados son enviados a los módulos ROB (Read Out Buffers) donde la información se convierte en accesible para las siguientes decisiones de trigger.
- Monitoriza los datos para asegurar una buena calidad en el registro de los datos.
- Genera las señales BUSY para el CTP (Central Trigger Processor) en caso de problemas en el tratamiento de los datos.

Tendremos un total de 32 Tarjetas ROD VME64x para leer la totalidad de los datos del detector TileCal, cada una de ellas con cuatro PUs. La siguiente figura muestra una fotografía de la Motherboard del Sistema ROD Final.

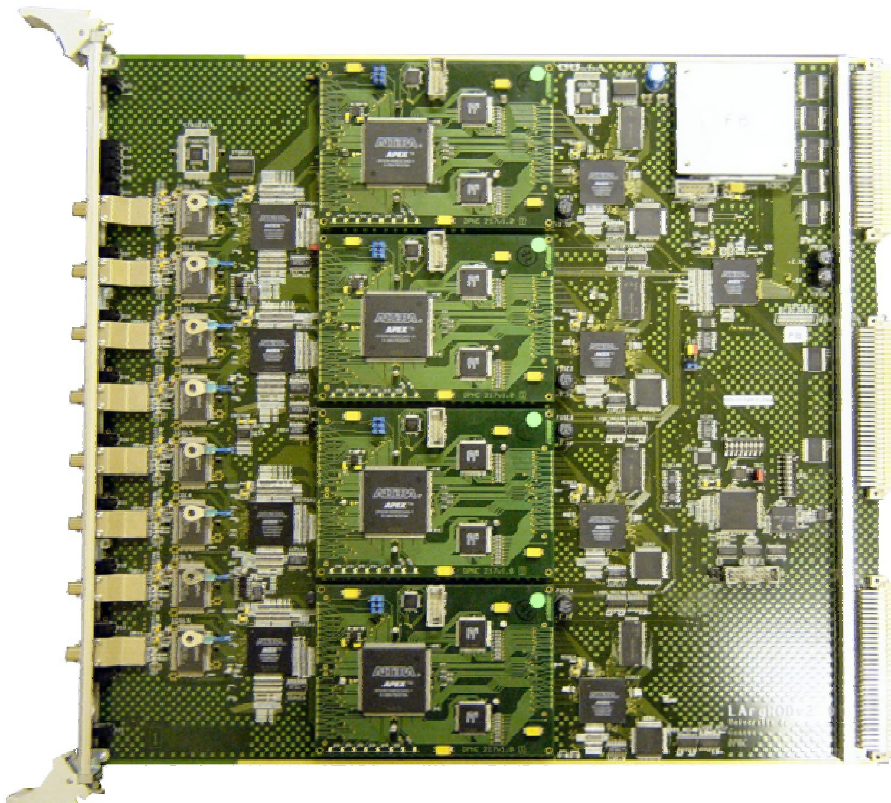


FIGURA 3.1 Fotografía de la nueva Motherboard del Sistema ROD.

Estos RODs operan en una conversión óptica-electrónica a través de ocho receptores ópticos (ORX) y que nos aseguran una compatibilidad completa con la OMB y las tarjetas del Front End (FEB).

Las principales diferencias entre las dos motherboards, la del RODDemo comentado en el capítulo anterior y la del ROD Final, podemos observarlas en la siguiente tabla.

Ítem	RODDemo	ROD Final
Frecuencia	40 MHz	40 MHz
Entrada	2 receptores ópticos	8 receptores ópticos
Salida	1 S-Link óptico	4 S-Links óptico
Número de canales	2 x 64	8 x 128
Tarjetas Mezzanine	Tarjetas PU Tarjeta TTC	Tarjetas PU Receptores ópticos
	Receptores ópticos y G-Links S-Link en el Módulo de Transición	S-Link en el Módulo de Transición

TABLA 3.1 Principales diferencias entre la Motherboard del RODDemo y del ROD Final.

Comentar que la principal diferencia reside en la entrada de datos; en el RODDemo se realizaba por el Módulo de Transición y ahora se realiza directamente por el panel frontal del ROD Final. Así mismo, se han integrado FPGAs a la entrada para el primer tratamiento de los datos y para la implementación del protocolo de comunicaciones con la Optical Multiplexer Board.

Como también se puede observar, la salida de los datos se realiza ahora por medio de un Módulo de Transición Pasivo conectado por VME a la Motherboard. El diagrama de bloques de la nueva Motherboard del Sistema ROD se puede observar en la siguiente figura.

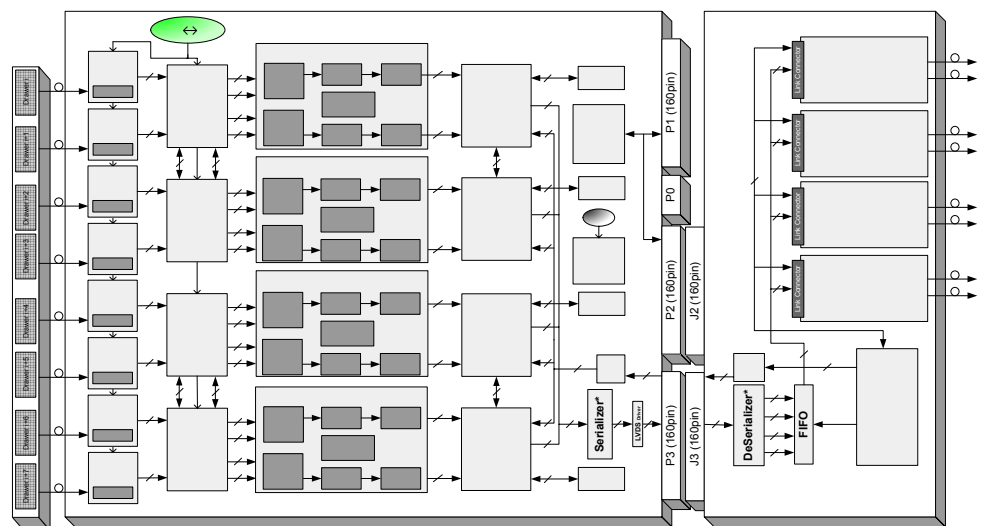


FIGURA 3.2 Diagrama de bloques de la Motherboard del Sistema ROD Final.

Una descripción de las principales funciones que realizan los elementos presentes en la Motherboard del ROD se da en la siguiente tabla.

Componente	Principal Función	Chip
8 G-Links Chips	Deserializa los datos de entrada	HDMP-1024
4 Staging FPGAs	Ruta los datos de entrada hacia las PUs Monitoriza la temperatura de los G-Links Envia los datos del evento hacia las PUs para Tests	ACEX EP1K50
4 Output Controller FPGAs	Tratamiento de los datos de salida	ACEX EP1K100
1 VME FPGA	Interface del ROD con VME Tratamiento de la señal Busy y de las Interrupciones	ACEX EP1K100
1 TTC Controller FPGA	Toma el Reloj general del LHC desde el Back Plane de VME y lo ruta por todo el ROD	ACEX EP1K30

TABLA 3.2 Descripción de las funciones de los elementos principales del ROD.

2.2 DESCRIPCIÓN DE LA STAGING FPGA

Cuatro Staging FPGAs [3] son las que se usan en la Motherboard del Sistema ROD Final. La principal razón para escoger este dispositivo es el amplio número de pines que presenta y que se ajustan a nuestras necesidades. Los datos de entrada de dos canales, una vez deserializados en los G-Links, son enviados a una Staging FPGA.

En el comienzo del LHC, las Motherboards del ROD estarán equipadas sólo con la mitad de las PUs que pueden montarse en ellas. Esto es lo que llamamos modo staging de funcionamiento (probablemente TileCal nunca necesite trabajar con las cuatros PUs). Para este modo de funcionamiento, se necesita un bus que comunique internamente las Staging FPGAs. Por tanto, en este modo, los datos de entrada de cuatro canales serán rutados a una única PU.

La distribución de los datos en este modo de funcionamiento se puede observar en la siguiente figura.

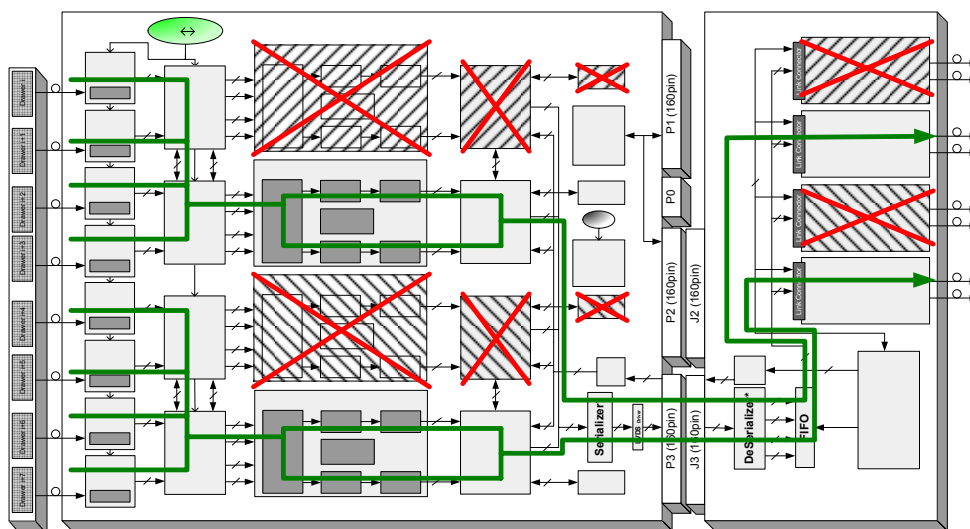


FIGURA 3.3 Diagrama de bloques de Motherboard del ROD en Modo Staging.

La identificación de las Staging FPGAs se realiza por hardware, dos pines de cada Staging FPGA están conectados a GND and/or a VCC. En particular, cada Staging FPGA queda identificada por un número binario como 00, 01, 10, 11.

Internamente, se han implementado en VHDL los siguientes bloques que definen la funcionalidad completa de este dispositivo.

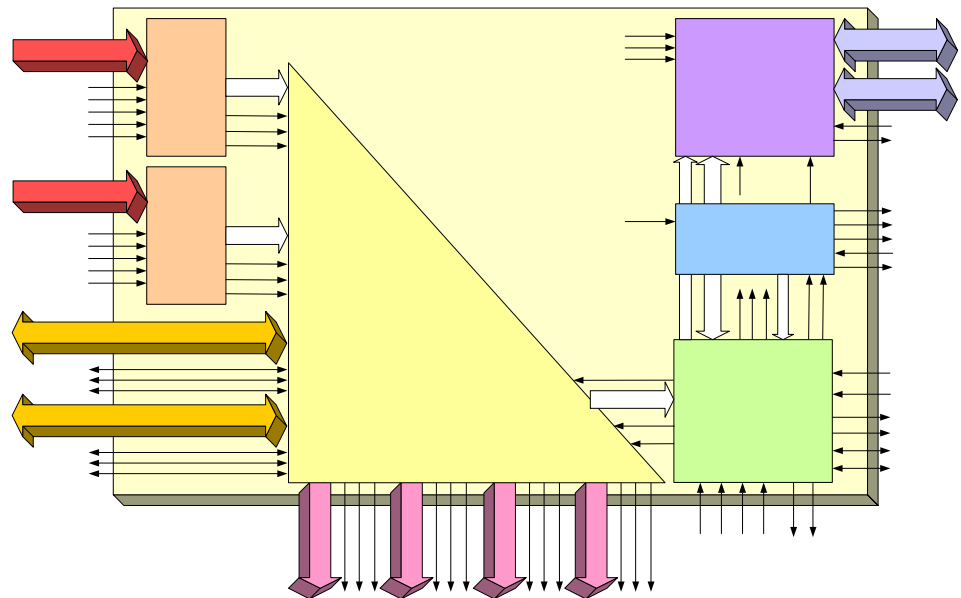


FIGURA 3.4 Diagrama de bloques interno de la Staging FPGA.

Las principales funcionalidades son:

- En el inicio, resetear y configurar los G-Links.
- Recibir los datos deserializados de los dos G-Links (en modo de funcionamiento Normal) o de cuatro G-Links (en modo de funcionamiento Staging).
- Realizar un control y sincronización de los relojes de envío y recepción.
- Proporcionar acceso por VME para tareas de configuración y estado.
- Monitorizar la temperatura de los G-Links para evitar calentamientos que puedan dañar el componente.
- Enviar los datos testados hacia las PUs.

3. OPTICAL MULTIPLEXER BOARD (OMB)

TileCal es un sistema de adquisición redundante; del Front End [4] nacen dos fibras ópticas con el mismo dato. Esto se hizo necesario debido a los fenómenos de radiación que se producen en el detector. Dichos fenómenos pueden causar funcionamientos erróneos dentro de la electrónica del Front End y errores de ráfaga sobre los datos en el momento de transmisión hacia el ROD.

En un principio esto no fue tenido en cuenta, y el ROD se diseñó para recibir únicamente una fibra por canal. Así se respondía a las especificaciones iniciales del proyecto. Más tarde, y con un mayor número de pruebas de radiación en la mano, se decidió incluir una segunda fibra por canal. Es en ese momento donde nace la Optical Multiplexer Board.

El principal objetivo de esta tarjeta es mejorar la tolerancia a errores del sistema, analizando las dos fibras por canal que presenta TileCal. Ambas fibras con los mismos datos, decidiendo cual de ellas es correcta y enviándola al ROD.

El interés de este proyecto quedó justificado y aprobado por la colaboración en Febrero del 2003 cuando se presentaron los estudios preliminares que se estaban realizando. Esta propuesta mostraba una solución basada en un exhaustivo análisis on-line de los datos enviados por las fibras usando FPGAs para su implementación.

3.1 ESTUDIOS DE RADIACIÓN EN EL FRONT END DE TILECAL

ATLAS requiere una electrónica que pueda operar con buena fiabilidad. Para ello, hablando en el campo de la radiación, necesitamos una electrónica que sea estable durante un periodo de 10 años. Después de realizar estudios de radiación para cada uno de los subsistemas que presenta ATLAS, se llegó a la conclusión de que las exigencias en materia de radiación para cada uno de ellos eran diferentes [5].

Así pues, basándonos en el campo de radiación calculado, la electrónica de TileCal recibirá una dosis de radiación ionizada de ~ 2 Gy/year (0,2 Krad/year) correspondientes a una dosis total de tiempo de vida de 20 Gy. También estará expuesto a una fluidez de neutrón de 10^{11} 1-MeV-neutrones equivalentes/cm²/year. El espectro de energía relevante del neutrón está entre ~ 100 KeV y ~ 2 MeV. Se debe aplicar un factor de 5 como factor de seguridad a toda la electrónica que se diseñe y que pueda presentar efectos de radiación.

Tratando la Optical Interface Board (OIB), tarjeta situada cerca del centro de TileCal (Front End) y emisora de datos hacia el ROD, vemos que un estudio a fondo de la radiación que incide sobre ella se hace necesario. Esta tarjeta, como recordatorio, recibe los datos digitales de las tarjetas digitalizadoras situadas en los drawers de TileCal. Los datos de cada uno de los eventos son ensamblados, formateados y enviados fuera del drawer mediante fibra óptica hacia el Sistema ROD.

Esta tarjeta (OIB) [6] presenta sólo dos chips que mostraron sensibilidad y pérdidas por radiación mediante un test con radiación ionizada a 50 Krad. Los requerimientos de diseño eran de 10 Krad, incluyendo un factor de seguridad de 5 y para 10 años de operación a la luminosidad de diseño. Para este nivel de exposición, ambos chips funcionaron con normalidad, no hubo destrucción de los mismos.

Así mismo, la tarjeta fue irradiada por neutrones y tampoco se observó deterioro en los componentes. Los chips en cuestión eran una FPGA de Altera y un DAC de Exar. La posición sobre la cual se incidió el haz de protones en la tarjeta fue variando para cada test. Aquí se puede ver como se realizó.

Ninguno de ellos provoca una destrucción del componente o de la tarjeta pero sí nos pueden producir un error en la transmisión de los datos hacia el ROD. Esos datos pueden estar corruptos, con lo que todo el tratamiento que hagamos de ellos puede ser en vano.

Para evaluar el impacto real de la radiación sobre las tarjetas, deberíamos tener las 256 tarjetas y un entorno parecido al que tendrá finalmente el LHC. Como eso es inviable, tenemos que conformarnos con estos tests. Así pues, la Colaboración decidió reducir la posibilidad de que se tuvieran pérdidas de datos en el Calorímetro duplicando los sistemas. ¿Qué quiere decir eso?, quiere decir que tendremos el mismo número de tarjetas pero que cada una de ellas presentará dos FPGAs y dos DACs.

Por tanto, tendremos los mismos datos por dos canales distintos, paralelos y que llegarán al Sistema ROD por dos fibras diferentes. Ahí es donde surge la Optical Multiplexer Board.

3.2 DESCRIPCIÓN DE LA OPTICAL MULTIPLEXER BOARD

Pasamos a continuación a describir con total detalle la Optical Multiplexer Board [7], objeto principal de estudio, diseño e implementación de esta Tesis Doctoral. Es en ella donde se ha centrado nuestro trabajo en el último año y medio. Esta tarjeta ha sido pensada, diseñada, estudiada, construida y puesta en funcionamiento entre 2003 y 2004. En un principio, los objetivos preliminares y primordiales de la misma eran:

- Implementar un multiplexaje óptico para sacar provecho de la redundancia de datos que presenta TileCal.
- Obtener los costes reales en producción de los módulos necesarios.
- Diseñar una plataforma software y hardware para su desarrollo.
- Probar diferentes alternativas para el análisis de los errores en los datos.

La Optical Multiplexer Board estaba pensada para trabajar en “Modo Multiplexor”, pero debido a su naturaleza y a ser el paso previo a la Motherboard del ROD, se pensó también para trabajar en “Modo Inyector de Datos”. En este modo de funcionamiento, podemos inyectar patrones de datos conocidos por nosotros a la Motherboard del ROD, realizando verificaciones del funcionamiento de dichos módulos.

Se trataba en un principio de una característica más de la Optical Multiplexer Board, que con el tiempo ha alcanzado una importancia tremenda. Al no tener siempre un haz que nos proporcione datos para realizar nuestros tests, la capacidad de la Optical Multiplexer Board para generar datos de cualquier tipo, la convierte en un Banco de Test de gran importancia para testear cualquier diseño electrónico de nuestro experimento.

En cuanto a la mecánica, la Optical Multiplexer Board ha sido diseñada como una arquitectura 6U VME64X [8], tratándola como un módulo esclavo. Sus medidas son 160,0/233,35/2,4 mm correspondientes a altura/anchura/profundidad. La figura siguiente muestra las dimensiones reales de esta tarjeta.

Una descripción más detallada de cada bloque es la siguiente:

3.2.1 Descripción del Conector de Fibra Óptica de Entrada/Salida

Cuatro fibras ópticas que provienen del Front End y que representan a dos canales de datos son las entradas a la Optical Multiplexer Board. Dos fibras que salen de la esta tarjeta y que terminan en la Motherboard del ROD son las salidas de la misma.

El componente que permite la conexión de las fibras ópticas es un Small Form Factor Emisor/Receptor de la casa Infineon (V23818-K305-L15) [9], es de tipo multimodo 850 nm y cumple el estándar Gigabit Ethernet. Opera entre los 1,0625 y 1,25 Gbit/s y su función es la de convertir la señal óptica en una señal eléctrica en formato LVPECL. Nos permite distancias de hasta 700 m y una conexión mediante un conector LC.

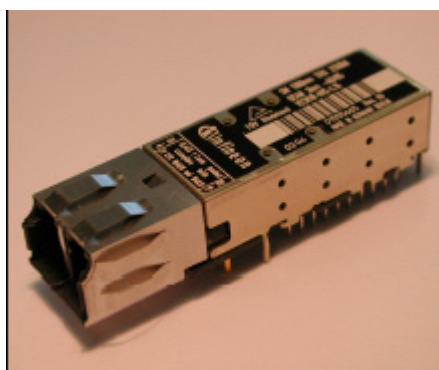


FIGURA 3.9 Fotografía del conector de fibra óptica.

3.2.2 Descripción del Deserializador

Una vez los datos han sido convertidos del dominio óptico al dominio eléctrico, son enviados al siguiente componente de nuestra tarjeta. Se trata de un deserializador, concretamente el chip G-Link (HDMP-1034) [10] que se usa para convertir los datos serie de entrada en datos paralelos. Añadiéndole una serie de señales de control que nos permiten definir el protocolo de entrada de nuestro sistema.

El receptor HDMP-1034 se usa junto a un transmisor (HDMP-1032) colocado en el emisor para establecer conexiones punto a punto de alta velocidad. Se trata de dispositivos con tecnología bipolar que vienen en un encapsulado de 64 pines tipo PQFP.

Desde el punto de vista del usuario, estos dispositivos actúan como un interfaz para la transmisión de datos y de palabras de control. Una palabra en paralelo cargada en el transmisor del emisor, es transmitida vía fibra óptica y recibida por el receptor en modo serie. Éste se encarga de reconstruirla y de convertirla nuevamente en la palabra paralela original.

El flujo de los datos se puede seleccionar entre tres rangos que van desde 208 hasta 1120 Mbit/s. Los datos paralelos que salen del receptor hacia la FPGA presentan una tecnología TTL.

Así mismo, se envían una serie de palabras de control que nos indican el inicio y el final de los datos. Permitiendo así, una sincronización correcta de los mismo. Algo imprescindible en velocidades tan altas de tratamiento de datos.

La siguiente figura nos muestra el enlace que hemos escogido para nuestro sistema, donde Tx representa el transmisor colocado en el Front End de TileCal y Rx el receptor de nuestra tarjeta.

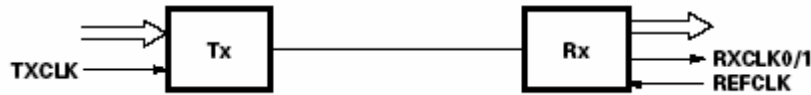


FIGURA 3.10 Transmisión simple de 16 bits.

Para la selección del rango de operación y para ajustarlo a nuestras especificaciones (40 MHz), nos servimos de dos señales que podemos configurar desde la FPGA. Estas señales son DIV1 y DIV0. Siguiendo la siguiente tabla, escogimos una configuración de DIV1 = 0 y DIV0 = 1.

HDMP-1032A (Tx), HDMP-1034A (Rx)
 Typical Operating Rates^{1,2}
 T_c = -20°C to +85°C, V_{CC} = 3.15V to 3.45V

DIV1	DIV0	Parallel Word Rate (MWord/sec)		Serial Data Rate (MBits/sec)		Serial Baud Rate (MBaud)	
		Range	Range	Range	Range		
0	0	40	70 (max)	640	1120 (max)	800	1400 (max)
0	1	20	45	320	720	400	900
1	0	13 (min)	26	208 (min)	416	260 (min)	520

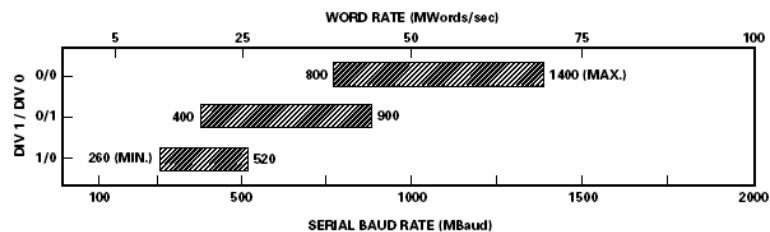


FIGURA 3.11 Rangos de operación del HDMP.

El diagrama de bloques interno del HDMP-1034 Rx, figura 3.12, nos da una idea de cómo recibe los datos en serie y los convierte en paralelo. Es de vital importancia aquí el tener un reloj de entrada (REFCLK) de características correctas para que se produzca el enganche entre el dispositivo de recepción y el de transmisión.

Lo primero que hace el HDMP es compensar posibles distorsiones mediante el bloque Clock Data Recovery (CDR) y recibir el reloj antes mencionado. Dicho reloj es un reloj colocado en nuestra tarjeta que mediante el bloque Clock Generator nos da dos nuevos relojes que marcarán el proceso de salida de los datos (RXCLK0/1).

En los siguientes bloques, se demultiplexan los datos de entrada y se identifican los que son datos y los que son señales de control. El último proceso que tienen los datos antes de salir lo hace el bloque Pass System, el cual realiza una sincronización de todas las señales para su correcto envío.

Aquí podemos observar lo anteriormente explicado:

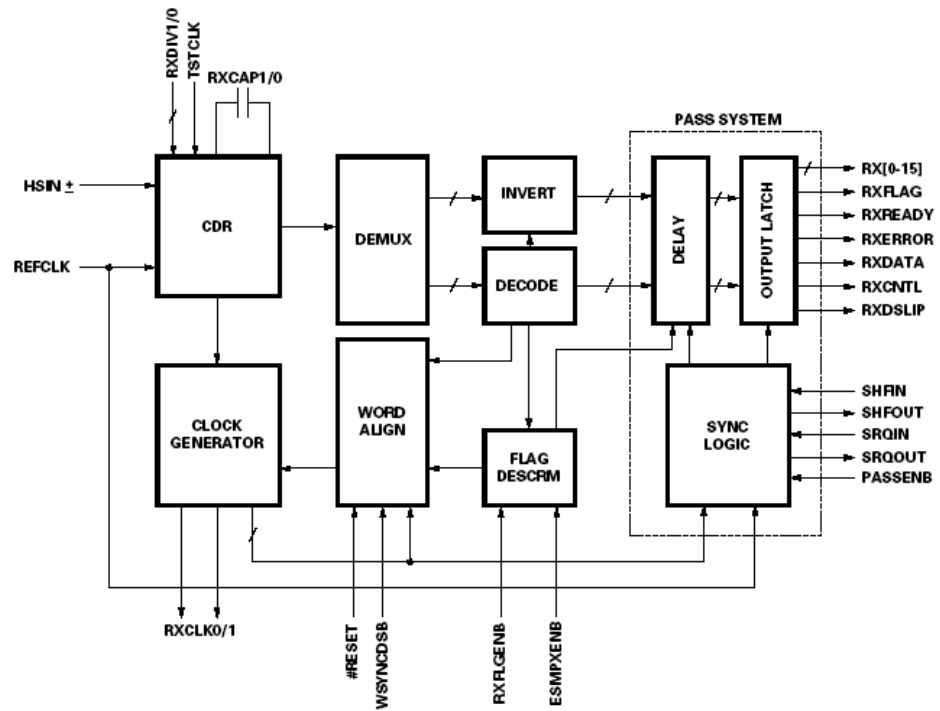
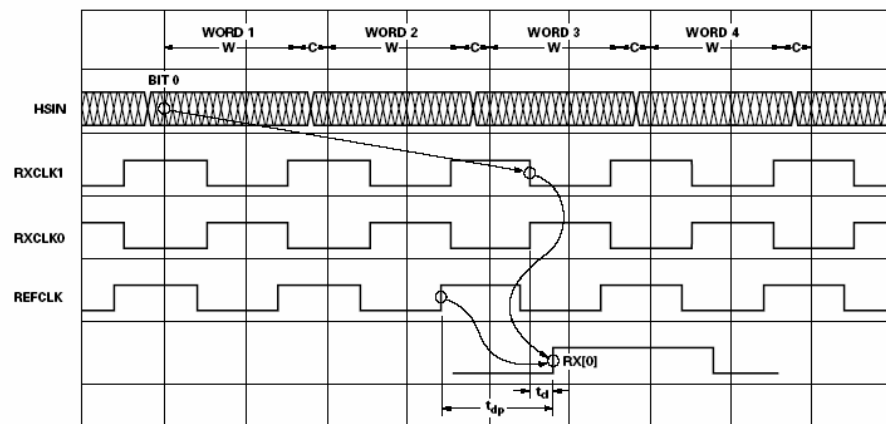


FIGURA 3.12 Diagrama de bloques interno del HDMP-1034 Rx.

El diagrama de tiempos de este dispositivo viene determinado por:



NOTE: W = 16 BIT WORD FIELD,
C = 4 BIT CODE FIELD

FIGURA 3.13 Diagrama de tiempos del HDMP-1034 Rx.

Tener unos relojes síncronos que nos mantengan el protocolo uniforme y compacto es de vital importancia.

3.2.3 Descripción de la FPGA de CRC

Esta FPGA recibe el nombre por ser su principal función el cálculo del CRC de los datos que recibimos del Front End. Se trata de una FPGA de Altera, concretamente de la familia Cyclone (EP1C12) [11]. La principal razón para usar esta FPGA ha sido su bajo coste, manteniendo las principales características que nuestro diseño imponía.

Los datos deserializados en el HDMP son enviados a esta FPGA para su posterior tratamiento, es aquí donde se configura el protocolo de comunicaciones que mantiene el enlace activado. Se reciben los datos de las dos fibras ópticas del mismo canal para su análisis y su posterior decisión de canal correcto.

Así mismo, una vez tomada la decisión de qué fibra lleva los datos correctos, estos son enviados a un serializador para su posterior envío a la Motherboard del ROD. Como ya hemos comentado anteriormente, este modo de funcionamiento es el principal, siendo capaz la tarjeta de funcionar como inyectora de datos. Más adelante veremos con mayor detalle su funcionamiento.

Ahora vamos a explicar cómo es internamente una FPGA Cyclone. Es importante su conocimiento para conseguir un rutado, un emplazamiento y una síntesis lo más correcta posible.

La familia Cyclone pertenece a los dispositivos llamados Field Programmable Gate Array (FPGAs, ya referidos con anterioridad), en concreto a los que están basados en tecnología de 1,5 V y 0,13 μm con densidades de hasta 20.060 elementos lógicos (LEs) y 288 Kbits de RAM.

Internamente presentan PLLs para una mejor señal de reloj y una DDR dedicada para mejoras en los requerimientos de memoria. Estos dispositivos soportan diferentes estándares de entrada/salida, incluyendo LVDS con tasa de datos de hasta 640 Megabits por segundo (Mbps).

A su vez, permiten la interconexión de periféricos PCI como conexión a dispositivos ASIC. Las siguientes tablas nos dan una idea de la variedad de dispositivos que tenemos dentro de la familia Cyclone, sus características principales, su tamaño y los pines de entrada/salida de que disponen.

Feature	EP1C3	EP1C4	EP1C6	EP1C12	EP1C20
LEs	2,910	4,000	5,980	12,060	20,060
M4K RAM blocks (128 × 36 bits)	13	17	20	52	64
Total RAM bits	59,904	78,336	92,160	239,616	294,912
PLLs	1	2	2	2	2
Maximum user I/O pins (1)	104	301	185	249	301

Device	100-Pin TQFP (1)	144-Pin TQFP (1), (2)	240-Pin PQFP (1)	256-Pin FineLine BGA	324-Pin FineLine BGA	400-Pin FineLine BGA
EP1C3	65	104				
EP1C4					249	301
EP1C6		98	185	185		
EP1C12			173	185	249	
EP1C20					233	301

Dimension	100-Pin TQFP	144-Pin TQFP	240-Pin PQFP	256-Pin FineLine BGA	324-Pin FineLine BGA	400-Pin FineLine BGA
Pitch (mm)	0.5	0.5	0.5	1.0	1.0	1.0
Area (mm ²)	256	484	1,024	289	361	441
Length × width (mm × mm)	16 × 16	22 × 22	34.6 × 34.6	17 × 17	19 × 19	21 × 21

FIGURA 3.14 Características de la familia Cyclone.

La programación de esta FPGA se ha realizado mediante un dispositivo de configuración serie (EPCS4). Se trata de una memoria Flash de 4 Mbits que configura de modo serie el dispositivo Cyclone usando el esquema de configuración serie que se muestra en la siguiente figura:

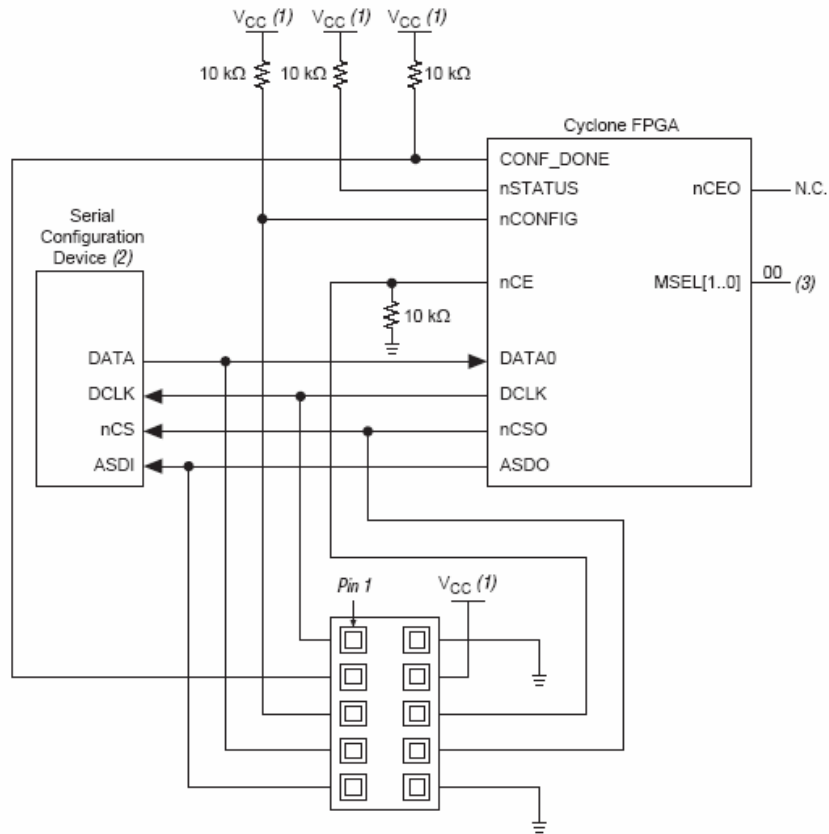


FIGURA 3.15 Programación serie de la FPGA Cyclone.

Se basa en un interfaz entre el ordenador y el software que proporciona el fabricante con la conexión física de la memoria mediante un cable de programación. Es de bajo coste, de pocos pines de conexión y de memoria no volátil. El dispositivo debe ser reseteado cada vez que se realiza una nueva programación del mismo.

3.2.4 Descripción de la FPGA de VME

El siguiente paso, es explicar el dispositivo de VME. Éste nos sirve como interfaz entre nuestra tarjeta y la CPU instalada en la Crate del Sistema ROD. Recordemos que nuestra tarjeta actúa como un módulo esclavo de VME. Por tanto, todas las acciones son comandadas y controladas por la CPU del Sistema.

Este interfaz con el mundo exterior, está implementado en otra FPGA. En este caso se trata de una ACEX EP1K100 de Altera [12]. El porqué de su elección sigue criterios del CERN. Nuestra tarjeta es una más dentro del mundo electrónico en el cual se mueve ATLAS, por ello, debemos seguir algunos estándares dados por la Colaboración.

Así mismo, nos permite acceder desde el exterior a las FPGAs de CRC que es el lugar donde se está tratando la información. ¿Qué conseguimos con ello?, conseguimos

poder monitorizar por el bus de VME los datos que llegan y que salen de nuestra FPGA de CRC. Podemos, por tanto, tener un control externo de las operaciones que se estén realizando en ellas. Y un aspecto muy importante, conseguimos decidir de forma externa si nuestro sistema debe actuar en “Modo Multiplexor” o en “Modo Inyector de Datos”.

Igual que en el apartado anterior, dedicaremos un punto aparte para el desarrollo completo de esta FPGA. Ahora nos centraremos en definir algunas de las principales características de este dispositivo.

La ACEX 1K es un dispositivo de lógica programable de Altera (PLD), integrado en un System-On-a-Programmable-Chip (SOPC) de bajo coste. Contiene un array para la implementación de Megafunctions del fabricante, consiguiendo una mayor eficiencia en los procesos de memoria. Su alta densidad de integración permite que presente entre 10.000 y 100.000 puertas típicas, además de 49.152 bits de RAM.

Es muy usado en aplicaciones donde la relación coste-optimización es importante. Al igual que en el caso de las Cyclone, las siguientes tablas definen perfectamente las características más importantes de esta familia.

Feature	EP1K10	EP1K30	EP1K50	EP1K100
Typical gates	10,000	30,000	50,000	100,000
Maximum system gates	56,000	119,000	199,000	257,000
Logic elements (LEs)	576	1,728	2,880	4,992
EABs	3	6	10	12
Total RAM bits	12,288	24,576	40,960	49,152
Maximum user I/O pins	136	171	249	333

Device	100-Pin TQFP	144-Pin TQFP	208-Pin PQFP	256-Pin FineLine BGA	484-Pin FineLine BGA
EP1K10	66	92	120	136	136 (3)
EP1K30		102	147	171	171 (3)
EP1K50		102	147	186	249
EP1K100			147	186	333

Device	100-Pin TQFP	144-Pin TQFP	208-Pin PQFP	256-Pin FineLine BGA	484-Pin FineLine BGA
Pitch (mm)	0.50	0.50	0.50	1.0	1.0
Area (mm ²)	256	484	936	289	529
Length × width (mm × mm)	16 × 16	22 × 22	30.6 × 30.6	17 × 17	23 × 23

FIGURA 3.16 Características de la familia ACEX.

A la hora de hablar de la programación de este dispositivo, hemos usado una configuración basada en JTAG y Passive Serial. Ocurre lo mismo que en el caso de las Cyclone, el dispositivo debe ser reseteado para que la programación quede completada debido al uso de una memoria intermedia (EPC2) entre conector y FPGA.

Usando estas configuraciones para las dos familias de FPGAs, hemos conseguido asegurarnos una programación correcta. A su vez, con un único cable conectado al puerto paralelo (ByteBlasterMV) podemos programar las tres FPGAs de nuestro módulo.

Esta figura muestra el esquemático utilizado para dicha programación.

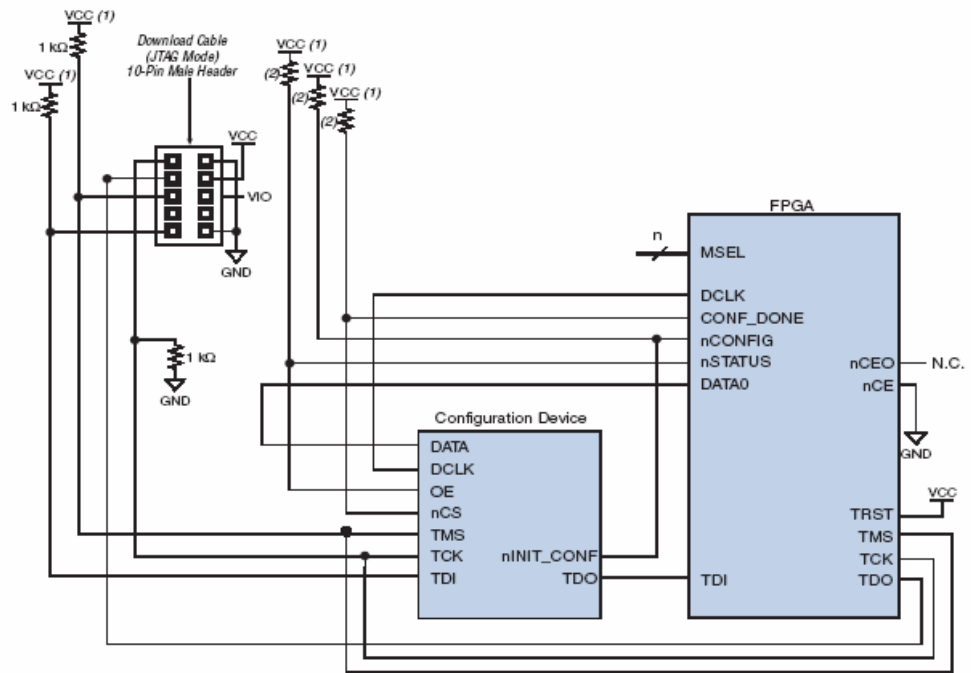


FIGURA 3.17 Diagrama de bloques Reformatting Altera FPGA.

3.2.5 Descripción del Serializador

Siguiendo con la descripción de los elementos presentes en la Optical Multiplexer Board, le llega el turno por orden de paso y tratamiento de los datos (objetivo primordial de esta tarjeta) al Serializador de salida.

Una vez los datos han sido tratados en la CRC FPGA, bien como multiplexado de los mismos, bien como inyector de datos, estos pasan al chip G-Link (HDMP-1032) [10] que se usa para convertir los datos y señales de control enviadas en paralelo desde la FPGA en datos series, los cuales son transmitidos por fibra óptica a la Motherboard del Sistema ROD.

El enlace escogido para la salida de los datos se corresponde a la Figura 3.10. El rango de operación también es el mismo que para el receptor o deserializador explicado en el apartado 3.2.2.

Lo que sí varía es tanto el diagrama de bloques interno del dispositivo como el diagrama de tiempos que cumple este protocolo de comunicaciones.

Uno de los aspectos más importantes de este dispositivo, y que lo diferencia del de recepción, es el reloj. Mientras que el Rx necesita un reloj de placa que continuamente este enviándole dicha señal, el Tx sólo lo necesita cuando hay un envío de datos desde la FPGA hasta él. El reloj va incorporado a las señales de control y a los datos que queremos serializar.

Como en Rx, éste reloj debe tener unas características bien definidas (hablaremos de él y de su distribución en un punto posterior). Es muy importante que sea de muy buena calidad por ser el que primero enganchará con el Rx de la Motherboard del ROD. En caso contrario, el enlace será inestable y perderemos datos en la transmisión.

El diagrama de bloques del chip Tx (HDMP-1032) es el siguiente:

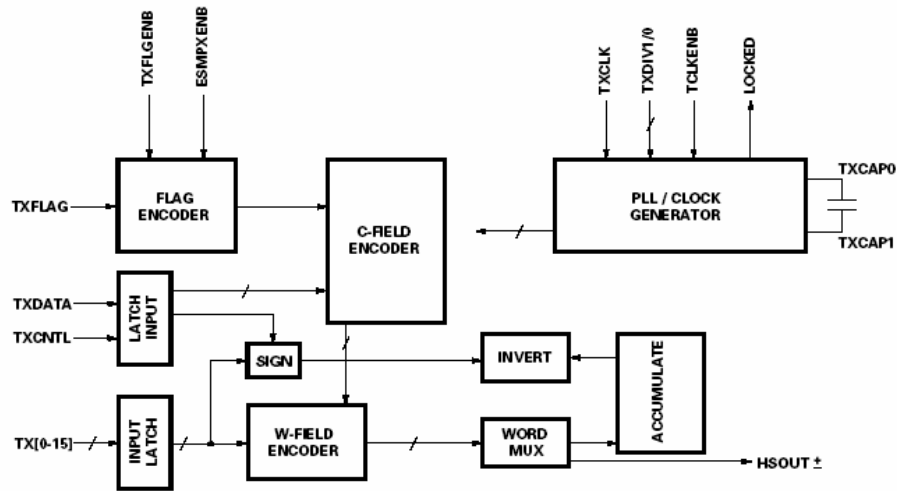


FIGURA 3.18 Diagrama de bloques interno del HDMP-1032 Tx.

Como se puede observar, este transmisor acepta los 16 bits de datos en paralelo (TX[0..15], queda configurado desde la FPGA (TXDIV1/0, TCLKENB), admite las señales de control del protocolo (TXFLAG, TXDATA, TXCNTL) y se coloca en fase con el reloj que se le envía (TXCLK).

Este reloj se engancha al PLL del bloque PLL/Clock Generator que genera todos los relojes internos necesarios para que el transmisor realice sus operaciones correctamente. Por otro lado, los datos y las señales de control pasan por la lógica de los bloques C-Field y W-Field Encoder Logic, donde se decide si el envío es de una palabra de control o de una palabra de datos.

Por último, el bloque Word Mux acepta los veinte bits totales que salen de los bloques anteriores y los multiplexa en datos serie que mantienen la velocidad del reloj definido al principio.

El diagrama de tiempos que sigue este dispositivo es:

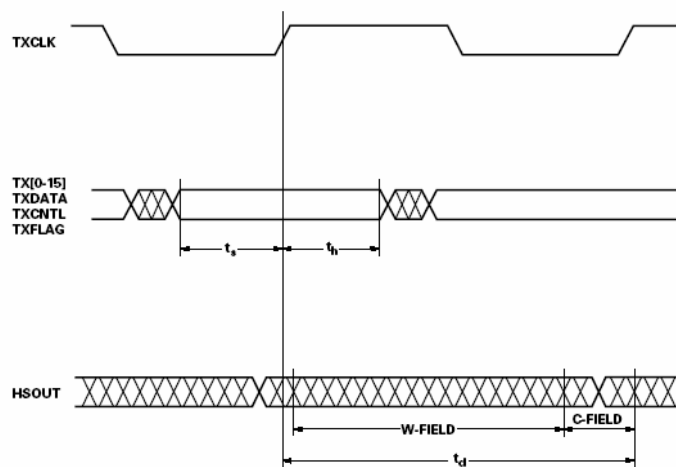


FIGURA 3.19 Diagrama de tiempos del HDMP-1032 Tx.

3.2.6 Generación y distribución de los relojes en la OMB

Dependiendo del modo de operación de la Optical Multiplexer Board, el reloj de la placa puede ser obtenido de dos formas. La primera forma como un Modo Local y la segunda forma como un Modo VME.

En el primer caso, el reloj se obtiene de los diferentes osciladores de cristal presentes en la placa. Tenemos uno por cada uno de los receptores (Rx) y otro rutado a las tres FPGAs que soportan los procesos del módulo. El tener un solo reloj para las tres FPGAs se debe a la necesidad de sincronización entre ambas. Una de nuestras reglas de rutado se ha centrado en que los relojes se distribuyeran uniformemente por la placa y que los retardos entre ellos fueran mínimos. Se estudiará con mayor detalle en puntos posteriores.

En el segundo caso, el reloj se obtiene por VME. No se trata del reloj de la Norma VME (16 MHz), sino de uno dedicado que hemos rutado para esta solución. Entra directamente en la FPGA de VME y es ella la que se encarga de rutarlo a las otras dos FPGAs. Los receptores, mantienen el reloj de placa para su funcionamiento. Se trata de un modo de trabajo de emergencia, lo lógico y correcto es funcionar en Modo Local. Si nos falla el reloj de placa, siempre tendremos esta posibilidad con este Modo VME.

El reloj que hemos usado es el DS1073 de Maxim [13], cuyas características principales son:

- Frecuencia de 40 MHz.
- Tolerancia inicial de $\pm 0.5\%$.
- Variación de temperatura y voltaje de $\pm 1\%$.
- Ciclo de Trabajo entre 40 y 60%.

El diagrama de tiempos que define su funcionamiento es el siguiente:

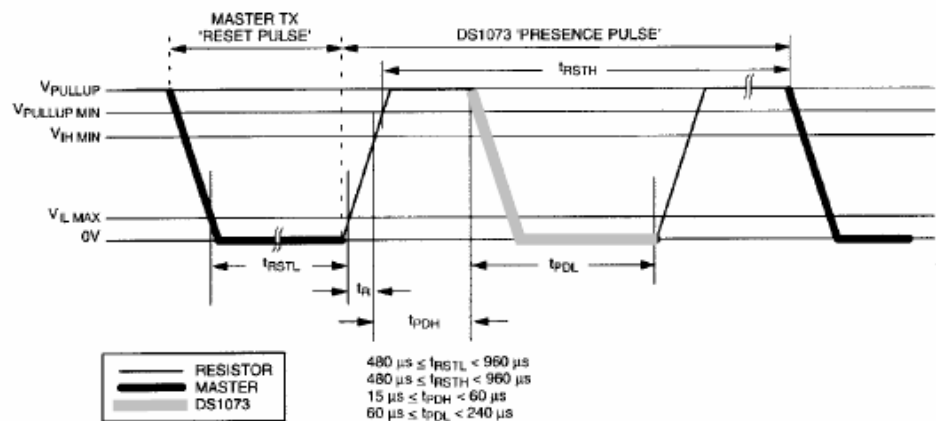


FIGURA 3.20 Diagrama de tiempos del reloj.

Según nuestras necesidades de diseño, transcurrido un tiempo inicial donde se realiza un proceso de inicialización del reloj, los tiempos de subida y bajada se ajustan a los valores predefinidos por nosotros.

El diagrama de bloques siguiente nos muestra la generación de los relojes y su distribución en Modo Local.

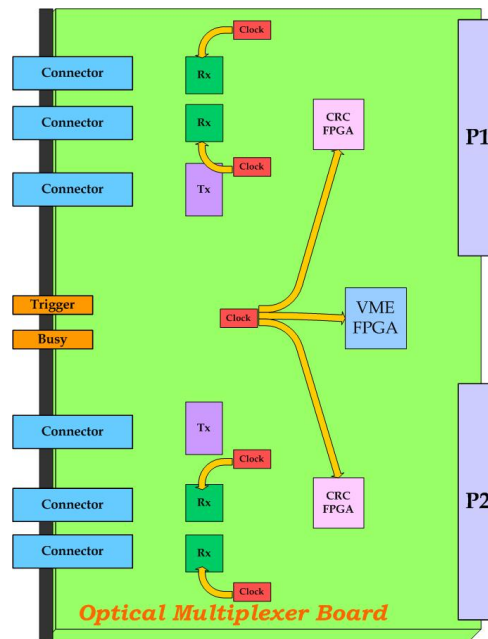


FIGURA 3.21 Diagrama de bloques de la distribución del reloj en Modo Local.

En el caso de usar el Modo VME, la generación y distribución del reloj sigue este diagrama:

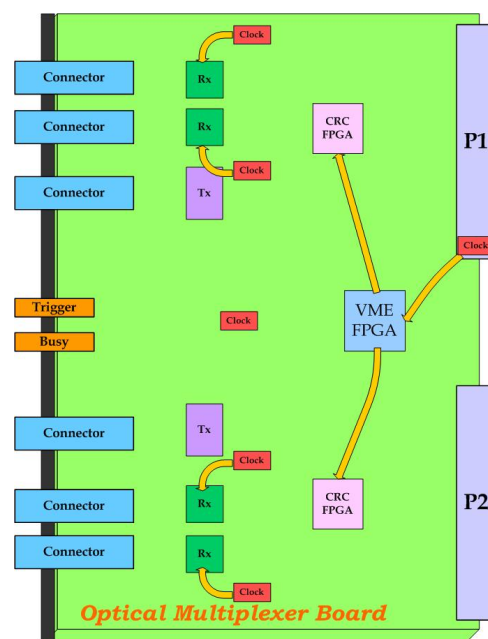


FIGURA 3.22 Diagrama de bloques de la distribución del reloj en Modo VME.

3.2.7 Distribución de los datos en la OMB

Como se ha comentado anteriormente, la Optical Multiplexer Board fue pensada inicialmente como multiplexora de los datos redundantes que envía el Front End al ROD. Una vez comenzado el desarrollo de la misma, y en paralelo al desarrollo de la Motherboard del ROD, se vio la posibilidad y necesidad de que tuviera una segunda función.

Esta segunda función nos permite inyectar datos al ROD sin depender del Front End, la Optical Multiplexer Board actúa como un Front End virtual. Internamente, en la FPGA de CRC, se generan datos conocidos que son enviados a la Motherboard del ROD.

Por tanto, tenemos dos modos de funcionamiento, el Modo Multiplexor y el Modo Inyector. La distribución de los datos según el criterio de funcionamiento sigue estos dos diagramas de bloques. En puntos siguientes veremos en profundidad ambos modos.

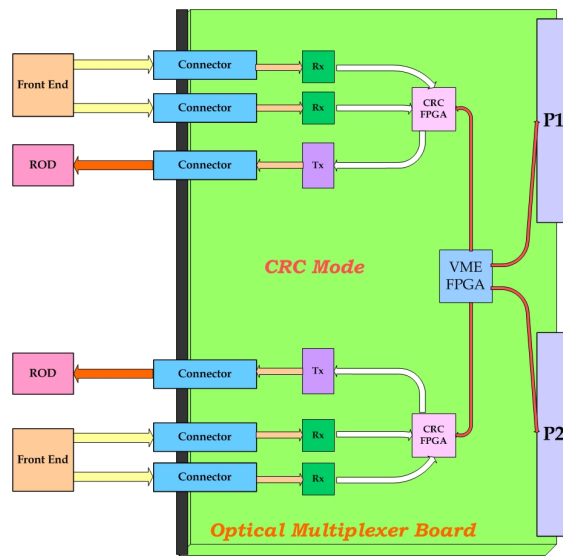


FIGURA 3.23 Flujo de los datos en Modo Multiplexor.

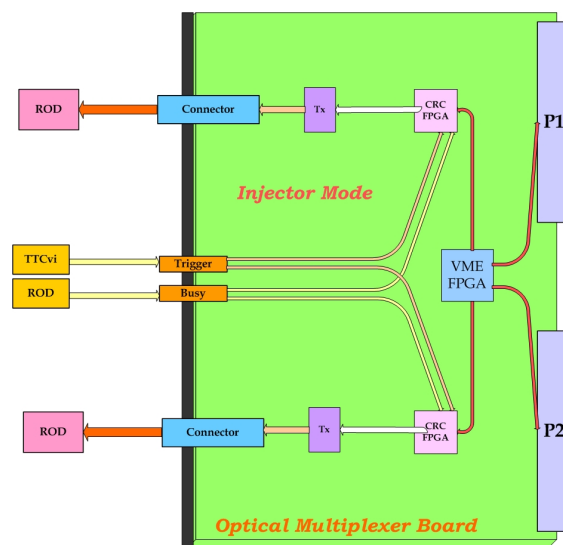


FIGURA 3.24 Flujo de los datos en Modo Inyector.

3.2.8 Distribución de la alimentación en la OMB

Tenemos dos formas de aplicar la alimentación principal (3,3 V) a la tarjeta:

- Por medio de los pines de conexión VME.
- Por medio de la conexión a alimentación externa de 5 V y su posterior conversión a 3,3 V mediante el convertor DC/DC (PT5801) incorporado en la tarjeta.

La alimentación de 5 V se usa para los componentes de trigger y busy, de ella se obtienen mediante otro convertor DC/DC (ADP3330-2.5) los 2,5 V que necesitamos para la FPGA de VME.

Las demás alimentaciones presentes en la placa son las de 12 V, -12 V y -5V. Todas ellas pueden ser tomadas desde VME si la tarjeta está incorporada en la crate o desde alimentación externa si la tarjeta está presente en un banco de pruebas externo.

A su vez, se han colocado dos portafusibles que nos permiten decidir desde donde debe tomar la alimentación nuestra tarjeta. Es una medida de seguridad para posibles errores o alteraciones de la alimentación.

3.2.9 Conectores J1/P1 y J2/P2

Los conectores J1 y J2, también llamados P1 y P2, son conectores estándar de VME64x [8] con 160 pines. La distribución de las alimentaciones, del reloj y de los datos se distribuye entre ambos conectores. El pinout de los conectores se puede ver en las siguientes figuras:

VME64x P1 Connector					
Pin	Signal Name	Signal Name	Signal Name	Signal Name	Signal Name
	Row z	Row A	Row B	Row C	Row d
1	MPR	D00	BBSY*	D08	VPC
2	GND	D01	ECLR*	D09	GND
3	MCLK	D02	ACFAIL*	D10	+1V
4	GND	D03	BGOIN*	D11	+V2
5	MSD	D04	3G0OUT*	D12	RstU
6	GND	D05	BG1IN*	D13	-V1
7	MMD	D06	3G1OUT*	D14	-V2
8	GND	D07	BG2IN*	D15	RstU
9	MCTL	GND	3G2OUT*	GND	GAP*
10	GND	SYSCLK	BG3IN*	SYSFAIL*	GAD
11	RESP*	GND	3G3OUT*	BERR*	GA1
12	GND	DS1*	BR0*	SYSRES*	+3.3v
13	RsvBus	DS0*	BR1*	LWORD*	GA2*
14	GND	WRITE*	BR2*	AM5	+3.3V
15	RsvBus	GND	BR3*	A23	GA3*
16	GND	DTACK*	AM0	A22	+3.3V
17	RsvBus	GND	AM1	A21	GA4*
18	GND	AS*	AM2	A20	+3.3V
19	RsvBus	GND	AM3	A19	RsvBus
20	GND	IACK*	GND	A18	+3.3V
21	RsvBus	IACKIN*	SERCLK	A17	RsvBus
22	GND	IACKOUT*	SERDAT*	A16	+3.3V
23	RsvBus	AM4	GND	A15	RsvBus
24	GND	A07	IRQ7*	A14	+3.3V
25	RsvBus	A06	IRQ6*	A13	RsvBus
26	GND	A05	IRQ5*	A12	+3.3V
27	RsvBus	A04	IRQ4*	A11	LIA*
28	GND	A03	IRQ3*	A10	+3.3V
29	RsvBus	A02	IRQ2*	A09	LIO*
30	GND	A01	IRQ1*	A08	+3.3V
31	RsvBus	-12V	+5V Standby	+12V	GND
32	GND	+5V	+5v	+5V	VPC

L. Davis

VME64x P2 Connector					
Pin	Signal Name	Signal Name	Signal Name	Signal Name	Signal Name
	Row z	Row A	Row B	Row C	Row d
1	UstDef	UstDef	+5V	UstDef	UstDef
2	GNC	UstDef	GND	UstDef	UstDef
3	UstDef	UstDef	RETRY	UstDef	UstDef
4	GNC	UstDef	A24	UstDef	UstDef
5	UstDef	UstDef	A25	UstDef	UstDef
6	GNC	UstDef	A26	UstDef	UstDef
7	UstDef	UstDef	A27	UstDef	UstDef
8	GNC	UstDef	A28	UstDef	UstDef
9	UstDef	UstDef	A29	UstDef	UstDef
10	GNC	UstDef	A30	UstDef	UstDef
11	UstDef	UstDef	A31	UstDef	UstDef
12	GNC	UstDef	GND	UstDef	UstDef
13	UstDef	UstDef	+5V	UstDef	UstDef
14	GNC	UstDef	D16	UstDef	UstDef
15	UstDef	UstDef	D17	UstDef	UstDef
16	GNC	UstDef	D18	UstDef	UstDef
17	UstDef	UstDef	D19	UstDef	UstDef
18	GNC	UstDef	D20	UstDef	UstDef
19	UstDef	UstDef	D21	UstDef	UstDef
20	GNC	UstDef	D22	UstDef	UstDef
21	UstDef	UstDef	D23	UstDef	UstDef
22	GNC	UstDef	GND	UstDef	UstDef
23	UstDef	UstDef	D24	UstDef	UstDef
24	GNC	UstDef	D25	UstDef	UstDef
25	UstDef	UstDef	D26	UstDef	UstDef
26	GNC	UstDef	D27	UstDef	UstDef
27	UstDef	UstDef	D28	UstDef	UstDef
28	GNC	UstDef	D29	UstDef	UstDef
29	UstDef	UstDef	D30	UstDef	UstDef
30	GNC	UstDef	D31	UstDef	UstDef
31	UstDef	UstDef	GND	UstDef	GND
32	GNC	UstDef	+5v	UstDef	VPC

L. Davis

FIGURA 3.25 Señales de los conectores J1 y J2 según estándar VME64x.

3.2.10 Señales del Panel Frontal

Por último, definiremos los componentes que tenemos presentes en el Panel Frontal de la Optical Multiplexer Board. Tenemos seis grupos de LEDs que nos indican el estado de las conexiones de entrada y salida, cuatro para la recepción y dos para la transmisión. Cada uno de ellos con tres colores, indicando el verde un enlace correcto, el amarillo un error en los datos y el rojo un error en el enlace.

A su vez, tenemos dos grupos de LEDs para obtener información de la alimentación. Cada grupo con tres luces verdes que hacen referencia a cada una de las alimentaciones presentes en la tarjeta (2,5 V, 3,3 V, 5 V, -5 V, 12 V y -12 V).

Para las señales de Trigger y Busy externas, tenemos dos conectores LEMO en el Panel Frontal. Será ahí donde conectaremos nuestras señales mediante cable coaxial para obtener un Trigger que nos de sincronismo con todo el Sistema ROD y para enviar un Busy a la Motherboard del ROD en caso de no tener capacidad de cálculo.

Como final de desarrollo, y una vez explicados todos los componentes principales de la Optical Multiplexer Board, exponemos una fotografía de la misma. Se puede comparar fácilmente con los diagramas de bloques expuestos anteriormente, identificando cada uno de estos componentes.

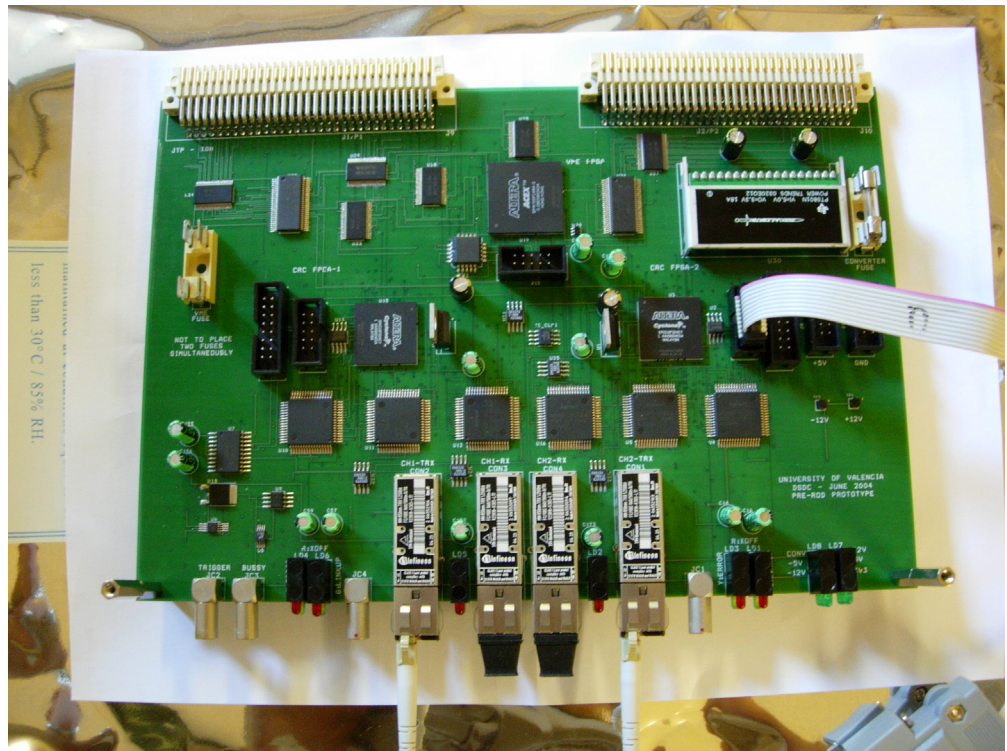


FIGURA 3.26 Fotografía de la Optical Multiplexer Board.

3.3 DESARROLLO Y COSTES DE LA OMB

La complejidad que presenta el desarrollo de esta tarjeta nos ha forzado a usar las últimas herramientas de diseño electrónico. El principal objetivo de las mismas es hacer la tarea del diseñador más fácil y conseguir circuitos más fiables. Dentro de todas las herramientas que hay en el mercado, nos decantamos por usar Cadence PSD en su versión 15.0 [14]. Entre otras cosas, su elección fue debida al conocimiento previo de la herramienta, lo cual esperabamos nos diera un punto a nuestro favor.

Dentro de todos los programas que incorpora esta herramienta, usamos para la captura del esquema inicial Capture CIS, el diseño de la placa de circuito impreso se ha realizado con Layout Plus, el rutado de la misma se ha desarrollado con Specetra y finalmente el análisis de la integridad de la señal se ha llevado a cabo con SpecetraQuest.

El objetivo de este punto, por tanto, es explicar el desarrollo real de la tarjeta. Hasta ahora hemos conocido el interés de la misma, los diagramas de bloques de su funcionamiento y los componentes principales que la componen. Es por tanto el momento de conocer cómo se diseñó realmente, cómo se emplazaron todos y cada uno de los componentes, el rutado de los mismos, el número de capas que la componen, los análisis realizados antes y después del rutado...

Para una exposición coherente, el orden de este punto pasa por ser el orden natural que se siguió en el desarrollo práctico y real de la Optical Multiplexer Board. Estamos acostumbrados a ver las hipótesis previas del trabajo, los desarrollos software y por fin los resultados obtenidos. En esta Tesis se pretende además dar una visión de cómo se realizó el hardware, qué herramientas se usaron, cuáles fueron los principales problemas que se tuvieron y qué coste tuvo todo este proceso.

3.3.1 Realización de los esquemáticos

Al tratarse de un diseño tan complejo y amplio, la realización de los esquemáticos se hizo siguiendo una estructura jerárquica vertical [15]. Esto es, se creó un diagrama de bloques general de cada una de las partes fundamentales de la tarjeta (bloques jerárquicos) que interiormente contienen los esquemas referentes a dicho bloque.

La estructura del diseño, por tanto, presenta varios niveles. Así mismo, existe una correspondencia entre los bloques jerárquicos y los esquemas a los que hacen referencia.

Existen dos métodos para crear diseños jerárquicos:

- Método Top-Down: se crean uno o más bloques y se cablean, pudiendo establecerse un diagrama de bloques funcional, que puede ser utilizado como el escalón más alto del diseño. Una vez hecho esto, hay que introducirse en cada bloque y dibujar el esquema.
- Método Bottom-Up: si un esquema resulta interesante para utilizar en otros diseños, se puede crear un símbolo jerárquico que represente el esquema. Este símbolo puede ser conectado eléctricamente en otro diseño.

Los bloques jerárquicos aparecen en el esquema como un rectángulo con dos o más puertos de entrada y salida, representan un circuito en forma de uno o más esquemas de bajo nivel, pudiendo colocarse más de un bloque en un esquema.

Los cables y buses que terminan en los límites del bloque quedan conectados a él automáticamente, creando de este modo los pines jerárquicos.

En nuestro caso, hemos usado el método Top-Down. En las figuras 3.27 y 3.28 se pueden observar los diferentes niveles usados. Con un mayor detalle, se pueden observar todos los esquemáticos de la tarjeta en el CD Anexo que se adjunta.

La figura 3.29 muestra cómo está estructurado nuestro diseño jerárquico.

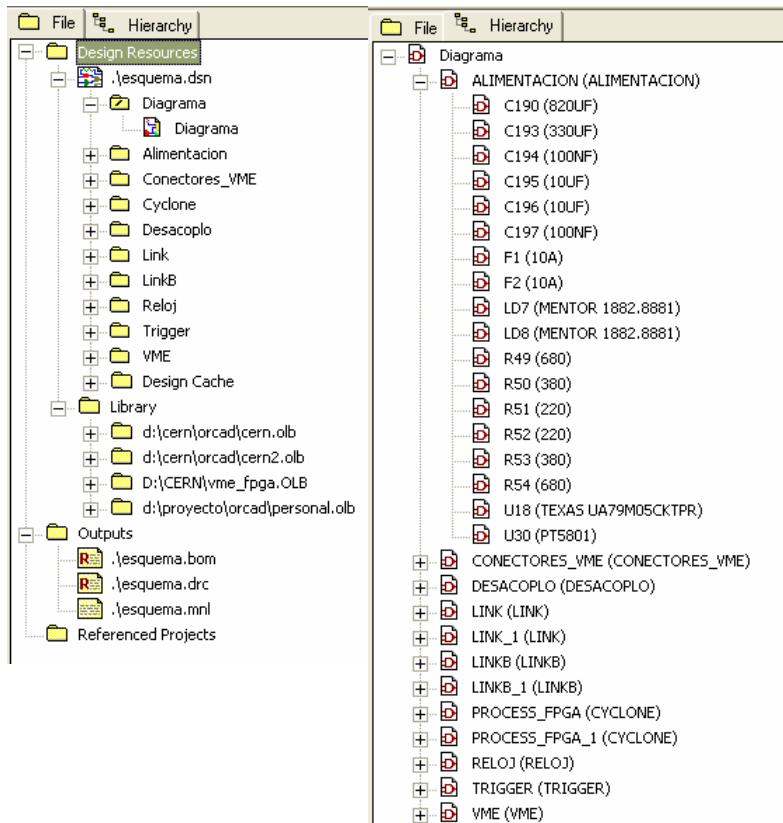


FIGURA 3.29 Desarrollo del diseño jerárquico de la OMB.

Se pueden observar los diferentes niveles que tenemos en el diseño. Dos cosas importantes, la primera la definición del diseño desde el nivel superior (Diagrama) hasta los componentes de cada una de las partes, la segunda la necesidad que tuvimos de crearnos librerías propias de componentes al no existir en las que nos ofrece el programa de diseño.

La creación de una nueva librería con los componentes que necesitábamos nos llevó a la creación de los mismos. Para ello, dibujamos los gráficos, colocamos los pines y definimos los atributos según especificaciones del fabricante.

Una vez terminado el desarrollo de los esquemas, el siguiente paso antes de la realización del circuito impreso es la especificación de los encapsulados de cada uno de los componentes presentes en el diseño. Así mismo, se debe verificar que el diseño capturado está conectado correctamente, en definitiva, se revisan los esquemas con el fin de comprobar el cumplimiento de unas reglas básicas de diseño.

Se trata de uno de los pasos más importantes del diseño que siempre se debe realizar antes de crear una lista de conexiones con el fin de no tener errores. Una vez se ha terminado con estas comprobaciones, estamos preparados para generar un fichero con la lista de conexiones que nos permite intercambiar la información con el programa desde donde se realice el circuito impreso (en nuestro caso Layout).

3.3.2 Diseño del circuito impreso

El llamado circuito impreso, placa de circuito impreso o simplemente placa (PCB en inglés) es, básicamente, un soporte para los circuitos electrónicos. Consta de un material base, aislante, sobre el que se disponen pistas conductoras, generalmente de cobre, que conforman el conexionado entre los distintos componentes.

En esta tarjeta, debido a la complejidad de la misma, se ha necesitado usar la tecnología multicapa en la cual se generan pares de caras que se van superponiendo unas a otras y que quedan conectadas a través de vías que atraviesan toda la placa. El resultado es una placa con una densidad muy alta de pistas donde los componentes se han colocado, evidentemente, en las dos capas externas.

En las PCBs más que hablar de componentes se debe hablar de encapsulados. Su objetivo es proteger el componente, darle rigidez mecánica y disipar el calor generado en su interior.

Para el diseño de la PCB hemos partido de los esquemáticos generados anteriormente y de las librerías creadas para los componentes no presentes por defecto.

Las siguientes fases de diseño que se realizaron en Layout se pueden resumir en:

- Fijación del contorno exterior de la tarjeta.
- Inserción de los componentes en la placa.
- Definición del orden y número de capas para el trazado.
- Definición de los tipos de nodos y las vías (forma y tamaño).
- Definición de las capas usadas como planos de masa y alimentación.

En la siguiente figura podemos ver cómo quedó definida la tarjeta.

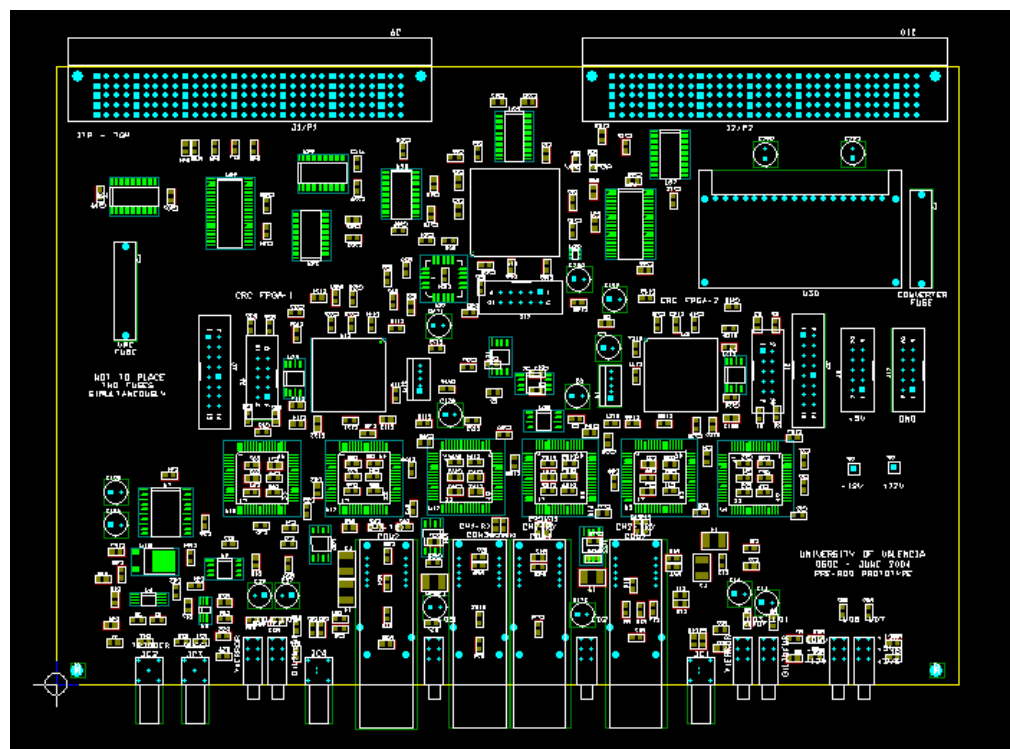


FIGURA 3.30 Definición de los componentes en la placa.

Una vez realizados estos pasos, llegamos al momento de la conexión física de todos los componentes. Es lo que se conoce como rutado de la placa. Éste, no presenta demasiadas dificultades si el número de conexiones es pequeño. En nuestro caso, al estar trabajando con FPGAs con un alto número de pines, la tarea se convirtió en un punto crítico del desarrollo.

Además, al tratarse de un diseño de alta velocidad, debemos tener presentes una serie de preparativos antes de comenzar con el rutado de las pistas. Debemos especificar qué capas serán planos de masa y alimentación, tener en cuenta la constante dieléctrica del sustrato a utilizar y el espacio entre capas.

Las capas de alimentación y masa son las primeras que hay que decidir. Para ello es necesario determinar los tiempos de subida de las señales, el número de señales y las dimensiones físicas de la tarjeta. Atendiendo a este último dato, se puede estimar el ancho de las pistas a rutar.

Otro aspecto muy importante a tener en cuenta antes del rutado de la tarjeta es la distribución de la señal de reloj en la tarjeta. Debemos saber que la señal de reloj es la que más rápidamente conmuta en un sistema digital. Para cada transición en las líneas de datos, existirá una señal de reloj que conmuta dos veces como mínimo.

Por estas dos características, carga y velocidad de conmutación, las señales de reloj merecen especial atención. En nuestro caso, nos decidimos por colocar un reloj para cada uno de los receptores de datos y otro reloj para las tres FPGAs presentes en el diseño.

La intención era que cada receptor trabajara con un reloj independiente que le diera solidez, permitiendo además un rutado más preciso y directo. Esto nos evitaría problemas de retardo difícilmente solucionables una vez rutada la tarjeta.

Por otra parte, y en previsión de una programación en las FPGAs que incluyera la necesidad de una sincronización entre ellas, se decidió colocar un único reloj para las tres. Para ello necesitamos de un buffer que distribuyera correctamente la señal, imponiendo en el rutado la condición de igual distancia de rutado para evitar retardos entre ellas.

Otro punto crítico e importante fueron las señales diferenciales que se distribuyen desde el conector de fibra óptica hasta los receptores y transmisores. Estas señales fueron tratadas para el rutado de una forma especial, fueron definidas como lo que son (señales diferenciales) y su trazado fue realizado en paralelo y con la mínima distancia entre componentes.

Por último, ya antes de entrar de lleno en la descripción del rutado, hablaremos de los buses de datos. Para evitar posibles pérdidas de datos o retardos significativos, su rutado fue definido manteniendo la misma distancia para cualquier línea del bus y por la misma capa.

Por tanto, los buses son tratados como pares de líneas diferenciales pero con la restricción de usar la misma capa para su trazado. Eso nos da la seguridad de tener unos datos correctos y sin tiempos de propagación diferentes.

Así pues, y una vez decidida la estructura de la tarjeta, se intentó desarrollar el rutado con Layout Plus [14]. Después de diversas pruebas y probando los diferentes algoritmos presentes en este programa para realizar el rutado, no conseguíamos que se rutara completamente. Era imposible terminar el proceso de conexión física de los componentes.

Nuestro siguiente paso fue trabajar con un programa más potente que nos permitiera realizar el rutado manteniendo las características y condiciones impuestas en Layout Plus. Nos decidimos por realizar el rutado con Spectra [16].

3.3.3 Rutado del circuito impreso

Como hemos comentado, debido a la complejidad del diseño el rutado se realizó con un programa de altas prestaciones como es Spectra. Este programa se puede encontrar dentro del paquete de aplicaciones de Cadence PSD.

El rutado lo basa en un algoritmo multipaso para la reducción de conflictos, hasta encontrar una solución natural al camino de las vías. Se trata de un algoritmo adaptativo con el que se obtienen altos rendimientos de rutado.

El resultado de rutar la placa con Spectra fue el siguiente:

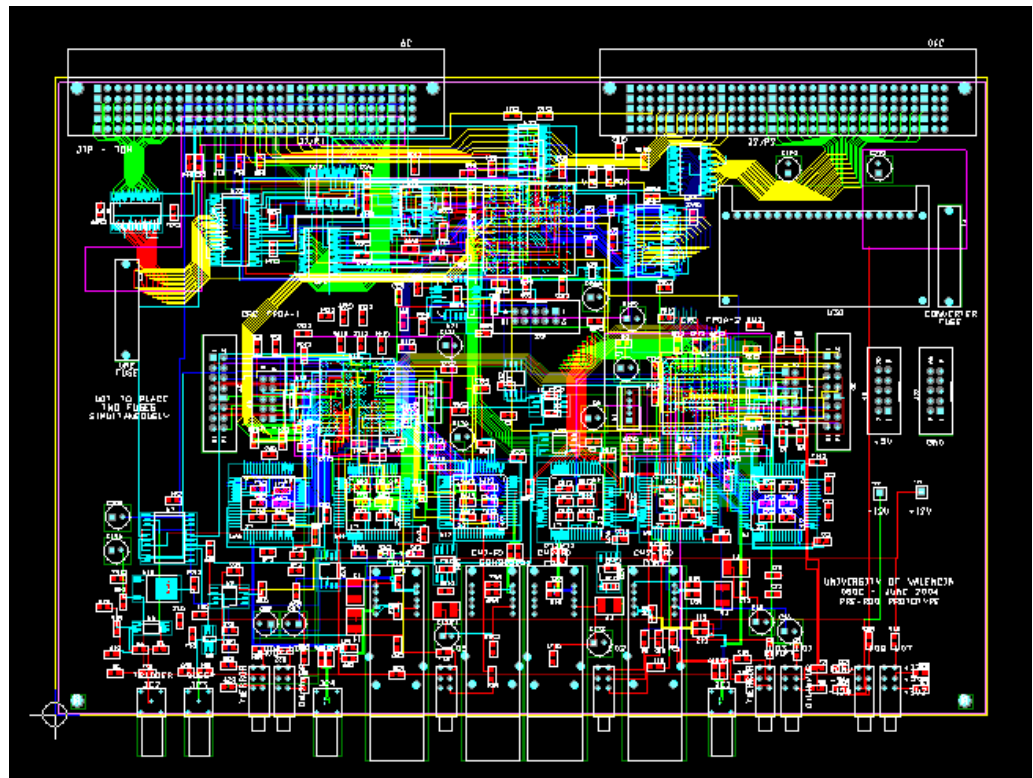


FIGURA 3.31 Rutado definitivo con Spectra.

Previamente al rutado final de la tarjeta, se realizó un estudio pre-layout sobre integridad de la señal que fue tenido en cuenta para su desarrollo. Tanto para este estudio como para el estudio post-layout de integridad de la señal, se ha dejado un punto aparte al final del capítulo.

Como ya hemos comentado anteriormente, la tarjeta está formada por una placa multicapa. La distribución de las capas es la siguiente, atendiendo a criterios ordenados de distribución de planos de masa, alimentación y rutado [17].

Esta distribución corresponde a la conveniencia de que cada capa de trazado de señales sea adyacente a un plano de masa o a un plano de alimentación. Así pues, la capa TOP es la más adecuada para el trazado de pistas sensibles o de más alta velocidad. Por otra parte, la capa BOTTOM ha sido usada para la colocación de los componentes pasivos en SMD (condensadores de desacoplo, resistencias y bobinas).

La disposición final de las capas es la siguiente.

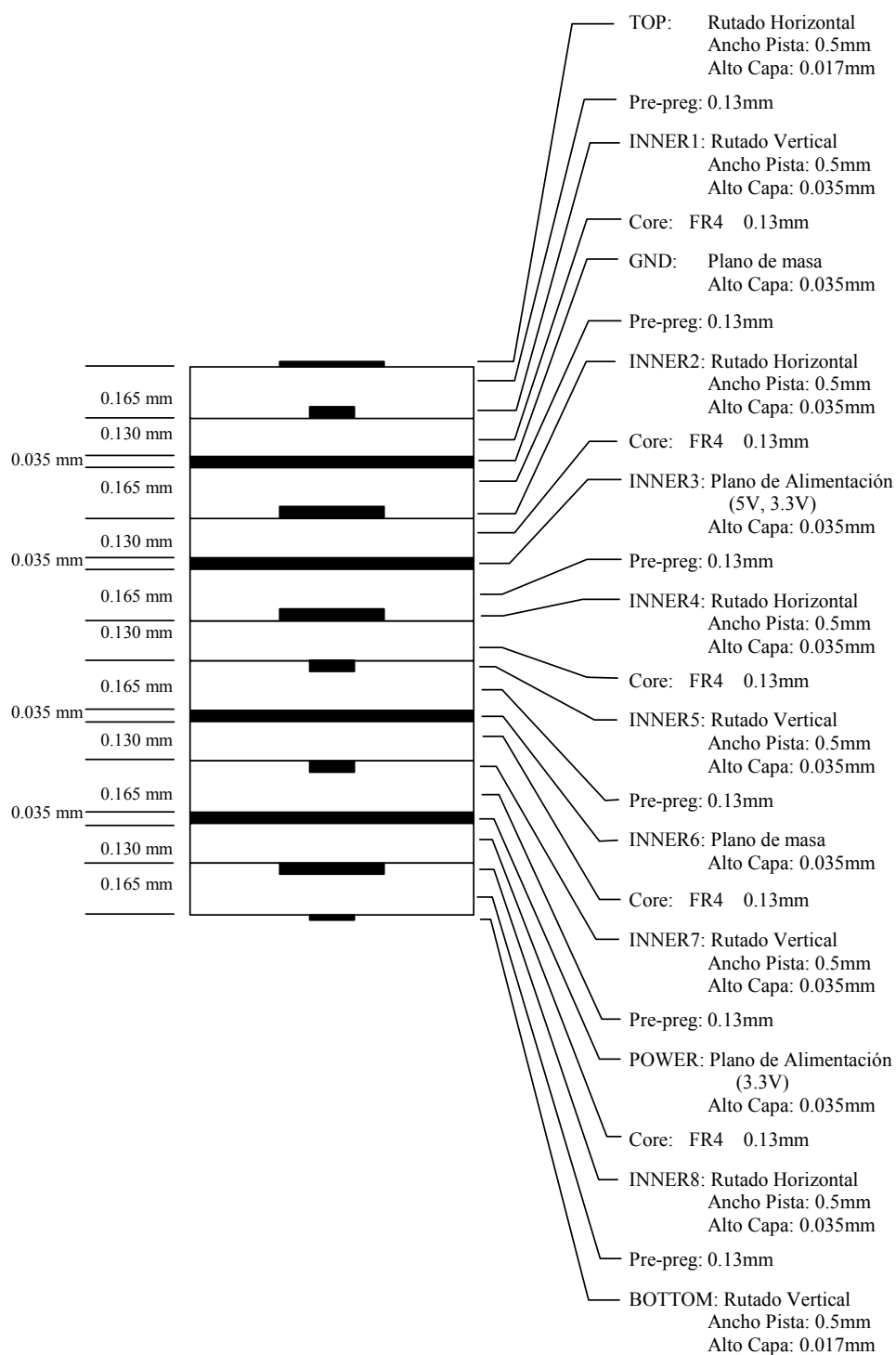


FIGURA 3.32 Distribución de las capas en la OMB.

Como se puede observar, las capas TOP, INNER1, INNER2, INNER4, INNER5, INNER7, INNER8 y BOTTOM se utilizan para rutado de pistas de componentes, mientras que las capas GND, POWER, INNER3 e INNER6 se han reservado para las alimentaciones y masas de toda la placa. Con ello conseguimos un buen aislamiento entre capas de rutado y capas de alimentación y masa, intentando conseguir el mínimo de problemas en cuanto a ruido e integridad de la señal se refiere.

La distribución de las alimentaciones es un aspecto muy importante, teniendo en cuenta la cantidad de alimentaciones distintas presentes en la placa. Hoy en día, en este tipo de tarjetas de alta velocidad, es fundamental asignar capas completas a las alimentaciones de referencia y a los planos de masa. De esta forma, se proporcionan planos de referencia sólidos para las señales, eliminando la alta inductancia inherente a las pistas de alimentación estrechas.

Más aún, se dota a las señales de un buen blindaje electromagnético procedente del exterior. Para ello, en la capa Power colocamos la alimentación principal de toda la placa, es decir, los 3,3 V que hacen falta distribuirse por todas las zonas de la tarjeta.

En la capa Inner3 están el resto de alimentaciones, así pues, tenemos los 2,5 V para la Acex (VME FPGA), los 1,5 V para las Cyclone (CRC FPGA), los 5V (tanto para la lógica analógica como para la entrada del convertor) y los 3,3V (tanto de la salida del convertor, como de la entrada de 3,3V de bus VME).

En la siguiente figura se puede observar con mayor detalle esta distribución.

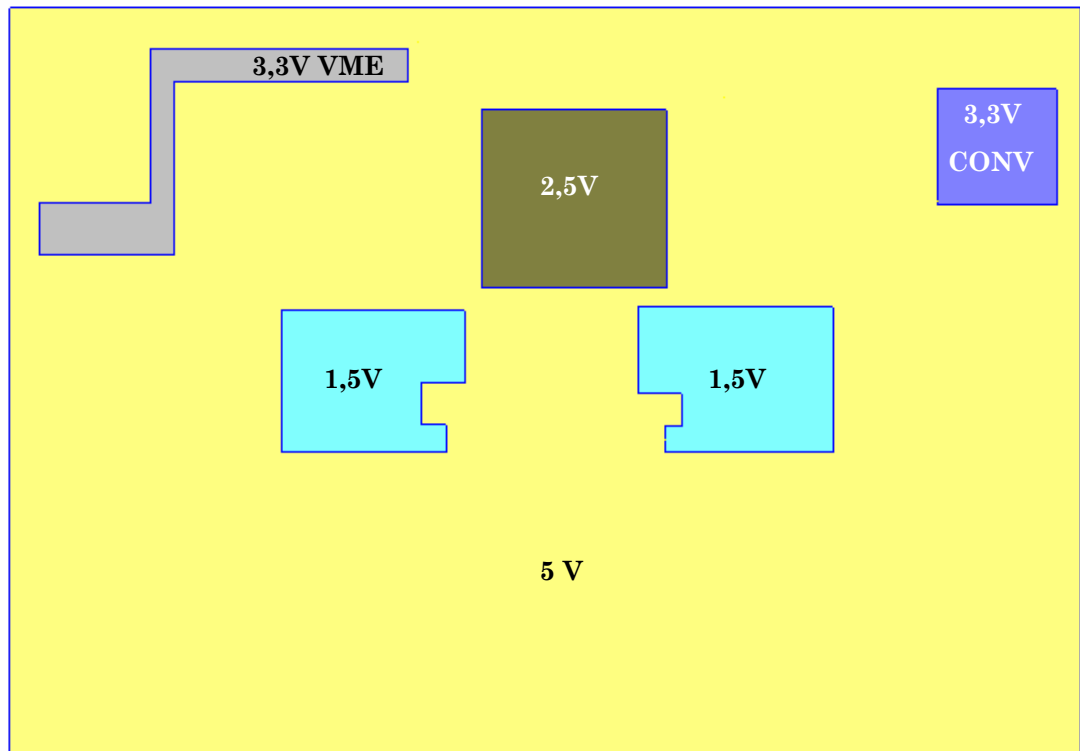


FIGURA 3.33 Distribución de la alimentación en la capa INNER3.

Las otras alimentaciones utilizadas (-5V y $\pm 12V$) no tienen plano de alimentación debido a su bajo consumo, por lo que se ha optado por usar únicamente pistas para su rutado.

Por otra parte, tanto la capa INNER6 como la capa GND son capas íntegramente de masa.

Hay que tener en cuenta que los planos de masa y alimentación en los sistemas digitales de alta velocidad son muy importantes. Realizan tres funciones críticas, una la de proporcionar tensiones de referencia estables para las señales digitales cambiantes, otra la de distribuir la energía a todos los dispositivos lógicos y la última la de controlar el acoplamiento entre señales.

3.3.4 Costes de la OMB

Una vez definidas todas las etapas y tareas en la realización de la tarjeta, debemos dedicar un apartado al estudio de los costes reales de la misma. Quizás no sea necesario determinar el coste en horas de trabajo y sí el coste de los principales componentes del módulo, de la realización de la PCB y del montaje de los componentes.

La siguiente tabla nos ofrece una aproximación del coste total en la realización completa de la Optical Multiplexer Board.

Desarrollo	Coste
Componentes	~ 1000 €
Fabricación PCB	~ 300 € / PCB
Montaje	~ 1000 € / PCB
Coste Total	~ 2.300 € / PCB

TABLA 3.4 Principales Costes de la OMB.

Este coste total de la OMB es más caro que si fuera en producción debido a la realización de sólo cuatro tarjetas; los costes asociados, tanto de fabricación como de montaje, a la creación de los modelos hacen aumentar el precio para una serie pequeña.

Estos gastos sólo se pagan una vez, sea cual sea el número de tarjetas a fabricar y montar. Para la producción definitiva de esta tarjeta (32 módulos) los costes por cada una de ellas disminuirán considerablemente.

3.4 LA OMB COMO MULTIPLEXORA DE DATOS

Una vez terminado el desarrollo hardware de la Optical Multiplexer Board, es el momento de analizar la programación que se ha realizado para su funcionamiento. El primer paso es explicar cómo trabaja la FPGA de CRC [11] en el momento se necesita como Multiplexora de Datos.

La secuencia de este punto se hará siguiendo el diagrama de bloques que nos define cada una de las partes en las que está definida la programación realizada.

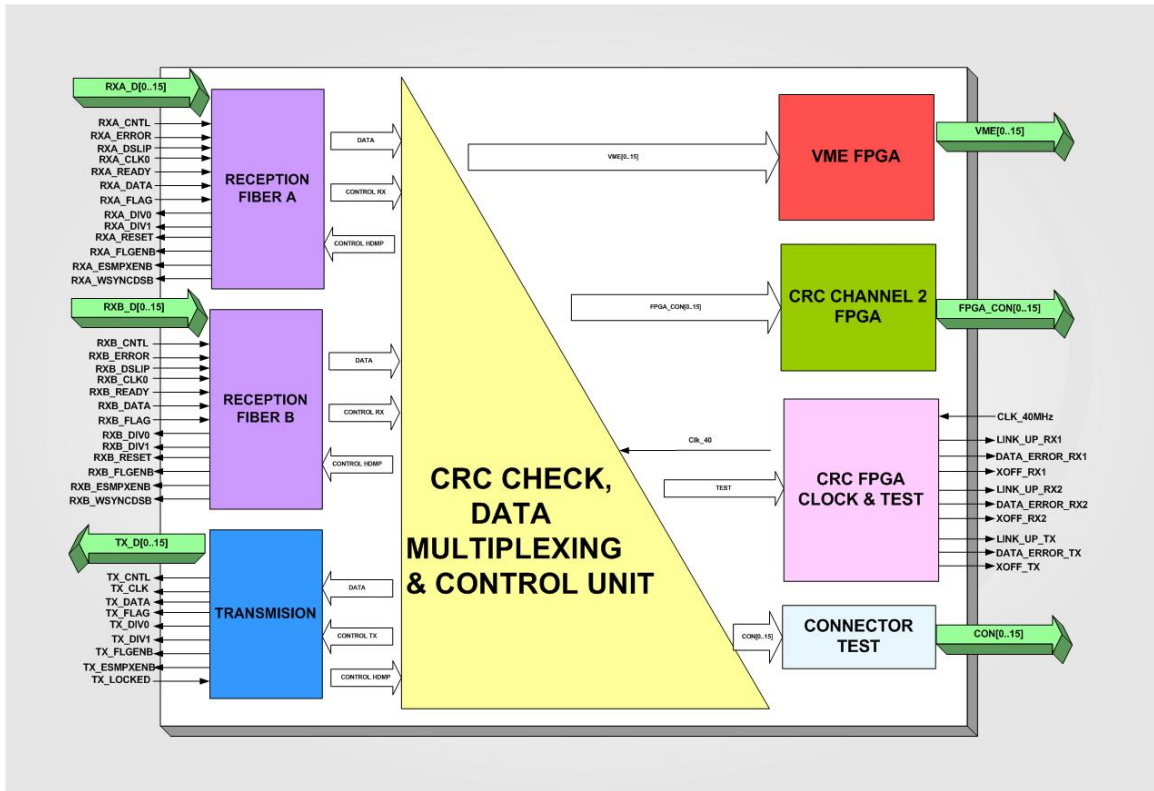


FIGURA 3.34 Diagrama de bloques de la FPGA de CRC en Modo Multiplexor.

Las características de cada uno de los bloques que componen este diseño se desarrolla a continuación, explicando en detalle cada una de las funciones internas que se realizan.

3.4.1 Bloque Reception Fiber A-B

Dedicado a controlar por parte del usuario el protocolo de comunicaciones G-Link entre la FPGA y el HDMP de Recepción [10]. Se pueden definir las características de lectura y de escritura de los datos que envía el HDMP a partir de las señales de control que tenemos aquí presentes.

Recibe, por supuesto, los datos que nos envía el Front-End y que son objeto de estudio y comprobación en esta FPGA. Los datos se reciben a una frecuencia de 40 MHz y son enviados mediante el reloj que genera internamente el HDMP.

Este bloque, por tanto, es el encargado de recibir los datos a tratar y de implementar el protocolo de recepción antes de que los datos sean analizados. Así mismo, controla el funcionamiento correcto del HDMP de Recepción.

Las señales de entrada-salida y sus funciones se describen en la siguiente tabla:

Señal	Tipo	Función
RX_FLGENB	Salida	Selecciona el uso de un bit extra para datos.
RX_ESMPXENB	Salida	Selecciona la compatibilidad con versiones anteriores de G-Link.
RX_DATA	Entrada	Indica el inicio de envío de una palabra de datos.
RX_CNTL	Entrada	Indica el inicio de envío de una palabra de control.
RX_D	Entrada	Datos de entrada.
RX_FLAG	Entrada	Valor del bit extra seleccionado
RX_READY	Entrada	Indica que el bloque de palabras a enviar no presenta error.
RX_ERROR	Entrada	Indica un error en los datos recibidos.
RX_DIV0/1	Salida	Programan el VCO del HDMP para trabajar a la frecuencia requerida.
RX_CLK0	Entrada	Reloj con el que son enviadas las señales de control y datos a la FPGA.
RX_DSLIP	Entrada	Indica que el retardo entre palabra y reloj es demasiado largo.
RX_RESET	Salida	Nos permite realizar un reset externo del HDMP.
RX_WSYNCD5B	Salida	Usado para testear la sincronización del HDMP.

TABLA 3.5 Descripción de las señales de entrada de la FPGA de CRC.

La correcta configuración y tratamiento de estas señales nos permite obtener un protocolo de entrada fiable, fundamental para un posterior procesado de los datos.

3.4.2 Bloque CRC Check, Data Multiplexing & Control Unit

Es el bloque principal de esta FPGA y de este modo de trabajo. En líneas generales su funcionamiento se resume en: la recepción de los datos, chequeo del CRC, elección de la fibra correcta, envío hacia el protocolo de salida y control de todos los bloques incluidos en la FPGA.

En comunicaciones el concepto fiabilidad está claramente aceptado como la forma de aportar a las conexiones extremo-extremo garantía plena de que la información que transfieren llega sin ningún error o, si aparecen errores, todos pueden ser detectados y corregidos. Aplicar a una red, a un protocolo, o a una tecnología de comunicaciones el calificativo de fiable implica que ésta puede aportar la garantía de transferencias libres de errores.

Para conseguir la fiabilidad total existen dos posibilidades: una consistente en aplicar un mecanismo de control en el que todos los tipos de datos transferidos son confirmados por el destino, con la intención de garantizar que no se pierde ni una sola unidad de transferencia; y la otra consistente en añadir información redundante a los paquetes de información que garantice a los receptores que esos paquetes no han sufrido ninguna variación en la red. En realidad, la fiabilidad se consigue uniendo estas dos técnicas, ya que la primera garantiza que no hay pérdidas de datos, y la segunda que los datos son correctos.

Eso es exactamente lo que se ha realizado en TileCal.

Así, se pueden usar códigos redundantes para la detección de errores y códigos redundantes para la corrección de esos errores. Para los primeros se usan los conocidos códigos de Hamming, mientras para los segundos se emplean los, también conocidos, códigos polinómicos o Códigos de Redundancia Cíclica (CRC) [18].

En nuestro caso, el polinomio generador de grado 16 usado en el CRC es de dos tipos:

- *CRC16*: $x^{16} + x^{15} + x^2 + 1$
- *CRC-CCITT16*: $x^{16} + x^{12} + x^5 + 1$

Es más que suficiente para considerar fiables los datos, y máxime cuando se conoce que en la fibra óptica, que es el medio físico de transmisión, los estudios realizados demuestran que el 99,64% de los errores son de un solo bit. Sabemos además que la probabilidad de que se produzca un error en un bit enviado a través de fibra es hoy inferior a 10^{-12} . Por tanto, la probabilidad de que aparezca un error de más de un bit está en torno a 10^{-15} .

A la vista de estos números podemos asumir que, aunque la fiabilidad no es absoluta, puede decirse que la probabilidad de recibir datos con errores debidos a la transmisión es nula.

Aún así, dentro del formato de los datos enviados se han incluido dos tipos de CRCs que deberemos calcular. El CRC-CCITT16 se calcula sobre el evento completo sin tener en cuenta las palabras de control y el otro es el CRC16 que se calcula para cada uno de los bloques representativos de un Demultiplexor.

El cálculo se realiza una vez nuestro programa detecta la palabra de control del formato de datos y la cabecera del primer bloque de datos. El cálculo se realiza en tiempo real, sin retrasos debidos al almacenamiento de los datos. Se opera según van llegando los datos hasta obtener el valor del CRC enviado, una vez obtenido se compara con el enviado y se señala como evento bueno o malo.

Una vez terminado el cálculo de todos los CRCs particulares y del CRC general, se verifica qué fibra presenta menor número de errores. Ésta, por tanto, será la que enviemos al ROD. Marcando en cada caso el bloque que, aun presentando errores, se envía.

La figura 3.35 nos muestra el formato de datos de entrada a la Optical Multiplexer Board.

Word nb.	Data word
1	START Control word (0x51115110)
2	Header from DMU1
3	Data word , high/low gain sample 1 from DMU1
4	Data word , high/low gain sample 2 from DMU1
5	Data word , high/low gain sample 3 from DMU1
6	Data word , high/low gain sample 4 from DMU1
7	Data word , high/low gain sample 5 from DMU1
8	Data word , high/low gain sample 6 from DMU1
9	Data word , high/low gain sample 7 from DMU1
10	CRC word from DMU1
11	Header from DMU2
12	Data word , high/low gain sample 1 from DMU2
13	Data word , high/low gain sample 2 from DMU2
14	Data word , high/low gain sample 3 from DMU2
15	Data word , high/low gain sample 4 from DMU2

16	Data word , high/low gain sample 5 from DMU2
17	Data word , high/low gain sample 6 from DMU2
18	Data word , high/low gain sample 7 from DMU2
19	CRC word from DMU2
.....
138	Header from DMU16
139	Data word , high/low gain sample 1 from DMU16
140	Data word , high/low gain sample 2 from DMU16
141	Data word , high/low gain sample 3 from DMU16
142	Data word , high/low gain sample 4 from DMU16
143	Data word , high/low gain sample 5 from DMU16
144	Data word , high/low gain sample 6 from DMU16
145	Data word , high/low gain sample 7 from DMU16
146	CRC word from DMU16
147	CRC from DIGITIZER
148	Final Link CRC16 (CRC16 of entire block in 16bits words)
149	END Control word (0xFFFFFFFF0)

FIGURA 3.35 Formato de los datos recibidos en la OMB.

3.4.3 Bloque Transmission

Una vez los datos han sido analizados y se ha escogido la fibra que no presenta errores o que presenta el menor número de ellos, estos son enviados al bloque de transmisión. Es aquí donde se unirán los datos a las señales de control y se montará el protocolo de salida de la FPGA hacia el serializador.

Se trata, por tanto, de montar la misma trama de datos que se han recibido y de enviarlos hacia el ROD con el protocolo correcto. Para ello, las señales que se deben generar en este bloque y que controlan la transmisión son las que se pueden ver en la siguiente tabla.

Igual que ocurría en el caso de recepción, es desde la FPGA desde donde se controla el HDMP de transmisión [10]. Es de vital importancia tener una señal de reloj adecuada que nos permita engancharnos correctamente con el PLL del transmisor.

Señal	Tipo	Función
TX_FLGENB	Salida	Selecciona el uso de un bit extra para datos.
TX_ESMPXENB	Salida	Selecciona la compatibilidad con versiones anteriores de G-Link.
TX_DATA	Salida	Indica el inicio de envío de una palabra de datos.
TX_CNTL	Salida	Indica el inicio de envío de una palabra de control.
TX_D	Salida	Datos de salida.
TX_FLAG	Salida	Valor del bit extra seleccionado
TX_LOCKED	Entrada	Indica que el reloj enviado se ha enganchado con el PLL del HDMP.
TX_DIV0/1	Salida	Programan el VCO del HDMP para trabajar a la frecuencia requerida.
TX_CLK	Salida	Reloj con el que son enviadas las señales de control y datos al HDMP.

TABLA 3.6 Descripción de las señales de salida de la FPGA de CRC.

3.4.4 Bloque VME FPGA

Este bloque comunica la FPGA de CRC con la FPGA de VME [12]. Su uso en el modo de funcionamiento del que estamos hablando se basa únicamente en marcar dentro de la FPGA que estamos en Modo CRC. La base de nuestro diseño es un único programa que aúna los dos modos de funcionamiento y gracias a este bloque sabremos identificarlo.

Mediante esta conexión con VME y desde cualquier PC que esté conectado al Sistema ROD podremos seleccionar si nuestra tarjeta debe trabajar en Modo Multiplexora o en Modo Inyectora. Este es el valor que tiene en este modo de funcionamiento. Más adelante veremos como en el Modo Inyectora, esta conexión presenta una mayor funcionalidad.

3.4.5 Bloque CRC Channel 2 FPGA

Comunica las dos FPGAs de CRC presentes en la OMB. Su función principal es la de testeo de los canales en caso de un mal funcionamiento de uno de ellos. Gracias a esta comunicación entre FPGAs tenemos una mayor versatilidad de trabajo en caso de errores hardware en la OMB.

Nos permite el envío de los 16 bits de datos de una FPGA a la otra. Una simple reprogramación de las dos FPGAs nos permitiría no perder un canal de datos. Eso sí, añadiría un retraso en el envío de los mismos hacia el ROD. Sería necesaria una FIFO que almacenara los eventos hasta que un canal ha sido enviado.

Así mismo, dentro del desarrollo de los tests de funcionalidad hardware, estas líneas nos ayudaron a comprobar que los errores eran distintos o iguales en ambas FPGAs. Fue una manera de acotar errores en el testeo de la OMB.

3.4.6 Bloque CRC FPGA Clock and Test

Por medio de este bloque recibimos el reloj que tenemos instalado en la OMB, dicho reloj nos sirve para sincronizar internamente los datos y las señales de control que permiten el cálculo del CRC.

En este mismo bloque tenemos las señales que nos dan el estado de cada uno de los canales de entrada y del canal de salida. Son las señales que envían su valor a los LEDs colocados en el panel frontal. Es una manera visual de conocer cómo está actuando cada canal y de la necesidad o no de actuar.

Para cada canal tenemos tres LEDs que nos ofrecen información de su estado.

- LINK_UP: Se enciende cuando el enlace físico es correcto.
- DATA_ERROR: Nos indica que existe algún error en los datos.
- XOFF: Da información de enlace roto o incorrecto.

3.4.7 Bloque Connector Test

Este es un bloque de test, pensado para poder sacar de la FPGA 16 líneas para su testeo. Nos permite conectar un Analizador de Lógico o un Osciloscopio a la OMB, asegurándonos un buen análisis de las señales que deseamos monitorizar.

Esta conexión fue algo que desde el momento en que trabajamos con el Módulo de Transición Activo explicado en el Capítulo anterior tuvimos claro en colocar. Es de vital importancia tener un medio de comunicación con el mundo exterior para analizar cualquier señal de la tarjeta. Nos da una versatilidad que sin él sería imposible conseguir.

3.5 LA OMB COMO INYECTORA DE DATOS

Al igual que ha ocurrido en el punto anterior, el diseño y funcionamiento de la Optical Multiplexer Board como Inyectora de Datos se hará siguiendo el diagrama de bloques siguiente.

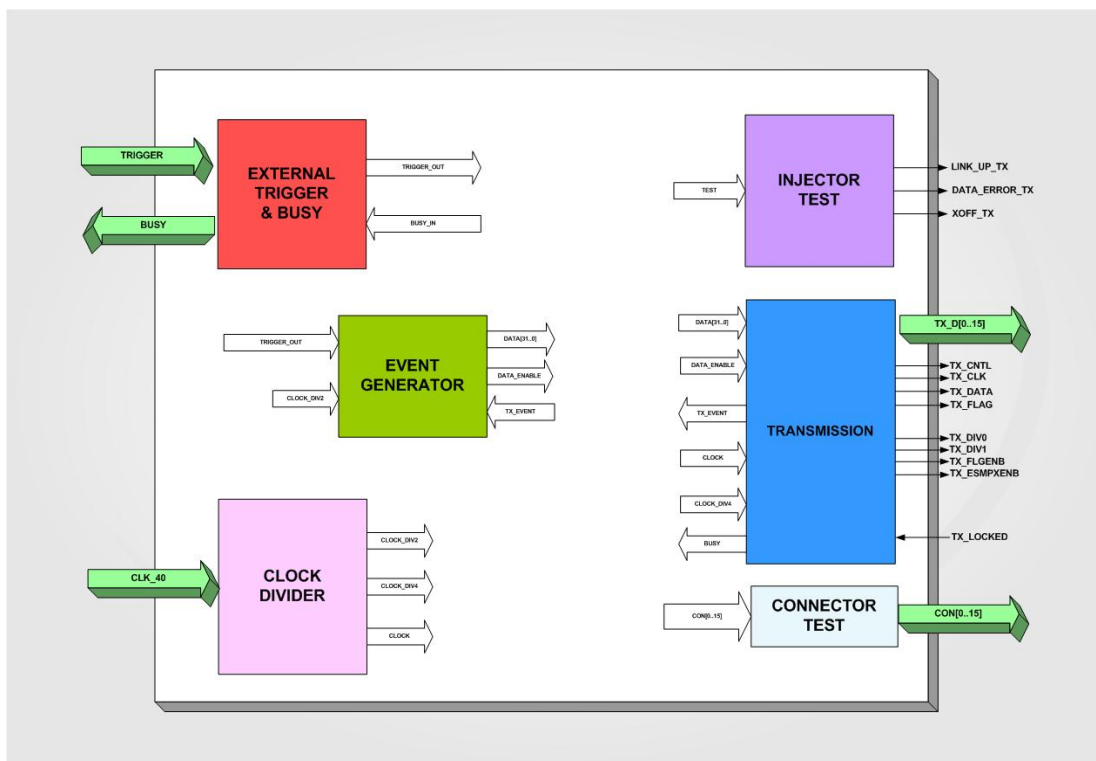


FIGURA 3.36 Diagrama de bloques de la FPGA de CRC en Modo Inyector.

El motivo de esta funcionalidad es la de simular la electrónica de Front-End [4] del calorímetro TileCal para el testeo de las Motherboards del Sistema ROD. La idea es utilizar la Optical Multiplexer Board como Inyectora de datos hacia el ROD sin la necesidad de estar dentro del Test Beam de TileCal.

Así pues, es necesario crear un programa que genere los datos tal y como ocurre en la realidad, enviando paquetes de datos con la misma frecuencia que se haría en TileCal. Para ello, explicaremos el funcionamiento de este modo a partir del diagrama de bloques anterior.

3.5.1 Bloque External Trigger & Busy

Dedicado a recibir la señal de Trigger externo que nos marcará el envío de los datos hacia el ROD. A su vez, envía una señal de Busy hacia el generador de Triggers en el momento nuestro sistema no sea capaz de seguir enviando datos o exista algún problema en la comunicación.

3.5.2 Bloque Clock Divider

Bloque encargado de la recepción del reloj de placa que nos sirve para generar los relojes internos necesarios tanto para la generación de los datos como para el control del protocolo de salida.

3.5.3 Bloque Event Generator

Es en este bloque donde a partir de recibir la señal de Trigger y la señal de reloj antes mencionadas se generan los bloques de datos que se enviarán como test al ROD.

Tenemos varias opciones para generar los datos:

- Una es la de enviar el mismo evento continuamente a partir de uno obtenido del Test Beam y almacenado en una RAM virtual interna. Esto nos permitiría aplicar en el ROD los algoritmos en tiempo real y observar sus resultados.
- Otra es enviar datos con el formato de TileCal pero generados a partir de un contador. No tendremos datos de Física pero tendremos un cambio constante de valor que nos permitirá realizar tests compactos.
- La última es la de generar datos de Física a partir de una función que genera valores de energía aleatorios. Esto nos permite simular el Front End de TileCal con valores aleatorios no conocidos.

Una vez generados los datos en cualquiera de sus formas, son enviados al siguiente bloque para su transmisión al ROD. Enviando previamente un data_enable para indicar su posterior envío. Los datos son enviados en bloques de 32 bits.

3.5.4 Bloque Transmission

Al igual que ocurría en el Modo Multiplexor, este bloque está preparado para generar las señales de control del transmisor y para el envío de los datos siguiendo el protocolo G-Link establecido.

Además, al recibir los datos en formato 32 bits, debe dividir estos en bloques de 16 bits para que el envío sea correcto. Se debe montar la trama manteniendo la estructura de datos de TileCal, las señales que controlan este proceso son las mismas que las del apartado 3.4.3 y se pueden observar en la tabla 3.5.

Una novedad con respecto al caso anterior es la necesidad de generar en este bloque el CRC de la trama a enviar y del paquete completo de datos. Aquí no recibimos datos con el CRC, los generamos nosotros y por tanto realizamos el proceso inverso al modo anteriormente explicado.

La generación del CRC se realiza siguiendo las mismas funciones utilizadas para su cálculo y explicadas con anterioridad.

3.5.5 Bloque Injector Test

Envía a los LEDs la señal que indica el momento en el que se encuentra el protocolo de comunicaciones.

3.5.6 Bloque Connector Test

Únicamente usado para realizar tests externos a la tarjeta.

3.6 PROTOCOLO VME EN LA OMB

La Optical Multiplexer Board es un módulo VME esclavo, todos los elementos de programación internos son controlados desde la CPU de VME instalada en el Sistema ROD. La comunicación entre la OMB y la CPU se realiza siguiendo el estándar VME64x [8].

La transferencia de datos se realiza mediante el bus de datos de 32 bits, el cual implementa ciclos de lectura/escritura y puede implementar transferencias de bloques de datos. El bus de direcciones es también de 32 bits para la transferencia de datos y de 24 bits para la configuración ROM y para los registros de control y estado (CR/CSR). El direccionamiento geográfico se estructura mediante 5 bits.

El siguiente diagrama de bloques nos muestra los diferentes elementos que se han programado en la FPGA de VME.

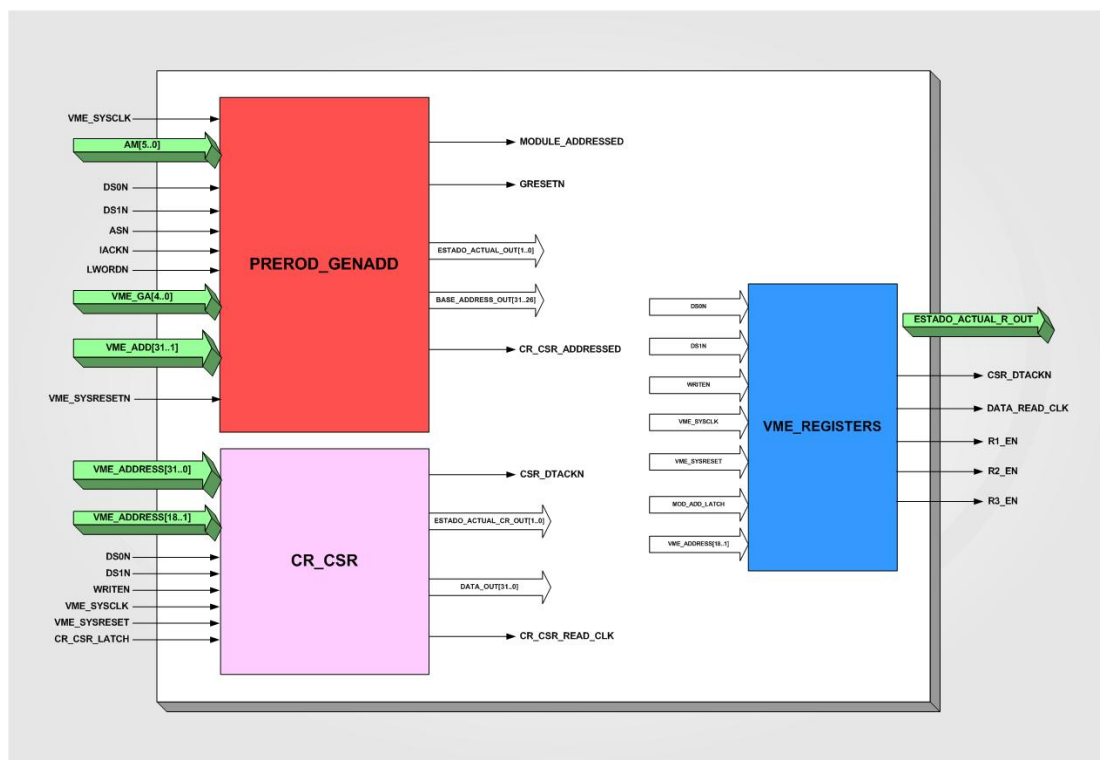


FIGURA 3.37 Diagrama de bloques de la FPGA de VME.

Cada bloque cumple las funcionalidades que se detallan a continuación.

3.6.1 Bloque Prerod_Genadd

Bloque destinado a determinar si se direcciona o no la Optical Multiplexer Board. Responde al ciclo de direccionamiento VME (líneas de dirección $vme_add[31..1]$, Address Modifier $am[5..0]$, $iackn$, $ds0n$ y $ds1n$, asn y $lwordn$) y activa las señales correspondientes en función de si se está accediendo al espacio CR_CSR ($CR_CSR_addressed$ y la correspondiente latcheada CR_CSR_latch) o al de registros generales ($module_addressed$ y la correspondiente latcheada mod_add_latch). Responde a direccionamientos geográficos (a través de las líneas $vme_ga[4..0]$) y normales (a través de las líneas de dirección $vme_add[31..24]$).

3.6.2 Bloque CR_CSR

Este bloque contiene los registros CR_CSR del OMB. El contenido de estos registros está determinado por la norma VME64 (ANSI/VITA 1-1994) y VME64X (ANSI/VITA 1.1-1997). Sus entradas son las líneas de dirección *vme_add[18..1]*, las de datos *data_in[31..0]*, *ds0n*, *ds1n*, la de identificación del ciclo como lectura o escritura *writen* y *CR_CSR_latch* que indica que el acceso es a un registro del espacio CR/CSR.

3.6.3 Bloque VME_Registers

Realiza la decodificación del mapa de memoria del OMB. En función de la parte baja de las direcciones (líneas 18 a 1) activa las señales particulares de lectura o escritura de los registros definidos en el mapa de memoria de la tarjeta (*RX_outen* o *RX_en*, respectivamente, donde X es el número de registro). El reloj para los registros es el de placa de 40 MHz. El bloque también activa las señales VME para terminación del ciclo tanto de lectura como de escritura. Las entradas de este bloque son las mismas que en el bloque CR_CSR (las líneas de dirección *vme_add[18..1]*, las de datos *data_in[31..0]*, *ds0n*, *ds1n*, la de identificación del ciclo como lectura o escritura *writen* y *mod_add_latch* que indica que el acceso es a un registro del espacio de registros).

El diseño también incluye la generación de la señal de reconocimiento del ciclo *dtackn* tanto para los accesos a CR/CSR como a los registros a partir de líneas de reconocimiento particulares *CSR_dtackn* y *reg_dtackn*.

El interface responde a ciclos single word de lectura y escritura con direccionamientos geográfico o normal. Para incluir otros ciclos, como block transfer o la posibilidad de producir y responder a ciclos de interrupción, se habría de modificar el bloque PREROD_GENADD.

La figura 3.38 nos muestra todas las funcionalidades que se han implementado para trabajar desde VME. Elección del modo de funcionamiento (CRC, INJ), tipo de trigger (EXT, VME), modo del trigger (SINGLE, LOOP, NON-STOP), reset de los contadores de CRC, test de los mismos y posibilidad de definir la frecuencia del trigger.

PREROD VME MEMORY MAP (A32 ACCESS)		
OFFSET	DESCRIPTION	TYPE
0x00	STATUS/CONTROL REGISTER	
b0	Link A mode (0 - CRC; 1 - Inj)	R/W
b1	Link A trigger (0 - Ext; 1 - VME)	R/W
b2-b3	Link A VME trigger mode	R/W
	01 - single	
	10 - loop	
	11 - non-stop	
b4	Link B mode (0 - CRC; 1 - Inj)	R/W
b5	Link B trigger (0 - Ext; 1 - VME)	R/W
b6-b7	Link B VME trigger mode	R/W
	01 - single	
	10 - loop	
	11 - non-stop	
b8	Reset Link A CH1 parity error counter	W
b9	Reset Link A CH1 CRC16 error counter	W
b10	Reset Link A CH1 CCITT-CRC16 error counter	W
b11	Reset Link A CH2 parity error counter	W
b12	Reset Link A CH2 CRC16 error counter	W
b13	Reset Link A CH2 CCITT-CRC16 error counter	W
b14	Reset Link B CH1 parity error counter	W
b15	Reset Link B CH1 CRC16 error counter	W
b16	Reset Link A CH1 CCITT-CRC16 error counter	W
b17	Reset Link A CH2 parity error counter	W
b18	Reset Link A CH2 CRC16 error counter	W
b19	Reset Link B CH2 CCITT-CRC16 error counter	W
b20	Reset all counters (1 - Reset)	W
b21-b31	Reserved for future use	-

0x10	LINK A CH1 PARITY ERRORS	R
0x14	LINK A CH1 CRC16 ERRORS	R
0x18	LINK A CH1 CCITT-CRC16 ERRORS	R
0x1C	LINK A CH2 PARITY ERRORS	R
0x20	LINK A CH2 CRC16 ERRORS	R
0x24	LINK A CH2 CCITT-CRC16 ERRORS	R
0x30	LINK B CH1 PARITY ERRORS	R
0x34	LINK B CH1 CRC16 ERRORS	R
0x38	LINK B CH1 CCITT-CRC16 ERRORS	R
0x3C	LINK B CH2 PARITY ERRORS	R
0x40	LINK B CH2 CRC16 ERRORS	R
0x44	LINK B CH2 CCITT-CRC16 ERRORS	R
0x50	LINK A VME TRIGGER FREQUENCY DIVIDER	R/W
0x54	LINK B VME TRIGGER FREQUENCY DIVIDER	R/W
0x58	LOOP LIMIT IN REPEAT VME TRIGGER MODE	R/W
	b31 - b16 LINK B LIMIT	
	b15 - b0 LINK A LIMIT	
0x100-0X1FF	INJECTION EVENT MEMORY	R/W

FIGURA 3.38 Registros de VME.

4. ESTUDIOS SOBRE INTEGRIDAD DE LA SEÑAL

Hemos dejado un apartado para el análisis de los estudios sobre integridad de la señal llevados a cabo en la Optical Multiplexer Board. Esto es debido a la importancia que adquieren o han adquirido en los últimos tiempos los fenómenos de alta frecuencia asociados al incremento en la velocidad de los buses. Hasta hace poco hablábamos de fenómenos de segundo o tercer orden mientras que ahora comienzan a serlo de primer orden.

Como consecuencia de esto, el diseño moderno de alta velocidad requiere que el ingeniero introduzca nuevos campos y que se fije en nuevas variables de desarrollo. Muchos diseños usaban el método rutado-simulación-rutado, esto para diseños pequeños podía ser factible. Se rutaba, se extraía y se simulaba. Cuando se encontraban errores de integridad de señal, el ingeniero de diseño determinaba las correcciones, el layout se modificaba y el proceso comenzaba de nuevo.

Hoy en día esto es inviable, los diseños cada vez son más complejos y más caros de producir. Por tanto, antes de rutar una placa, debemos seguir un procedimiento estructurado que nos asegure un layout trabajado y sin errores. En ese momento, podremos realizar la placa. Ese proceso previo es lo que se conoce como Pre-Layout [19].

Con este estudio de Pre-Layout, imponemos unas condiciones de rutado que se deben cumplir y que nos permiten tener la seguridad de que no aparecerán problemas de integridad de la señal en nuestra tarjeta.

Antes de analizar la OMB es necesaria una introducción a la problemática de la integridad de la señal. El siguiente apartado nos da una visión de este nuevo concepto.

4.1 FUNDAMENTOS DEL DISEÑO DIGITAL PARA ALTA VELOCIDAD

El diseño digital para alta velocidad se diferencia del de baja velocidad en el hecho principal de que hay que tener en cuenta algunos efectos considerados despreciables cuando se trabaja a baja frecuencia. Esta afirmación puede ser matizada por el hecho de que, como se verá, la necesidad de emplear técnicas especiales de diseño está más en función de las componentes de mayor frecuencia de las señales de interés que han de transmitirse obligatoriamente para conservar la integridad de la señal y no tanto en el valor de la frecuencia de la señal [20].

Así, por ejemplo, si debemos trabajar con señales de pulsos de frecuencia 1 KHz pero con tiempo de subida de 800 ps y deseamos, por el motivo que sea, mantener lo más fielmente posible la forma del pulso, no podemos diseñar nuestro sistema sin tener en cuenta los aspectos que se van a describir a lo largo de este punto. Este sería un caso extremo de un ejemplo donde baja frecuencia implica técnicas de alta velocidad. En realidad los sistemas digitales trabajan con señales donde un aumento de frecuencia implica una reducción de los tiempos de subida y bajada de la señal y por tanto lleva implícito un estudio como el que se va a ver de los aspectos que normalmente se desprecian a baja frecuencia. Entre estos aspectos cabe destacar los fenómenos producidos en los componentes pasivos, las conexiones, los acoplamientos, las terminaciones, las pistas del circuito impreso y el encapsulado de los circuitos integrados.

El estudio del diseño digital para alta velocidad es complejo y llega a tal punto que algún autor reconocido en el tema lo califica como “magia negra”. Esto es debido al gran número de parámetros implicados en el mismo. Por ello, muchas veces las expresiones y formulas empleadas carecen de una deducción formal o la misma es muy compleja y algunas se basan exclusivamente en medidas empíricas.

4.1.1 Revisión de los Conceptos de Frecuencia, Tiempo y Distancia

Vamos a empezar nuestro estudio repasando los conceptos de frecuencia, tiempo y distancia en lo que nos atañe en el diseño para alta velocidad.

En primer lugar, de sobra es conocido que, en general, los parámetros eléctricos dependen de la frecuencia. Cuando nos enfrentamos a un diseño a una frecuencia dada la cuestión que aparece es ¿cuál es la máxima frecuencia que nuestro sistema no debe modificar para no tener una distorsión en la señal demasiado importante?

Esta pregunta se puede responder realizando un sencillo análisis. Supongamos que tenemos un biestable D al cual se le inyecta una señal digital aleatoria y una señal de frecuencia f_{reloj} con un tiempo de subida del 1% de su periodo. Así, en cada periodo del reloj a la entrada del biestable tendremos un valor 1 ó 0 de forma aleatoria. Si representamos el espectro de la señal a la salida del biestable obtendremos algo como lo mostrado en la figura 3.39. Como se observa, el espectro presenta ceros en frecuencias múltiplo de la de la señal f_{reloj} y una caída de 20 dB/década desde f_{reloj} hasta una frecuencia que llamaremos frecuencia de codo f_{codo} donde la caída se acentúa.

Esta frecuencia de codo se relaciona con el tiempo de subida o bajada de la señal (el menor de los dos) de forma que:

$$f_{codo} = \frac{0.5}{t_r} \quad (\text{Ec. 4.1})$$

donde t_r es el tiempo de subida (o bajada) computado desde el 10% al 90% del valor máximo de la señal.

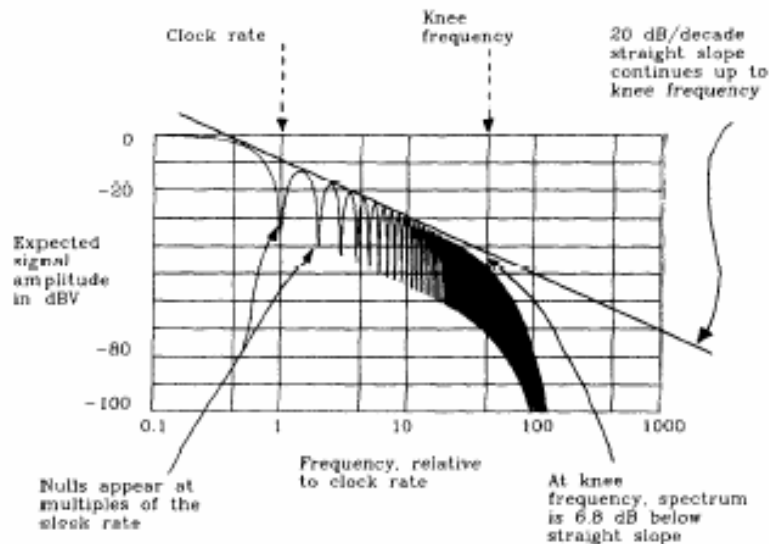


FIGURA 3.39 Espectro de una señal digital aleatoria.

Podemos entonces afirmar que las características importantes en el dominio del tiempo de cualquier señal digital vienen determinadas por el contenido espectral por debajo de f_{codo} puesto que por encima de él su amplitud es rápidamente atenuada.

Desde el punto de vista de la integridad de la señal podemos concluir respecto de la frecuencia en que:

- Cualquier circuito con respuesta plana hasta f_{codo} inclusive no distorsionará prácticamente la señal.
- El comportamiento del circuito por encima de f_{reloj} no influirá demasiado en cómo se procesa la señal.

Hay que comentar que f_{codo} sirve como método aproximado para conocer si ciertos efectos debidos a altas frecuencias van a afectar o no a nuestra señal y por lo tanto deben ser corregidos o minimizados. Si se desea conocer el comportamiento exacto del circuito habrá que realizar un análisis de Fourier completo.

¿Qué ocurre si nuestro circuito no tiene una respuesta plana hasta f_{codo} ? Evidentemente la señal se distorsionará. Veamos cómo con un ejemplo sencillo.

Imaginemos un circuito formado por un condensador y una resistencia como el mostrado en la figura 3.40a y que forma el bien conocido filtro paso alto RC. Conectamos un generador de señal que produce un escalón de tensión con un tiempo de subida de 1ns. La respuesta del circuito la tenemos en la figura 3.40b. Como se ve la señal aparece distorsionada a la salida del circuito debido a que la respuesta en frecuencia del mismo (figura 3.40c) no es plana por debajo de la f_{codo} que en este caso es de $0.5/1\text{ns} = 500 \text{ MHz}$. Sin embargo, inicialmente la señal de salida sí reproduce la entrada con fidelidad puesto que la rampa inicial tiene componentes iguales o cercanas a la frecuencia de codo.

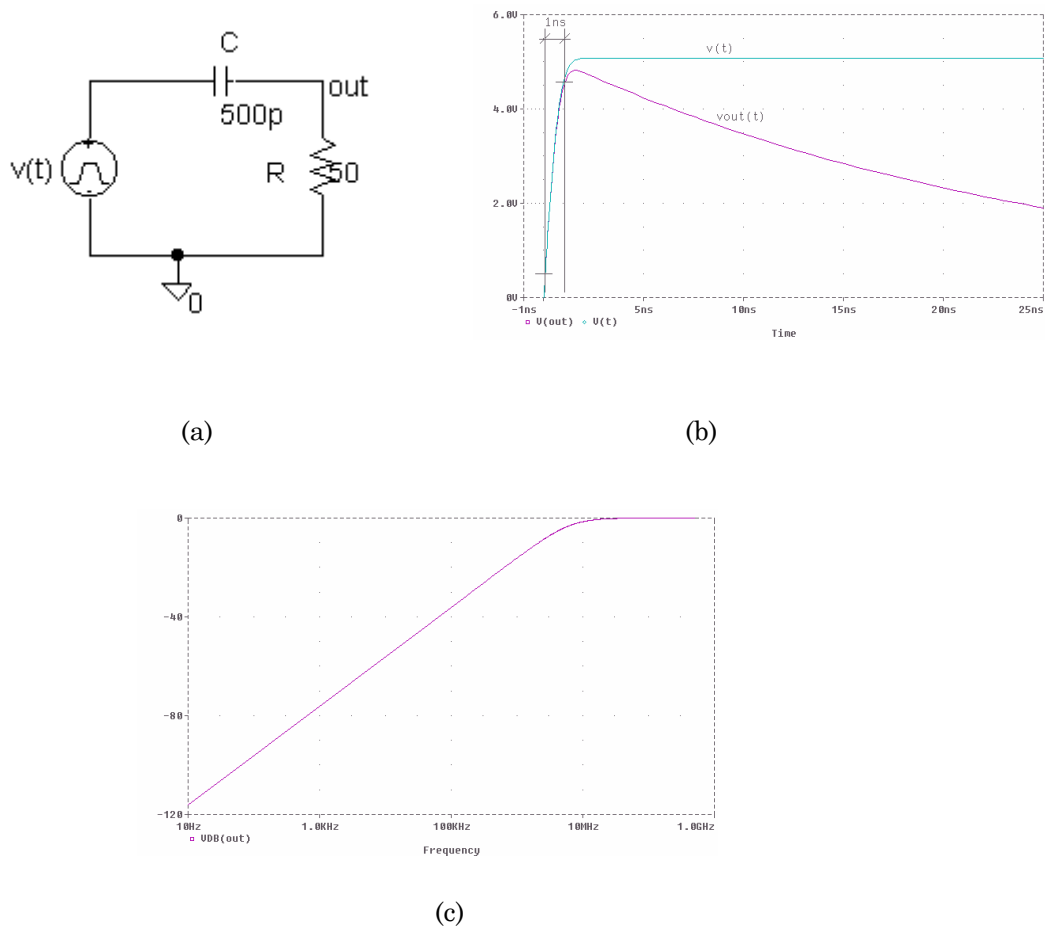


FIGURA 3.40 Ejemplo de circuito distorsionador.

Si hablamos en términos más electrónicos vemos que la impedancia del condensador para la frecuencia de codo es

$$X_c = \frac{1}{2\pi \cdot f_{codo} \cdot C} = \frac{t_r}{\pi \cdot C} = 0'6\Omega \quad (\text{Ec. 4.2})$$

que es prácticamente un cortocircuito y la señal de entrada se reproduce en la salida. Por el contrario, al cabo de 25 ns, lo que corresponde aproximadamente a la respuesta en frecuencia del circuito en 20 MHz ($f = 0.5/t$), $X_c = 15 \Omega$, que ya tiene un valor suficiente como para distorsionar la amplitud de la señal a la salida.

Este ejemplo tan sencillo puede darse de forma involuntaria en la realidad si suponemos que lo que tenemos es un generador conectado a una resistencia y la capacidad representa la de una pista de circuito o un cable. Evidentemente no tendremos una reproducción fiel de la señal de entrada tal y como hemos visto.

Respecto a la velocidad de propagación de la señales, podemos decir que la misma depende del medio que las rodea. El retardo o tiempo de propagación (t_p) medido en ns/m o ps/cm (y su inverso, la velocidad de propagación medida en m/ns o cm/ps) son las variables que nos dan la información sobre la propagación de la señal.

El tiempo de propagación depende de la raíz cuadrada de la constante dieléctrica del medio que rodea la señal. En pistas de circuito impreso también depende del trazado de las pistas, que determina si el campo eléctrico permanece en la placa o se transmite, en parte, al aire. Si queda confinado por la placa, la constante dieléctrica es mayor y, consecuentemente, la propagación de las señales es más lenta; al contrario ocurre si la pista está en contacto con el aire, pues la constante dieléctrica efectiva es menor que la de la placa y la velocidad de propagación aumenta (como consecuencia de esto, las pistas de capas intermedias en un circuito impreso son más lentas que las que se encuentran en contacto con el aire). La tabla 3.7 ofrece valores típicos de tiempo de propagación y constante dieléctrica para diferentes medios y materiales.

El tiempo de propagación también va a servir para determinar si un sistema es o no distribuido. Pero para ello, antes hemos de definir la longitud efectiva de una característica eléctrica, como puede ser el flanco de subida de una señal.

Se define la longitud efectiva como el cociente entre la duración de la característica y el tiempo de propagación de la señal en el medio correspondiente:

$$l_{eff} = \frac{t}{t_p} \quad (\text{Ec. 4.3})$$

Por ejemplo, para la familia ECL 10 KH, el flanco de subida típico es de 1 ns. Si esta señal se propaga por una pista estándar de circuito impreso con un tiempo de propagación de 70,9 ps/cm, su longitud efectiva será:

$$l_{eff} = \frac{1ns}{70'9 ps/cm} = 14cm \quad (\text{Ec. 4.4})$$

Medio	Tiempo de Propagación (ps/cm)	Constante Dieléctrica Relativa
Aire	33,3	1,0
Cable Coaxial (velocidad 0,75c)	44,5	1,8
Cable Coaxial (velocidad 0,66c)	50,8	2,3
PCB FR4, pista externa	55,1-70,9	2,8-4,5
PCB FR4, pista interna	70,9	4,5
PCB Aluminio, pista interna	94,5-106,3	8-10

TABLA 3.7 Tiempo de propagación del campo electromagnético en varios medios.

Esto quiere decir que si pudiésemos fotografiar el cable una vez enviado todo el flanco de subida, el mismo ocuparía (desde el nivel bajo al alto) una longitud de 14 cm. Si esto es así, evidentemente en función del punto donde se mida la tensión tendremos un valor u otro. Por tanto, no son aplicables en esos casos las leyes de Kirchoff y decimos que tenemos un sistema de parámetros distribuidos.

Aunque conceptualmente parece claro que un sistema es de parámetros distribuidos si la característica “cabe” dentro de la longitud del conductor por el que circula la señal, cuantitativamente no existe un acuerdo sobre cuándo podemos empezar a considerar un sistema como distribuido. Parece claro que deberemos fijarnos, en primer lugar, en la característica de la señal más rápida que sea de nuestro interés pues esto nos dará una longitud efectiva mínima y por tanto una longitud máxima de cable antes de considerar el sistema como distribuido. En los sistemas digitales la característica empleada será el tiempo de subida de la señal. En cuanto a la frontera, varía en función del fabricante de lógica o el texto que se consulte. Por una parte autores reconocidos establecen la frontera entre no distribuido (o localizado, como a veces se refiere) y distribuido en una longitud de conductor menor de 1/6 de la longitud efectiva. Por otra, fabricantes como Texas Instruments sitúan la misma en 1/2 de la longitud efectiva.

La conclusión del estudio de ver si el sistema es localizado o no es que esto no depende directamente de la frecuencia de la señal que se emplea sino de su tiempo de subida (de acuerdo con lo expuesto acerca de la frecuencia de codo). Aunque, por lo general, los dos parámetros van asociados (mayor frecuencia implica menor tiempo de subida) puede darse el caso de sistemas que trabajen con pulsos y frecuencias de repetición bajas o aleatorias donde sea de interés mantener la integridad de los mismos. En estos casos la frecuencia no va a tener ninguna influencia en el estudio del sistema y únicamente hay que considerar el tiempo de subida de los pulsos.

4.1.2 Líneas en PCBs

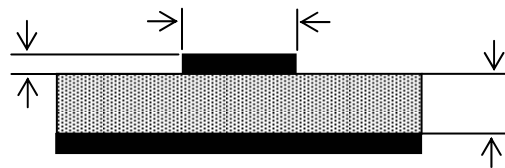
Las líneas de interconexión en los PCB pueden llegar a constituir verdaderas líneas de transmisión bajo ciertas circunstancias de distancia y frecuencia de trabajo. Conforme la tecnología avanza, estos casos son mucho más frecuentes y por tanto es siempre interesante conocer las características y tipos de líneas que podemos encontrar.

La principal diferencia entre los posibles tipos de líneas que podemos hallar en un PCB es si se encuentran o no en un medio homogéneo, es decir, si están o no en contacto con más de un medio. Así, podemos tener líneas microstrip, que son aquellas situadas en la superficie de la placa y en las que el campo electromagnético que se propaga lo hace a través del medio físico de la placa y del aire, y las líneas stripline, que son las trazadas en las capas internas del PCB y donde el campo de propaga delimitado por un medio homogéneo como es el dieléctrico del circuito.

En función de que la línea sea microstrip o stripline, sus características en cuanto a impedancia característica o tiempo de propagación difieren. También lo hacen dependiendo de la existencia o no de planos de masa o de si se encuentran centradas en el dieléctrico. Por todo ello, vamos a realizar un detalle de las características de cada uno de los dos tipos de líneas.

4.1.2.1 Líneas MicroStrip

La estructura de una línea microstrip puede verse en la figura 3.41. Consta de una



pista de anchura w y espesor t separada de un plano de masa una distancia d .

FIGURA 3.41 Diagrama de una línea microstrip.

La impedancia característica de la línea microstrip puede calcularse con ayuda de la siguiente expresión:

$$Z_0 \cong \frac{87}{\sqrt{\epsilon_r + 1,41}} \ln \left(\frac{5,98h}{0,8w + t} \right) \Omega \quad (\text{Ec. 4.5})$$

donde h , w y t están dados en pulgadas y ϵ_r es la constante dieléctrica del substrato. La expresión anterior es válida si $0,1 < w/h < 0,2$ y $1 < \epsilon_r < 15$.

El tiempo de propagación se puede calcular como $t_p = 33,3 \sqrt{\epsilon_{ef}} \text{ ps/cm}$ donde ϵ_{ef} es la constante dieléctrica efectiva que no coincide con el valor de la del dieléctrico por el hecho de que parte del campo electromagnético está propagándose en el aire. El valor de la constante dieléctrica efectiva se calcula con ayuda la siguiente expresión:

$$\varepsilon_{ef} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2} \left[\frac{1}{\sqrt{1 + \frac{12h}{w}}} + 0,04 \left(1 - \frac{w}{h}\right)^2 \right] \quad w < h$$

$$= \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2} \left[\frac{1}{\sqrt{1 + \frac{12h}{w}}} \right] \quad w > h$$

(Ec. 4.6)

La atenuación que sufre la señal al atravesar la línea microstrip presenta las mismas componentes que el cable coaxial. Las pérdidas en continua se deben a la resistencia del cobre que puede calcularse como:

$$R_{DC} = \frac{\rho L}{A} = \frac{\rho L}{wt} \Omega$$

(Ec. 4.7)

donde ρ es la resistividad del metal en ohmios-metro.

La componente en alterna de las pérdidas se debe al efecto pelicular que aparece con la frecuencia (figura 3.42). La profundidad en la que se distribuye la corriente por efecto pelicular puede calcularse como:

$$\delta = \sqrt{\frac{\rho}{\pi f \mu}} \text{ m}$$

(Ec. 4.8)

donde ρ es la resistividad del material, f la frecuencia y μ la permeabilidad del espacio libre.

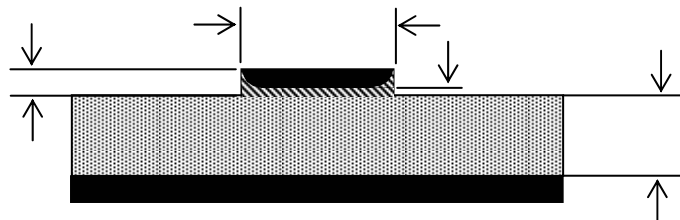


FIGURA 3.42 Distribución de corriente por efecto pelicular.

Tomando esa expresión y aplicándola a la de la resistencia, obtenemos una aproximación a la resistencia en alterna a falta de otros términos a comentar con posterioridad.

$$R_{ac\ cond} \cong \frac{\rho}{w\delta} = \frac{\sqrt{\rho\pi\mu f}}{w} \Omega/m$$

(Ec. 4.9)

Esta aproximación es válida sólo cuando la profundidad por efecto pelicular es menor que el espesor del conductor (t).

Para acabar con el cálculo de la resistencia en alterna, hay que tener en cuenta la resistencia del conductor de masa, que en este caso es el plano de masa por debajo de la línea microstrip. La distribución de corriente se concentra en su mayor parte en una profundidad igual a la del efecto pelicular y perpendicularmente a la dirección de

la pista, con un máximo justo por debajo de la misma. La expresión aproximada de la distribución de corriente es:

$$I(x) \cong \frac{I_0}{\pi h} \frac{1}{1 + (x/h)^2} \quad (\text{Ec. 4.10})$$

donde I_0 es la corriente que circula por el conductor.

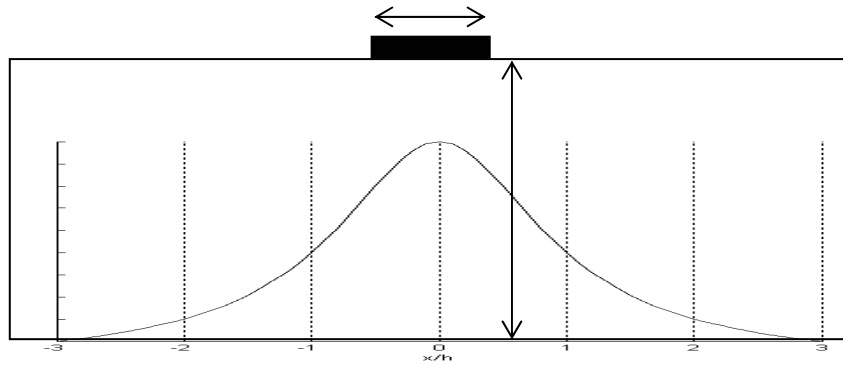


FIGURA 3.43 Distribución de la corriente en el plano de masa en una línea microstrip.

Además se puede observar que si hacemos

$$\int_{-3h}^{3h} \frac{I_0}{\pi h} \frac{1}{1 + (x/h)^2} dx = \frac{2I_0}{\pi} \arctan(3) \cong 0,795 \cdot I_0 \quad (\text{Ec. 4.11})$$

resulta que el 79,5% de la corriente está contenida en una distancia de $\pm 3h$ respecto de la posición del conductor. Si, como hemos supuesto, la corriente se distribuye en profundidad una longitud δ , podemos aproximar el plano de masa como un conductor de sección $\delta \times 6h$, por lo que aplicando la expresión de la resistencia de un conductor, tenemos:

$$R_{ac\ masa} \cong \frac{\rho}{6\delta h} = \frac{\rho}{6h} \sqrt{\frac{\pi \mu f}{\rho}} \Omega/m \quad (\text{Ec. 4.12})$$

Por tanto, la resistencia total será la suma de la resistencia de la pista y la del plano de masa:

$$R_{ac\ microstrip} = R_{ac\ cond} + R_{ac\ masa} \cong \frac{\sqrt{\rho \pi \mu f}}{w} + \frac{\sqrt{\rho \pi \mu f}}{6h} = \sqrt{\rho \pi \mu f} \left(\frac{1}{w} + \frac{1}{6h} \right) \Omega/m \quad (\text{Ec. 4.13})$$

Esta expresión debe ser corregida y aumentada en un 10% a un 50% si se tiene en cuenta que el perfil de la pista no es perfecto y que presenta una cierta rugosidad. Sin embargo, para la mayoría de las aplicaciones esta aproximación es válida.

Algunas veces se prefiere expresar esta resistencia como una resistencia superficial R_s que se obtiene a partir de la resistencia en alterna dividiéndola por la raíz cuadrada de la frecuencia. Esto es:

$$R_{ac} = R_s \sqrt{f} \quad (\text{Ec. 4.14})$$

En la práctica es común encontrar el valor de R_s referido a una superficie de conductor de anchura igual a su longitud. En este caso la resistencia en alterna viene expresada en ohmios por cuadro (Ω/\square) y corresponde a la resistencia de un cuadrado de conductor si importar su tamaño. De esta forma, el cálculo de la resistencia en alterna total del conductor se traduce en contar el número de cuadrados que caben en el área determinada y multiplicar ese número por $R_s \sqrt{f}$. El tamaño de lado mínimo será el de la menor de las dimensiones w o L .

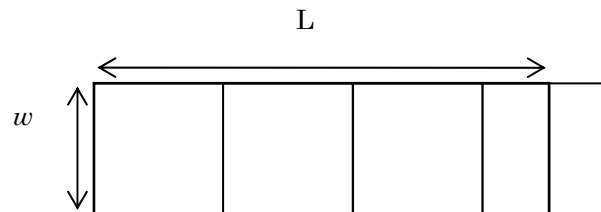


FIGURA 3.44 Cálculo de la resistencia en alterna mediante la resistencia superficial.

Como ejemplo puede observarse la figura 3.44 donde, en una sección de pista de longitud L y anchura w , se han podido dibujar 3,5 cuadrados. Por tanto, la resistencia en alterna será:

$$R_{ac} = 3,5 \cdot R_s \cdot \sqrt{f} \quad (\text{Ec. 4.15})$$

Por último, hay que tener en cuenta que el dieléctrico presenta pérdidas que en general, por debajo de 1 GHz no son comparables con las pérdidas del conductor. No obstante, si se tienen en cuenta estas pérdidas, y tal y como se comentó cuando se hablaba del cable coaxial, la constante dieléctrica se convierte en un número complejo $\epsilon = \epsilon' - j\epsilon''$ donde la parte imaginaria representa las pérdidas que pueden interpretarse como la conductividad del dieléctrico. Si suponemos esto, podemos obtenerla como:

$$1/\rho = 2\pi f \epsilon'' \quad (\text{Ec. 4.16})$$

donde ρ es la resistividad del material.

Sin embargo, en vez de hablar de conductividad del material, es más corriente encontrar expresada la misma como la tangente de pérdidas:

$$\tan|\delta_d| = \frac{\epsilon''}{\epsilon'} = \frac{1}{2\rho\pi f \epsilon} \quad (\text{Ec. 4.17})$$

Y si lo que buscamos es un equivalente como componente para introducir en nuestro modelo de línea de transmisión, las pérdidas del dieléctrico pueden

representarse como una conductancia en paralelo con la capacidad de la línea y de valor:

$$G = \frac{\varepsilon''}{\varepsilon'} (2\pi f C) \quad (\text{Ec. 4.18})$$

donde C es la capacidad distribuida de la línea.

En la práctica, si el diseño exige frecuencias superiores al gigahertzio, los substratos deben cambiar y entonces se emplean materiales cerámicos.

4.1.2.2 Líneas Stripline

Las líneas stripline son aquéllas que podemos encontrar en las capas internas de un PCB. Su estructura puede verse en la figura 3.45.

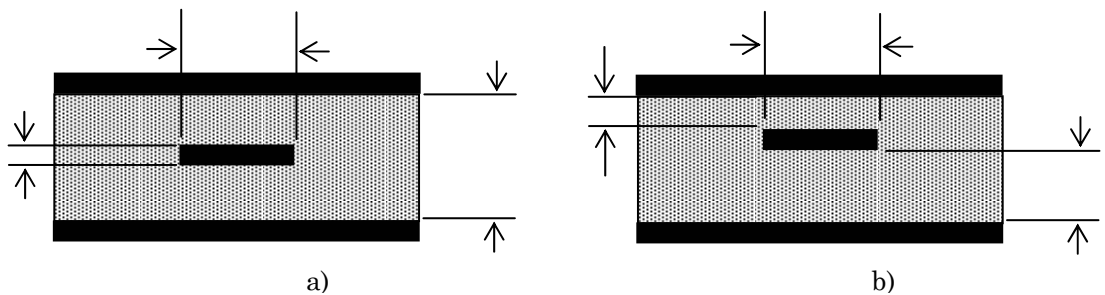


FIGURA 3.45 Estructura de una línea stripline a) simétrica y b) desplazada.

Como se observa la estructura de una línea stripline consta de una pista de anchura w y espesor t entre dos planos de masa. La posición de la pista respecto de los dos planos determina si la stripline es simétrica (equidistante de los dos planos) o desplazada (más próxima a un plano que a otro).

Las expresiones para la obtención aproximada de la impedancia característica de una línea stripline son:

$$Z_{0sim} \cong \frac{60}{\sqrt{\varepsilon_r}} \ln \left(\frac{4h}{0,67\pi(t + 0,8w)} \right) \Omega \quad (\text{Ec. 4.19})$$

para la stripline centrada y cuando $w/h < 0,35$ y $t/h < 0,25$, y

$$Z_{0despl} \cong 2 \frac{Z_{0sim}(2a, w, t, \varepsilon_r) \cdot Z_{0sim}(2b, w, t, \varepsilon_r)}{Z_{0sim}(2a, w, t, \varepsilon_r) + Z_{0sim}(2b, w, t, \varepsilon_r)} \Omega \quad (\text{Ec. 4.20})$$

para la línea stripline desplazada.

Respecto a las pérdidas, en una línea stripline, la corriente se concentra en los bordes superior e inferior del conductor y su distribución depende de la proximidad de los planos de masa. Si es una línea stripline centrada, la corriente se dividirán en

partes iguales en las mitades superior e inferior de la pista. Si la línea no está centrada, la distribución dependerá de la distancia a cada uno de los planos.

En los planos de masa, la distribución de corriente seguirá la misma expresión que en el caso de las microstrip, con la salvedad del valor de I_0 , que será función de la distancia a los planos de masa (figura 3.46).

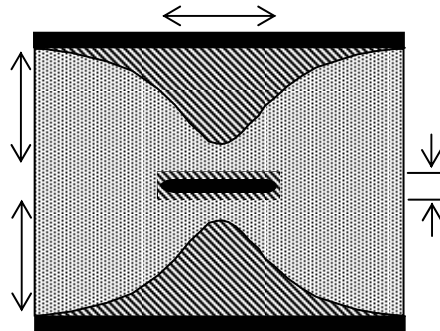


FIGURA 3.46 Distribución de corrientes en una línea stripline.

Por tanto, la resistencia en alterna de una línea stripline puede aproximarse por el equivalente paralelo de las resistencias de las mitades superior e inferior del conductor. Cada una de éstas puede calcularse como si se tratase de una línea microstrip empleando la fórmula deducida en ese punto tomando adecuadamente el valor de la distancia al plano de masa h .

$$R_{ac\ stripline} = \frac{R_{ac\ microstrip}(a) \cdot R_{ac\ microstrip}(b)}{R_{ac\ microstrip}(a) + R_{ac\ microstrip}(b)} \Omega / m \quad (\text{Ec. 4.21})$$

Al igual que en las líneas microstrip, también en este caso debemos tener en cuenta las pérdidas del dieléctrico siempre que la frecuencia de trabajo sea elevada.

4.1.3 Terminaciones

Para mantener bajo control el ruido debido a las reflexiones, se requiere el empleo de terminaciones en aplicaciones de alta velocidad, en todas las conexiones excepto, si acaso, en las más cortas, como cuantificaremos más adelante. Para ello, se emplean diferentes tipos de terminaciones.

A groso modo, podemos clasificarlas como: pasivas (resistencias) y activas (diodos). Las terminaciones pasivas pueden subdividirse en esquemas serie y paralelo, más sus variaciones. Los mecanismos básicos los estudiaremos en los siguientes apartados, si bien se encuentran resumidos en la tabla 3.3 a, b y c para CMOS y TTL y en tabla 3.4 a, b y c para ECL. La elección de un determinado tipo de terminación debe ser considerado al mismo tiempo que se estudia cuál es el camino de retorno de continua.

Las salidas ECL (seguidor por emisor), CMOS (drenador abierto) y TTL (colector abierto) requieren de un circuito externo para proveer el retorno de continua. Por otro lado, no es necesario ningún tipo de circuito externo para las configuraciones CMOS y TTL en totem-pole. En cualquier caso, los retornos de continua son, por sí mismos, un problema. Requieren de circuitería adicional y la corriente de standby no sólo consume potencia sino que degrada los niveles de salida VOH, debido a la caída óhmica en la resistencia de salida.

4.1.3.1 Formas de Onda en una Línea no Terminada

En la figura 3.47 se muestra una línea de transmisión definida como abierta, no terminada o stub. Se supone que la carga L no consume corriente. Cualquier camino de retorno para continua no está mostrado en la figura. Cualquier elemento de carga adicional ralentiza la línea (como veremos más adelante) y complica el patrón de reflexión sobre la carga debido a la reflexión en la misma. Para simplificar la discusión, supondremos que no hay cargas en la línea. El coeficiente de reflexión en el abierto B es, nominalmente, $+1$, mientras que en el extremo del generador A es $-0,8$ para una baja impedancia de salida. Este cambio de polaridad hace que la señal viaje hacia adelante y hacia atrás entre A y B, fenómeno conocido como *ringing*. La forma de onda resultante en B parece una senoide amortiguada, con periodo igual a $4t_d$ (siendo t_d el tiempo de retardo de la línea) y una atenuación entre periodos de valor el producto de los coeficientes de reflexión en A y B. El *ringing* tiene un efecto de degradación de la calidad de la señal y de su timing. Los márgenes de ruido se ven reducidos, haciendo que la línea sea susceptible de otros ruidos.

Un ringing excesivo (muestreado en un instante incorrecto) produce un valor lógico falso. Un valor de tensión excesivo puede llevar a los transistores de entrada a saturación con el correspondiente incremento en el tiempo de reacción, mientras que un valor negativo muy grande puede dañar la correspondiente entrada, por ejemplo en una RAM en las que el circuito de entrada es muy sensible a grandes excursiones de tensión. Para evitar el efecto de las reflexiones en un circuito síncrono es preciso muestrear las señales de datos después de que el ringing se haya establecido, con la penalización en la temporización que ello conlleva.

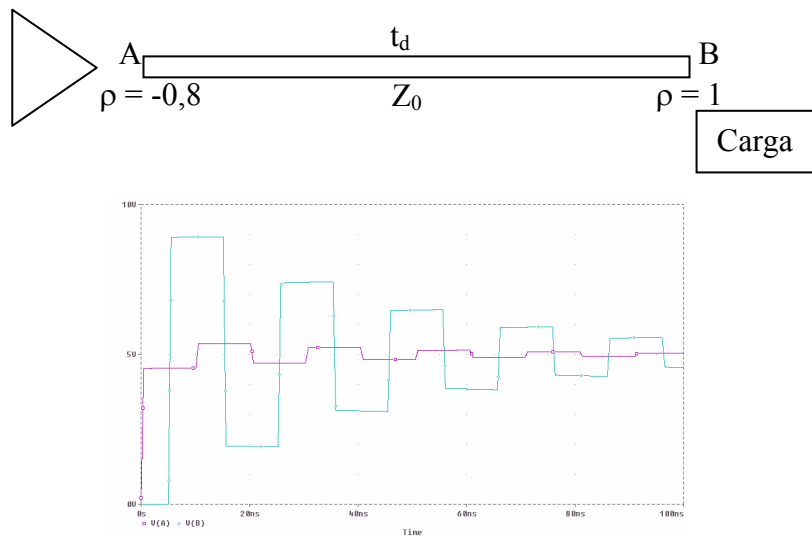


FIGURA 3.47 Línea abierta y forma de onda ante un escalón.

El ringing se puede “esconder” en el propio flanco de la señal cuando el tiempo de retardo es corto comparado con el tiempo de subida de la señal. De hecho, esto establece el criterio por el cual conocemos cuál debe ser la longitud máxima de una línea sin necesidad de incorporar una terminación. El primer cambio de valor de tensión debido a la reflexión (overshoot) se produce en el instante $2t_d$, que debe ser menor que el tiempo de subida (t_r), para que el overshoot permanezca oculto. Así, podemos escribir:

$$t_r \geq 2t_d = 2t_p l \quad (\text{Ec. 4.22})$$

donde t'_r es el tiempo de subida de la señal en el extremo abierto, teniendo en cuenta el valor doble de tensión. Su valor aproximado es de $0,75 t_r$ (ns); t_d es tiempo de propagación (ns); t_p es el tiempo de propagación por unidad de longitud sin carga, $t_p = \sqrt{LC}$ (ns/cm); l es la longitud de la línea (cm) y t'_p es el tiempo de propagación de la línea teniendo en cuenta el efecto de la carga C_L , cuyo valor es, como se verá más adelante:

$$t'_p = 2t_p l \sqrt{1 + \frac{C_L}{C \cdot l}} \quad (\text{Ec. 4.23})$$

Notar que los tiempos de subida y de bajada de la señal son menores en el extremo abierto de la línea que en el generador. La razón es que en ese punto la amplitud de la señal se dobla, debido a la reflexión, pero el intervalo de tiempo no. El ajuste de esa variación de tiempo de subida se realiza de forma empírica tomándose el valor de $0,75$ para obtener el valor t'_r .

La ecuación 4.22 se puede resolver para encontrar la máxima longitud del stub, en el sentido de que la primera sobreoscilación se produzca dentro del tiempo de subida. La solución de la ecuación cuadrática proporciona:

$$l_{\max} = \frac{1}{2} \left[\sqrt{\left(\frac{C_L}{C}\right)^2 + \left(\frac{t'_r}{t_p}\right)^2} - \frac{C_L}{C} \right] \quad (\text{Ec. 4.24})$$

Siempre que el stub no tenga una longitud mayor que la dada en esta expresión, la primera sobreoscilación se completará antes de la subida de la señal. De acuerdo con la bibliografía [20], el overshoot o undershoot queda confinado en un 15% de la amplitud de la señal en estos casos.

4.1.3.2 Terminación en paralelo

Esta es la elección para muchas aplicaciones ECL debido a que no solamente ofrece la mejor calidad sino que también proporciona el necesario camino de retorno dc. En este esquema el extremo de la línea se conecta al voltaje de retorno de continua (5V para TTL en colector abierto o 1,2 V en CMOS, y -2 V en ECL), a través de una resistencia R_t , como se muestra en la figura 3.48.

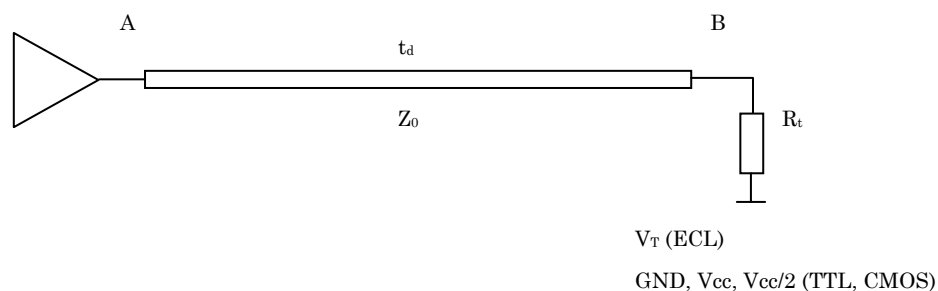


FIGURA 3.48 Terminación en paralelo.

Eligiendo el valor de R_t igual a la impedancia característica de la línea, cualquier señal que se propague a lo largo de la línea disipará su potencia en la carga y evitará cualquier tipo de reflexión. Además la onda de tensión que se propaga con toda su intensidad y la tensión emitida es igual a la recibida. En este caso, si la capacidad de entrada en el extremo de la línea es C_{in} , el tiempo de subida de la señal será $t_r = 2,2C_{in} \frac{Z_0}{2} = 1,1Z_0C_{in}$.

Para CMOS y TTL con configuración totem-pole, la terminación en paralelo consume mucha potencia.

Por ejemplo, conectar un valor de $R_t=Z_0=120 \Omega$ a $V_{cc}=5V$ en TTL, consumiría alrededor de 185mW cuando la salida está a nivel bajo. Si la conectamos a masa, consumiría 41,6 mW a nivel alto.

La configuración totem-pole para CMOS con baja impedancia de salida (R_o , de valor aproximado tanto para nMOS como pMOS, 50-100 Ω) puede terminarse para evitar las reflexiones en el extremo abierto. En este caso tenemos dos problemas:

- Cuando la salida está a nivel bajo, hay potencia dc que fluye debido a la tensión V_t .
- La salida no tiene una excursión V_{cc} -GND, sino que queda reducida debido al divisor de tensión: para el estado alto $\frac{V_{cc}R_t + V_tR_o}{R_t + R_o}$ y para el estado bajo

$\frac{V_tR_o}{R_t + R_o}$. Esto reduce el margen de ruido. Es habitual encontrar un compromiso con R_t entre Z_0 e infinito (el valor de línea abierta).

La combinación V_t - R_t en la terminación paralela puede reemplazarse por su equivalente de Thevenin usando V_{ee} y dos resistencias R_1 y R_2 , como se muestra en la figura 3.49.

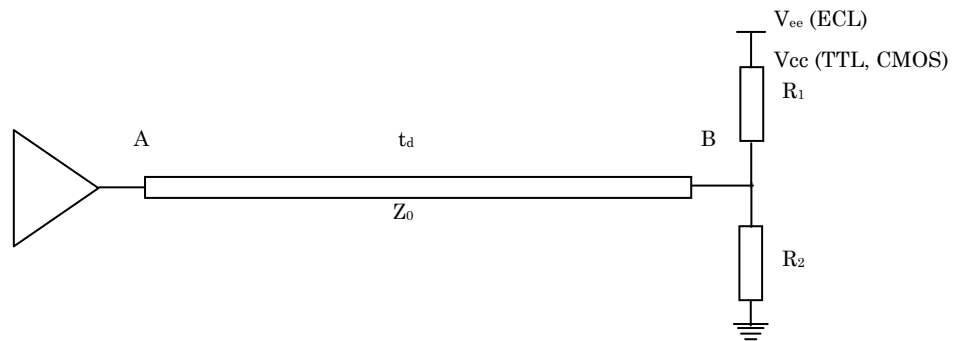


FIGURA 3.49 Terminación paralela usando el equivalente Thevenin.

Los circuitos de las figuras 3.48 y 3.49 son equivalentes a condición de que $R_t=R_1//R_2$ y siempre que $V_t = \frac{V_{ee}R_2}{R_1 + R_2}$. Es sencillo mostrar que $R_1=R_tV_{ee}/V_t$ y $R_2=R_1R_t/(R_1-R_t)$.

Habitualmente suele escogerse una relación $R_1:R_2=2:3$. Para CMOS, es importante no escoger R_1 próximo a R_2 ya que de lo contrario la entrada estaría polarizada cerca del umbral de tensión en el caso de que el driver falle y el receptor podría sufrir daños.

La ventaja de este segundo tipo de terminación con dos resistencias es que no necesita una alimentación adicional. Las desventajas son dos:

- Hay un consumo de potencia estática importante en R_1 y R_2
- Ocupa más espacio y aumenta la capacidad de carga. Esto puede minimizarse empleando resistencias empaquetadas comerciales que integren tanto R_1 como R_2 .

La potencia disipada en la resistencia de terminación depende de los niveles de lógicos de tensión de la tecnología empleada, de la impedancia de carga y de las tensiones de alimentación. La disipación de la carga es inversamente proporcional a la impedancia de terminación, por lo que las líneas de transmisión de alta impedancia disipan menos potencia en sus terminaciones. La potencia total disipada en el caso más general empleando dos resistencias y suponiendo que los dos estados lógicos son igualmente probables es:

$$P = \frac{(V_{HI} - V_{ee})^2 + (V_{LO} - V_{ee})^2}{2R_2} + \frac{(V_{cc} - V_{HI})^2 + (V_{cc} - V_{LO})^2}{2R_1} \quad (\text{Ec. 4.25})$$

4.1.3.3 Terminación en serie

En el esquema para la terminación serie se inserta una resistencia R_s en serie entre el generador de salida y la línea de transmisión, como muestra la figura 3.50. El propósito es eliminar la reflexión debida al generador.

Siendo la resistencia del generador R_0 , una adaptación perfecta requiere que $R_0 + R_s = Z_0$. En una transición BAJO-ALTO, la señal en B es $V_0 Z_0 / (R_0 + R_s + Z_0) = V_0 / 2$, debido al efecto del divisor de tensión. Este nivel se ve doblado en D (el extremo abierto de la línea), para dar la excursión correcta $V_D = V_0$, como se ve en la figura 3.51.

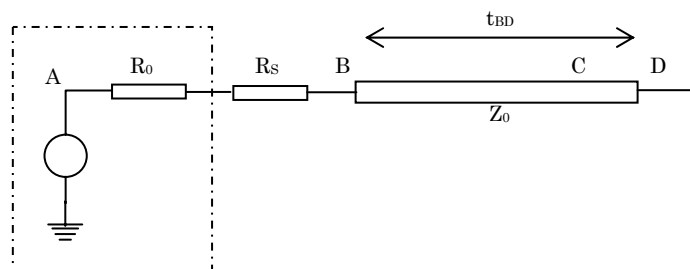


FIGURA 3.50 Circuito con terminación serie.

Todos los puntos de la línea alcanzarán el valor V_0 cuando reciban la reflexión del abierto. Desafortunadamente solamente se produce la transición “suave” en el punto D; el resto de puntos el valor final se alcanza en dos pasos, progresivamente separados en el tiempo. Un receptor en un punto arbitrario C verá los dos saltos separados por un tiempo $2t_{CD}$, que puede ser aceptable si $2t_{CD} \ll t_r$. Las formas de onda son complementarias para una transición ALTO-BAJO, suponiendo que la resistencia interna del generador es la misma en los estados ALTO y BAJO.

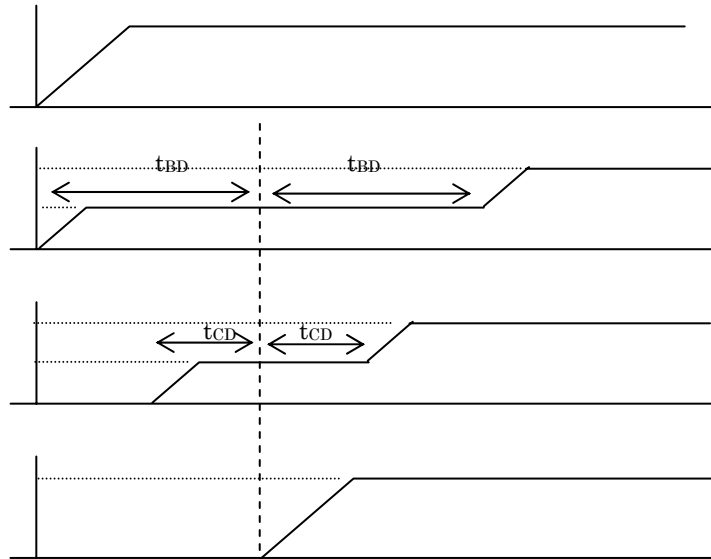


FIGURA 3.51 Formas de onda de subida en un circuito con terminación serie.

4.2 ANÁLISIS PRE-LAYOUT DE LA OMB

Una vez conocidos y desarrollados los principales fundamentos en el diseño de sistemas de alta velocidad que se deben tener en cuenta cuando se realiza un estudio de integridad de la señal, pasamos al análisis pre-layout (simulación previa al rutado de las líneas en la PCB) de la Optical Multiplexer Board.

Este análisis nos permite simular el curso de las señales a lo largo de la línea de transmisión, pudiendo añadir diferentes terminaciones para un acoplamiento correcto de las señales a tratar. Mediante esta funcionalidad observaremos qué ocurre con cada una de las líneas que se consideran críticas en el diseño. Los resultados serán importados por el programa de rutado para realizar una implementación real de las líneas conforme a nuestros estudios.

El programa que se ha usado para estos análisis pre-layout es SpectraQuest [16], una herramienta incluida en el paquete de diseño de Cadence PSD 15.0.

4.2.1 Líneas Diferenciales

El primer estudio que debemos realizar es el de las líneas diferenciales que transportan los datos desde el conector de fibra óptica hasta el deserializador. Estas líneas diferenciales se han simulado con la siguiente topología, siguiendo las instrucciones del datasheet del fabricante.

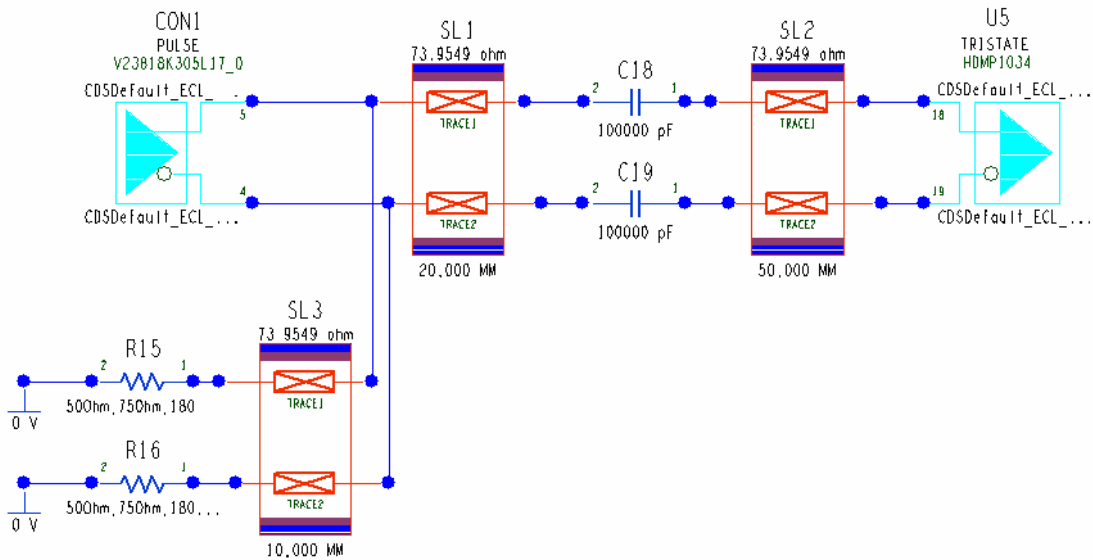


FIGURA 3.52 Topología de las líneas diferenciales.

Los siguientes valores fueron introducidos como datos en las líneas stripline usadas.

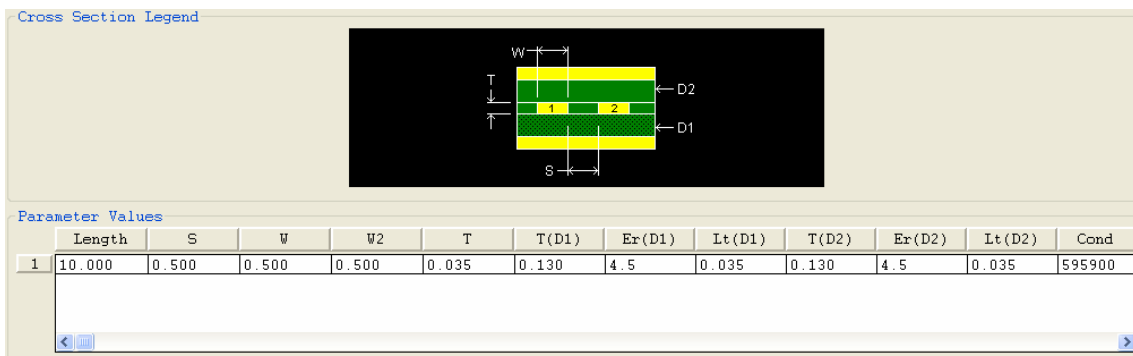


FIGURA 3.53 Valores de la línea stripline.

Y se obtuvieron los siguientes resultados, para un análisis que contiene dos pulsos, de una frecuencia de 640 MHz, con diferentes resistencias de terminación ($50\ \Omega$, $75\ \Omega$, $180\ \Omega$ y $220\ \Omega$). Se puede observar que obtenemos el mejor resultado para una terminación de 180 ohmios, cumpliéndose lo marcado por el fabricante.

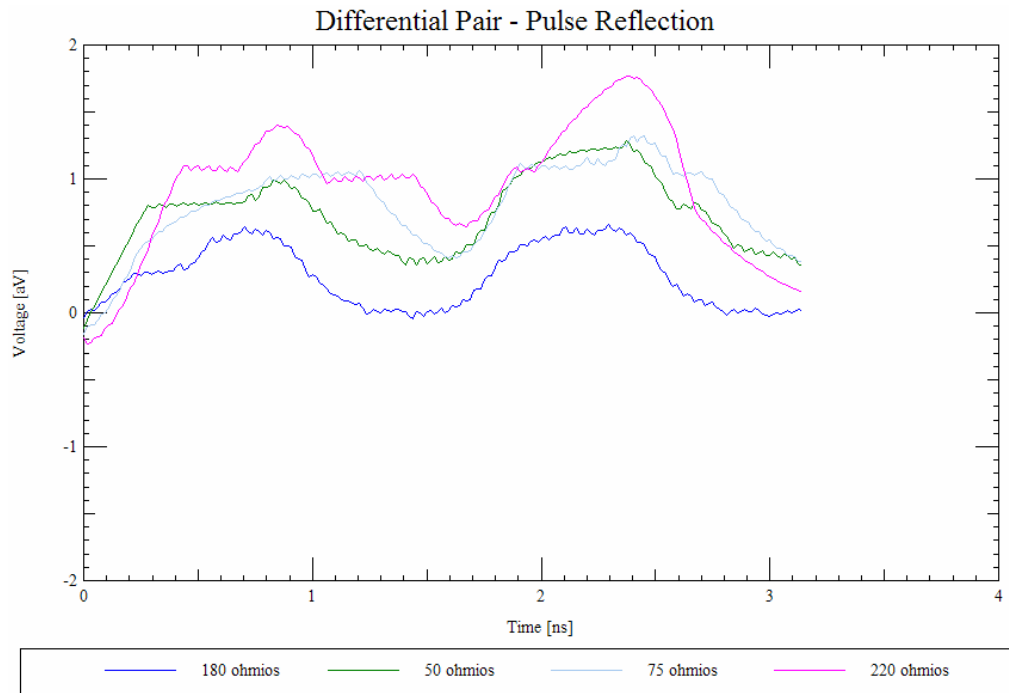


FIGURA 3.54 Línea diferencial con diferentes terminaciones.

Aquí se pueden ver detalladamente las señales correspondientes a la línea diferencial que mejores resultados ofrece, es decir, con resistencia de terminación de 180 ohmios.

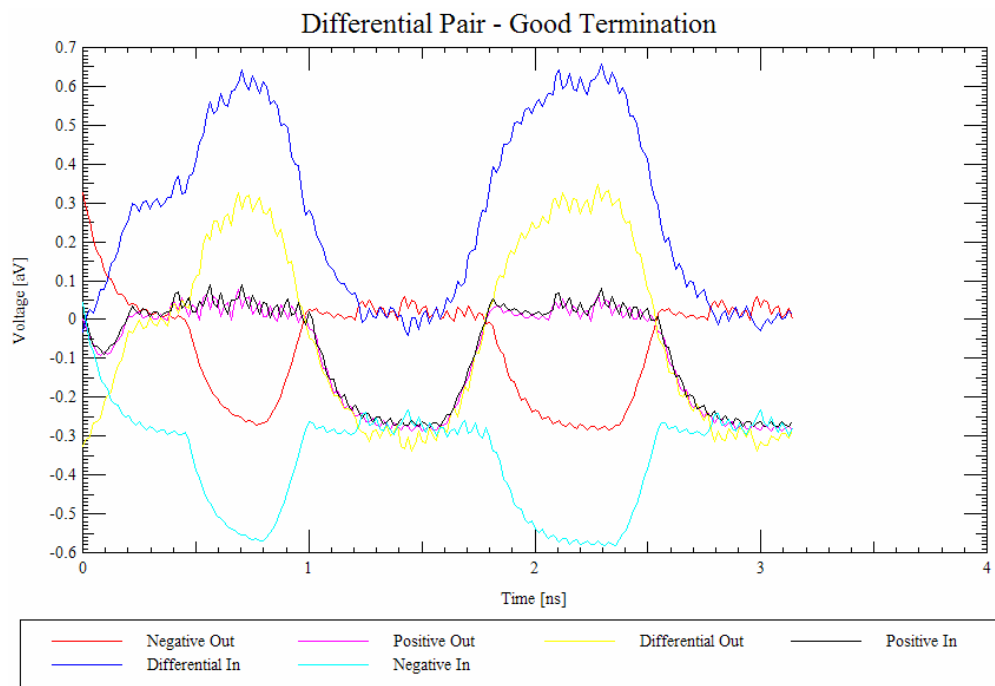


FIGURA 3.55 Señales en la línea diferencial.

Así mismo, se estudió la línea diferencial que transporta los datos de salida, desde el deserializador hasta el conector de fibra óptica.

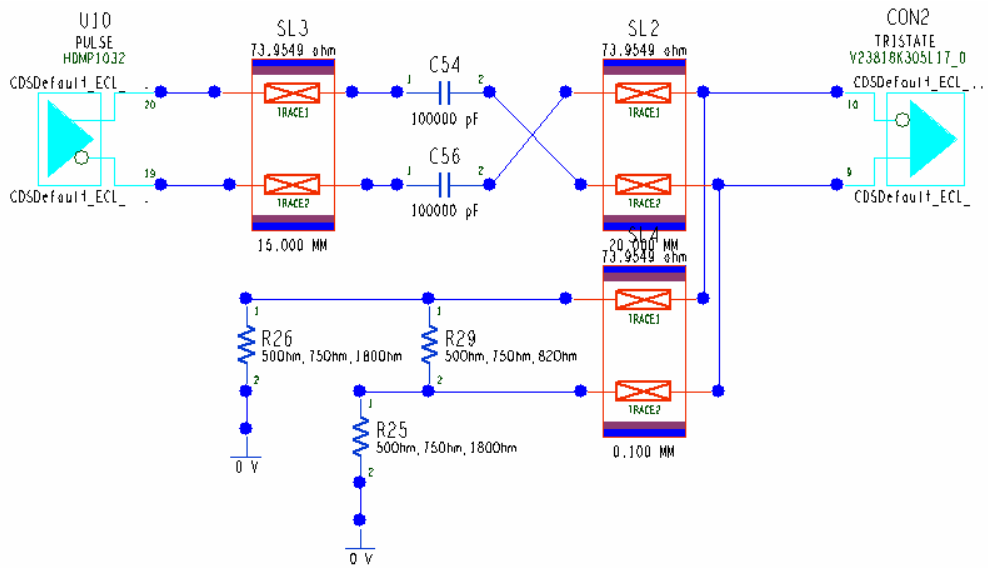


FIGURA 3.56 Línea diferencial de transmisión.

Simulando diferentes valores en las resistencias de terminación. Tomando finalmente la combinación que mejor resultado da; dos resistencias finales de 75 ohmios y una intermedia de 82 ohmios.

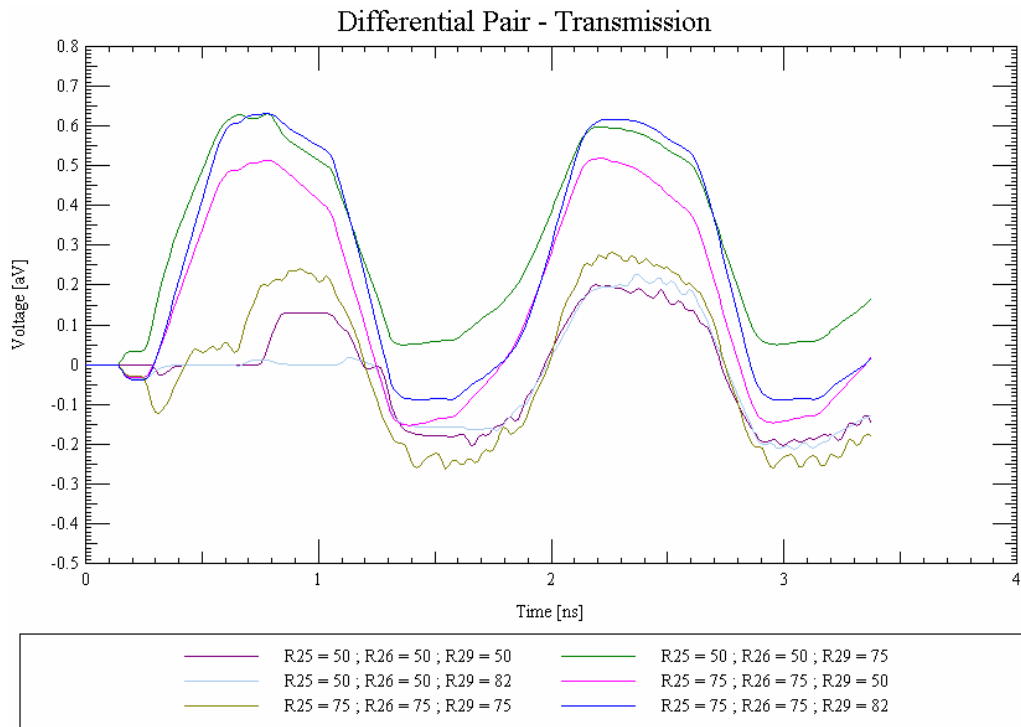


FIGURA 3.57 Análisis de las diferentes combinaciones en la terminación.

Además, se ha realizado el análisis para un valor arbitrario de pulsos marcado por la siguiente simulación.

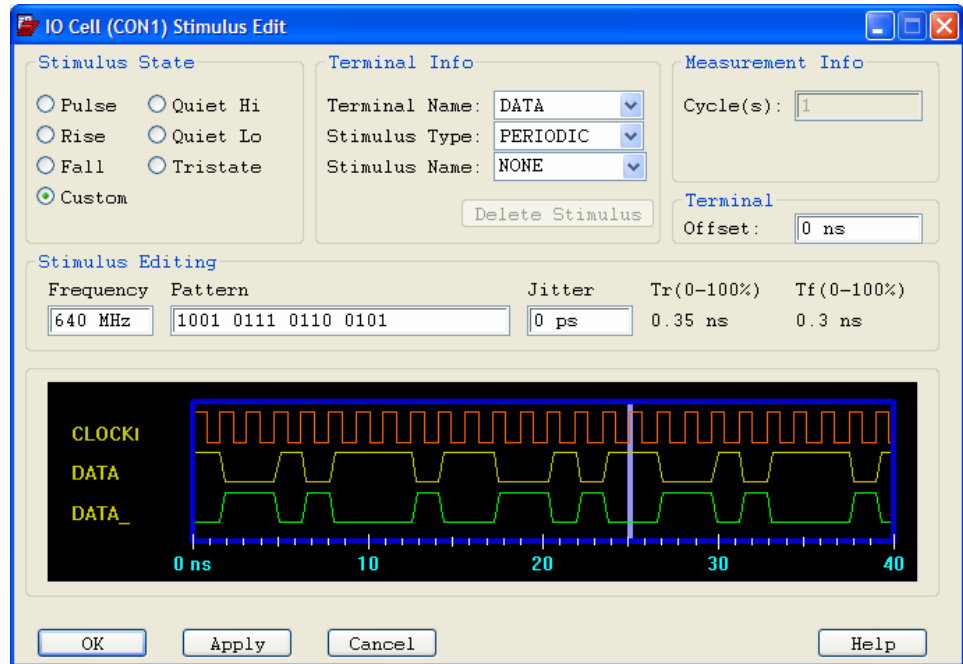


FIGURA 3.58 Análisis para un valor arbitrario de pulsos.

Obteniendo los siguientes resultados.

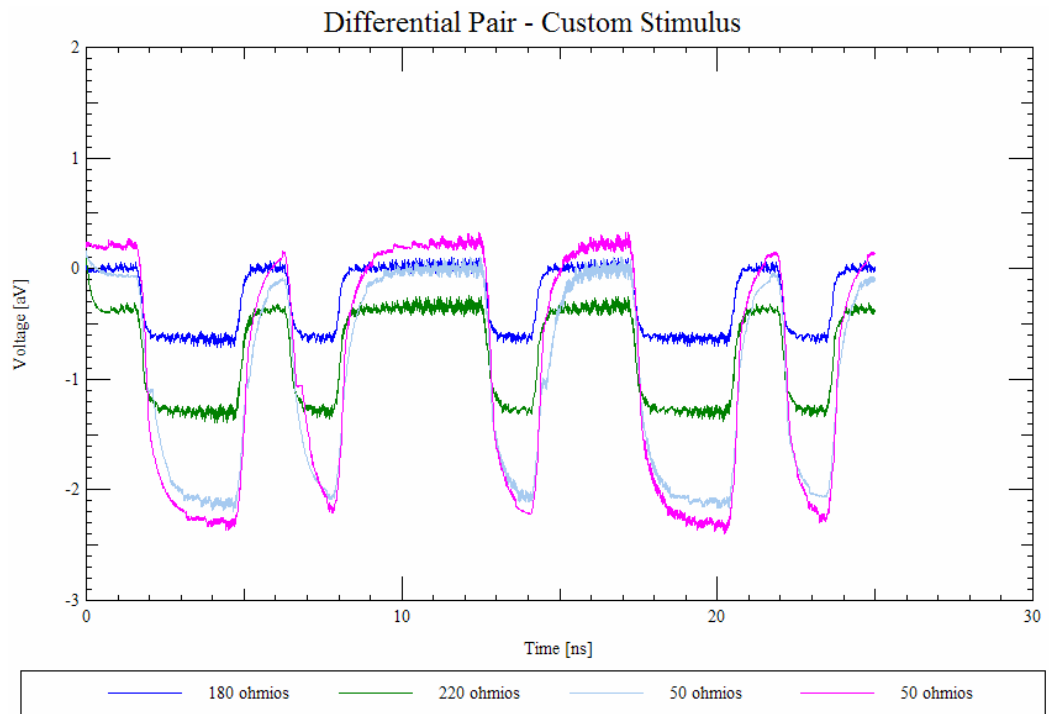


FIGURA 3.59 Resultados para un valor arbitrario de pulsos.

4.2.2 Bus de Datos y Señales de Control

Una vez analizadas las líneas diferenciales, pasamos al estudio de las señales generadas por el deserializador o recibidas por el serializador. Estamos hablando del transporte de datos y señales de control que en modo recepción van desde el HDMP Rx hacia la FPGA de CRC y que en modo transmisión van desde la FPGA de CRC hasta el HDMP Tx.

Haciendo un análisis a las líneas de los buses de datos y a las líneas de control hemos tomado como primera solución aquella en la que no se colocan resistencias de terminación, la topología para estas líneas es la siguiente.

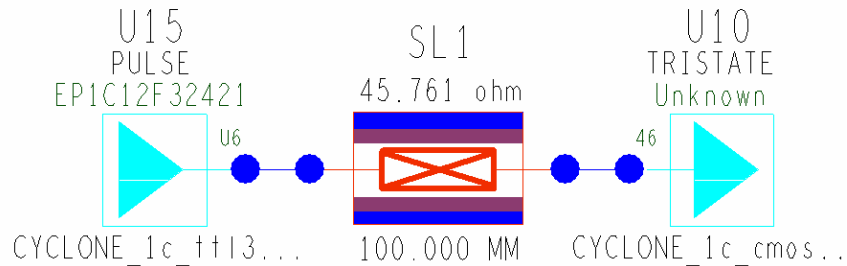


FIGURA 3.60 Topología para los buses de datos y líneas de control.

Se ha realizado un análisis por pulsos, teniendo en cuenta una longitud de línea suficientemente larga para unir ambos componentes (10 cm) y asegurándonos un correcto funcionamiento en el envío. Los resultados nos marcan un pico en recepción de 5,3 V para LVTTTL que trabaja a 3,3 V, este pico supera los márgenes de funcionamiento de dicha tecnología, necesitando por tanto una simulación con terminación para comprobar si hay mejora.

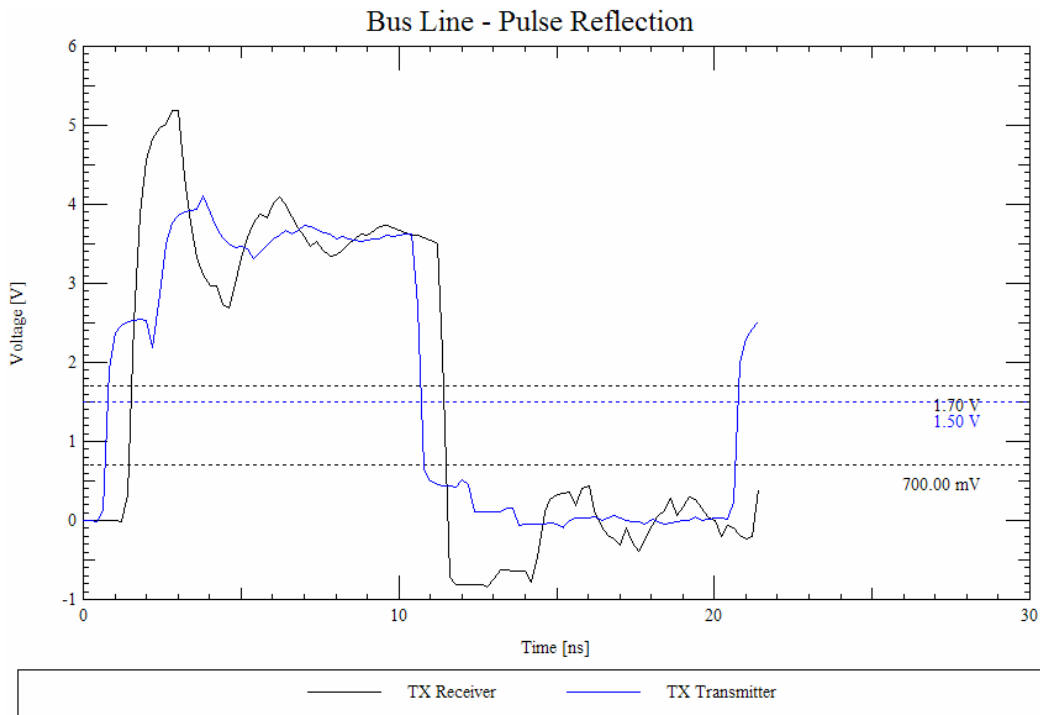


FIGURA 3.61 Reflexión ante un pulso para el bus de datos y control.

Además, también se han realizado simulaciones tanto del flanco de subida como del flanco de bajada. Observándose con mayor claridad el pico de tensión antes mencionado.

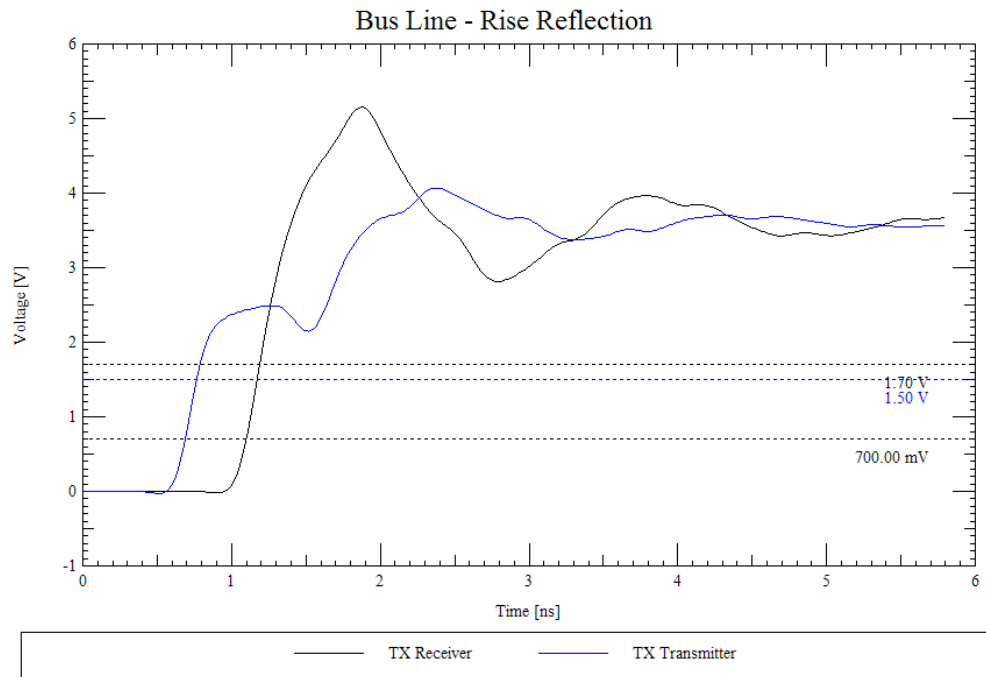


FIGURA 3.62 Simulación del flanco de subida.

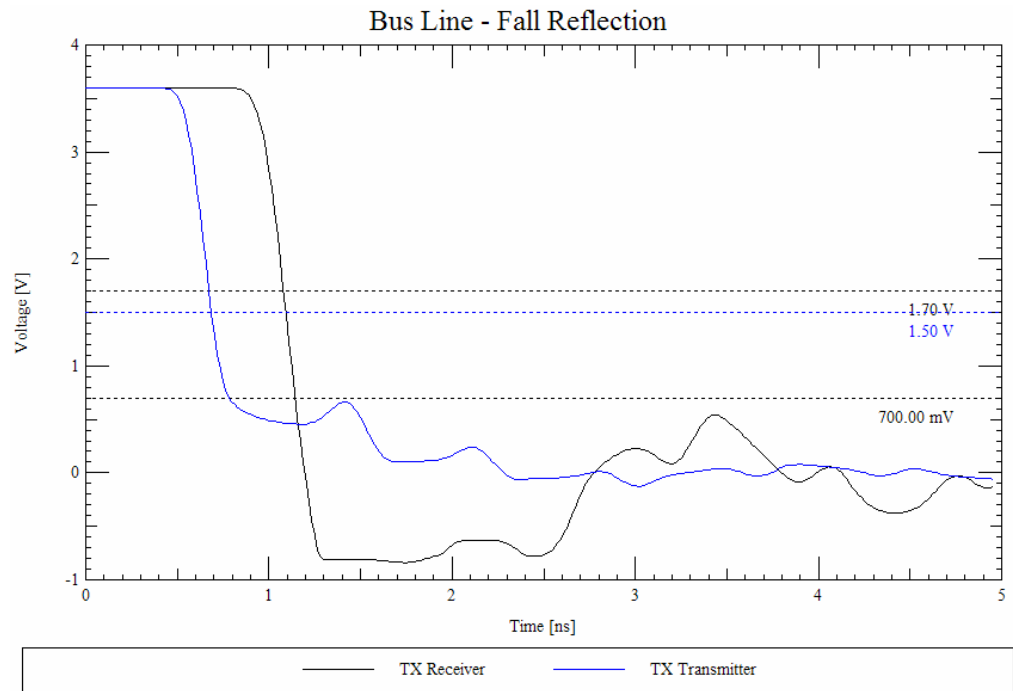


FIGURA 3.63 Simulación del flanco de bajada.

Por tanto, se ha realizado una nueva simulación con la siguiente topología que incluye una terminación en la línea de 50 ohmios.

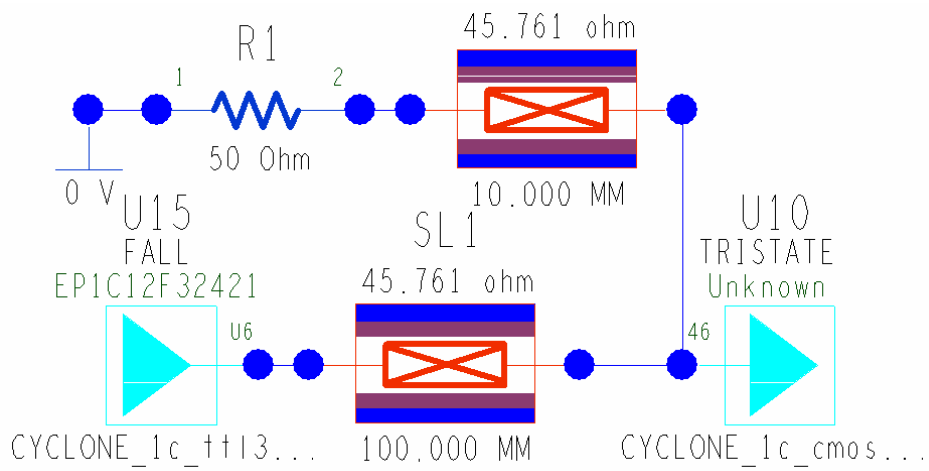


FIGURA 3.64 Topología para los buses de datos y líneas de control con terminación.

Aquí vemos el resultado del análisis para diferentes resistencias.



FIGURA 3.65 Reflexión con terminación ante un pulso para el bus de datos y control.

La siguiente figura muestra el análisis detallado para la resistencia de 50 ohmios. Se puede observar como el pico de tensión que se observaba en las mismas simulaciones pero sin terminación se ha reducido hasta entrar dentro de los márgenes de funcionamiento de la tecnología LVTTL (2,0 V hasta 3,3 V).

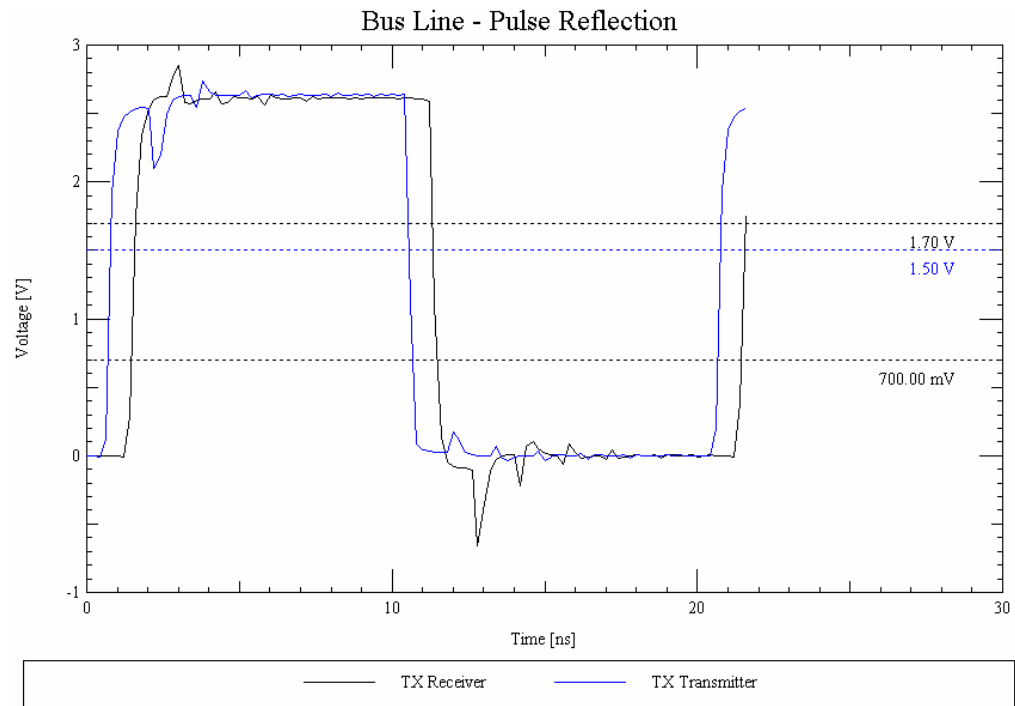


FIGURA 3.66 Reflexión con terminación de 50 ohmios.

Además, también se han realizado simulaciones tanto del flanco de subida como del flanco de bajada.

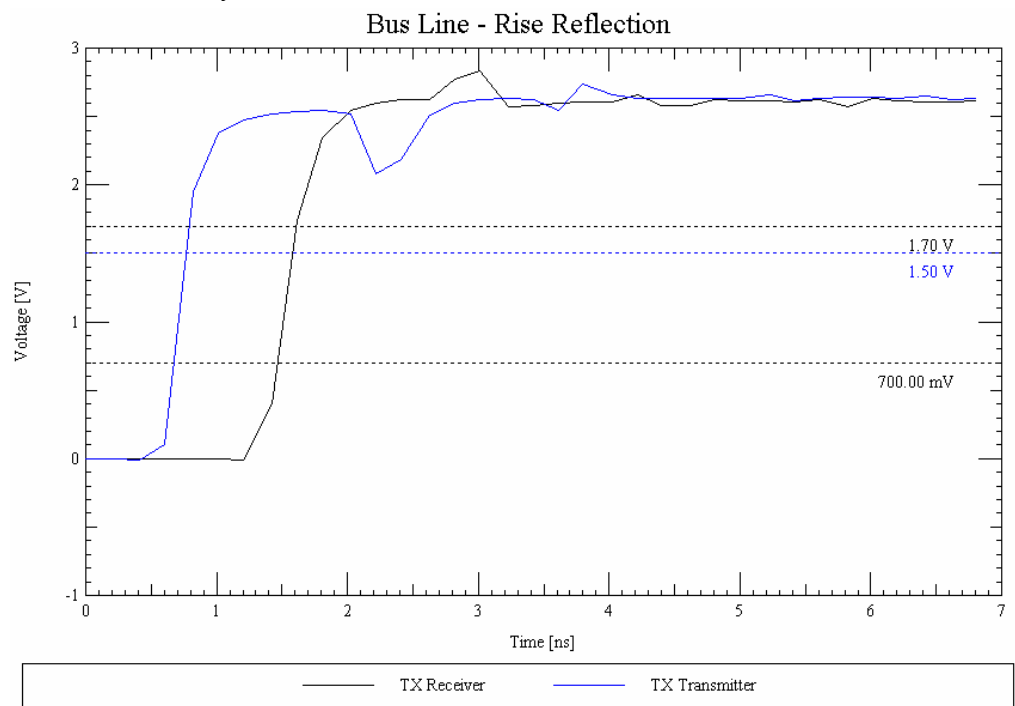


FIGURA 3.67 Simulación del flanco de subida.

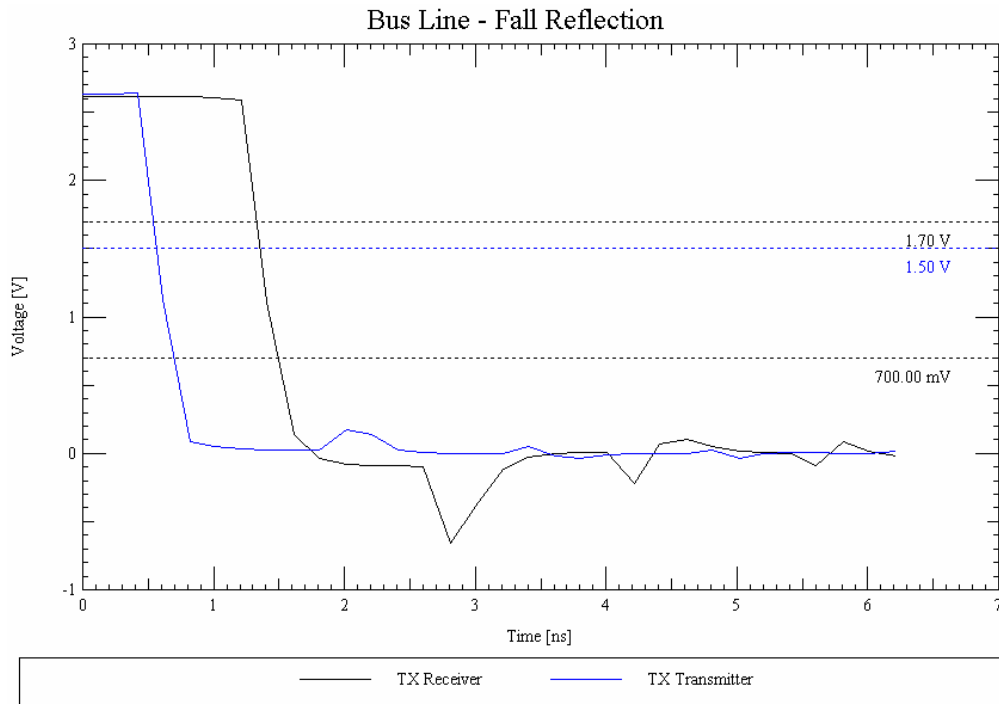


FIGURA 3.68 Simulación del flanco de bajada.

4.2.3 Relojes

Analizados los elementos que componen el sistema de recepción, transmisión y control de los datos que se procesan dentro de la Optical Multiplexer Board, es el momento de analizar los relojes usados y la topología usada.

Una posibilidad que nos aparecía era poner un único reloj para todos los HDMPs, así que se analizó esa topología.

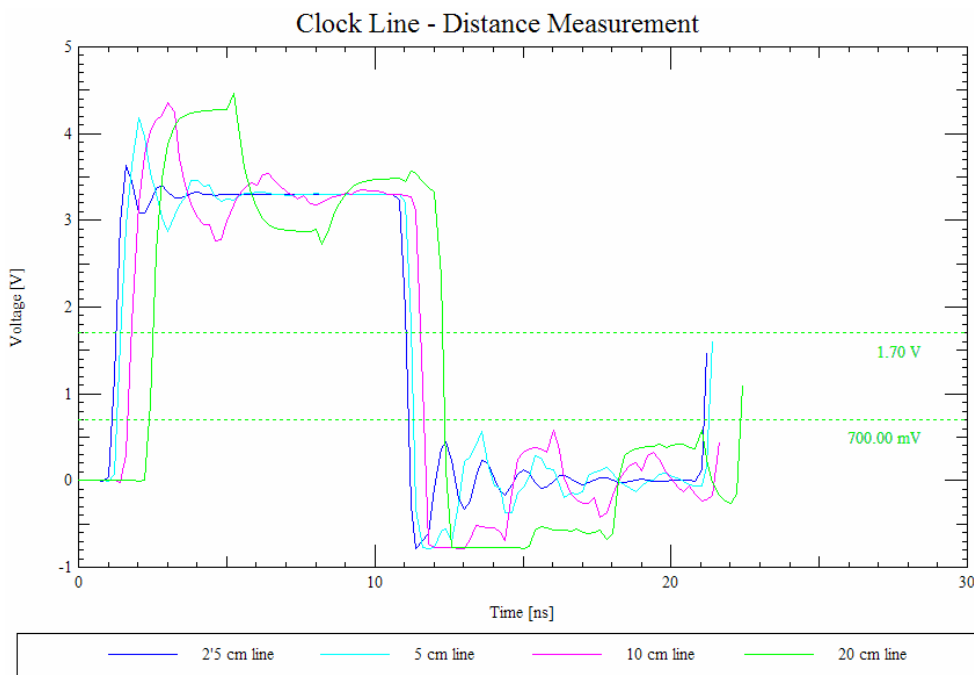


FIGURA 3.69 Análisis de la señal de reloj con diferentes distancias.

Como se puede observar, la distancia de las líneas deteriora bastante la señal de reloj además de obtener diferentes tiempos de recepción. Eso implicaría obligar a nuestro programa de rutado a mantener distancias iguales en la señal de reloj, esta restricción se salvó colocando un reloj para cada uno de los HDMPs.

Así pues, como los HDMPs no tienen porqué trabajar de manera síncrona entre ellos, se optó por la colocación de un reloj al lado de cada HDMP, de manera que la distancia sea mínima. La topología usada es la siguiente.

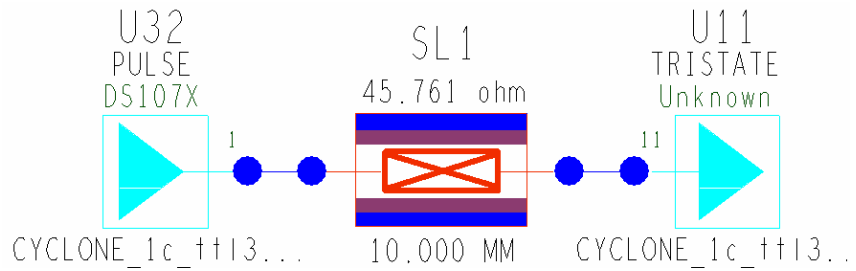


FIGURA 3.70 Topología en los relojes.

Esta topología nos da los siguientes resultados para un pulso.

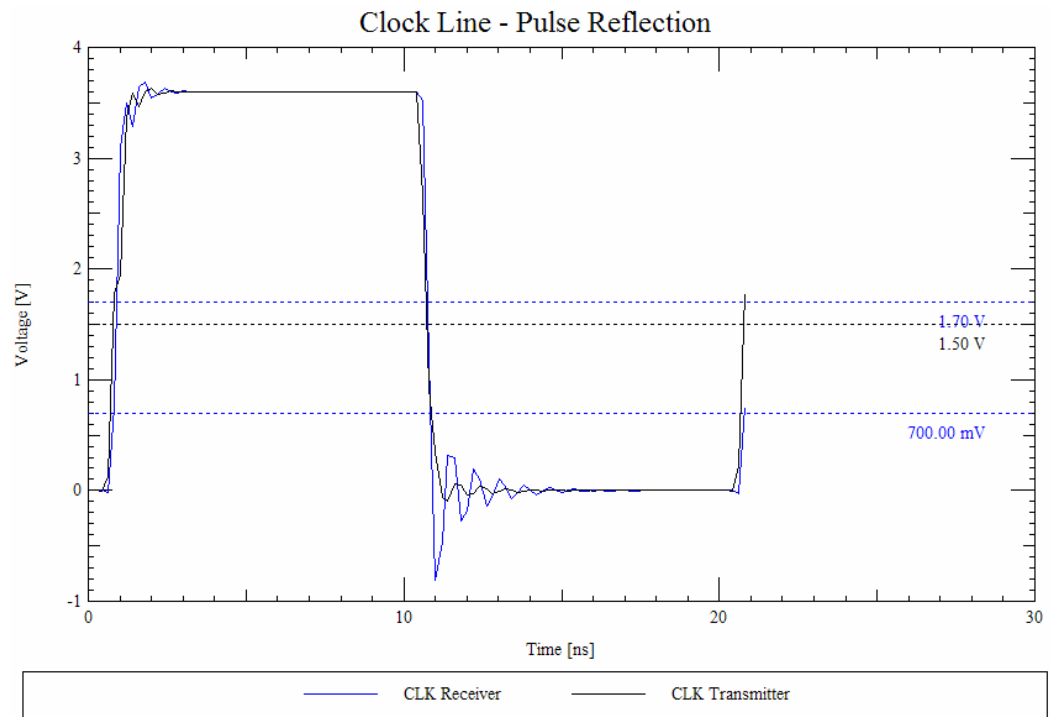


FIGURA 3.71 Simulación de la señal de reloj ante un pulso.

Para el flanco de subida.

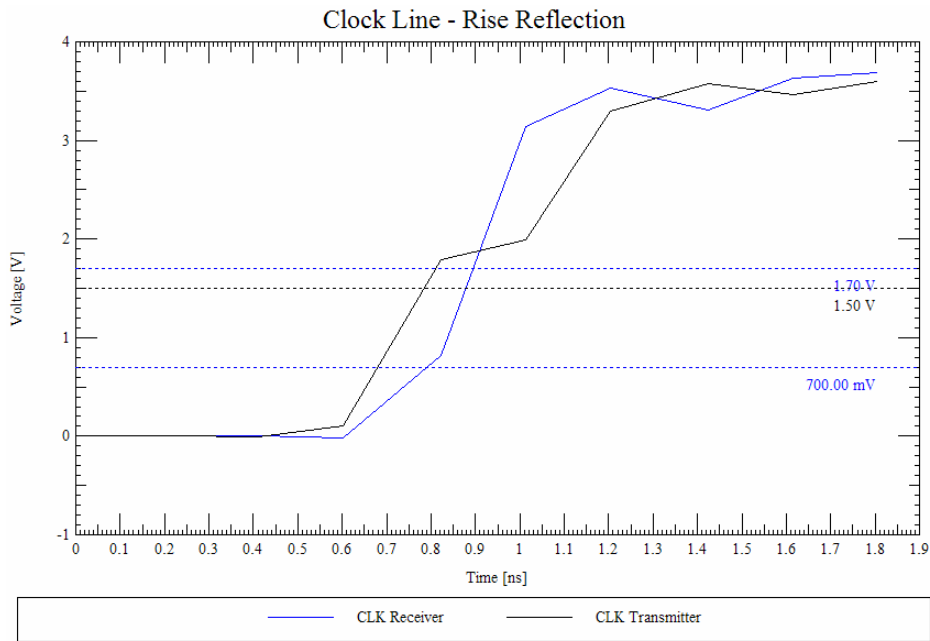


FIGURA 3.72 Simulación de la señal de reloj para el flanco de subida.

Y para el flanco de bajada.

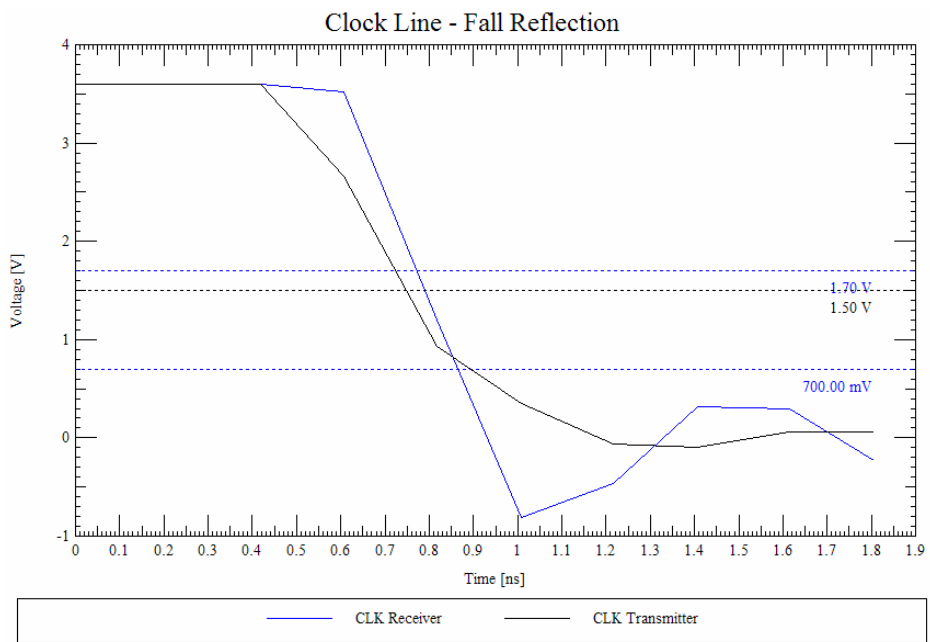


FIGURA 3.73 Simulación de la señal de reloj para el flanco de bajada.

Un último análisis que resulta importante observar es el crosstalk o acoplamiento entre señales próximas.

4.2.4 Crosstalk

La medida de crosstalk que obtenemos mediante la simulación de SpecctraQuest para una señal víctima (señal afectada por las tensiones y corrientes acopladas de los conductores próximos o agresores) se calcula como la diferencia entre la más baja/más alta excursión de la señal del agresor y el voltaje en estado estacionario de la víctima.

Para ello se necesitan generar tablas que contengan el peor caso posible.

```
#####
# SPECCTRAQuest 15.0
# (c) Copyright 1998 Cadence Design Systems, Inc.
#
# Report: Standard Crosstalk Summary Report Sorted By Worst Case Crosstalk
#
#####

*****
Timing Windows Neighbor Group Crosstalk (mV)
*****

Victim XNet      Victim Drvr      HSOddXtalk      HSEvenXtalk      LSOddXtalk      LSEvenXtalk
-----
RXA_D1           U11 2            932.8           0.808            825.8           14.97
RXA_D4           U11 61           469.6           319.8            420.9           476.8
RXA_D5           U11 60           460.3           342              454.7           441.8
RXA_D3           U11 62           443.5           320.4            435.5           407.7
RXA_D9           U11 54           382             262.4            331.2           302.5
RXA_D6           U11 59           357.9           241.3            309.7           371.3
RXA_D14          U11 47           286.2           250.2            369.5           348
RXA_D12          U11 51           365.4           231.7            289.6           323.1
RXA_D10          U11 53           356.7           209.9            231.9           343.5
RXA_D8           U11 55           345.9           213.9            265.9           261.3
RXA_D11          U11 52           343.9           232.1            292.5           285.7
RXA_D7           U11 58           329.3           305.7            243.3           250.1
RXA_D13          U11 50           310.7           221.8            307             268.5
RXA_D2           U11 63           234.6           125.1            154.6           274.9
RXA_D15          U11 46           205.3           144.6            246.1           199.3
RXA_D0           U11 3            0              0                6.499e-006     6.378e-006
-----
```

TABLA 3.8 Valores de análisis de crosstalk en el peor caso.

Obteniéndose mediante las siguientes simulaciones los diferentes valores de crosstalk.

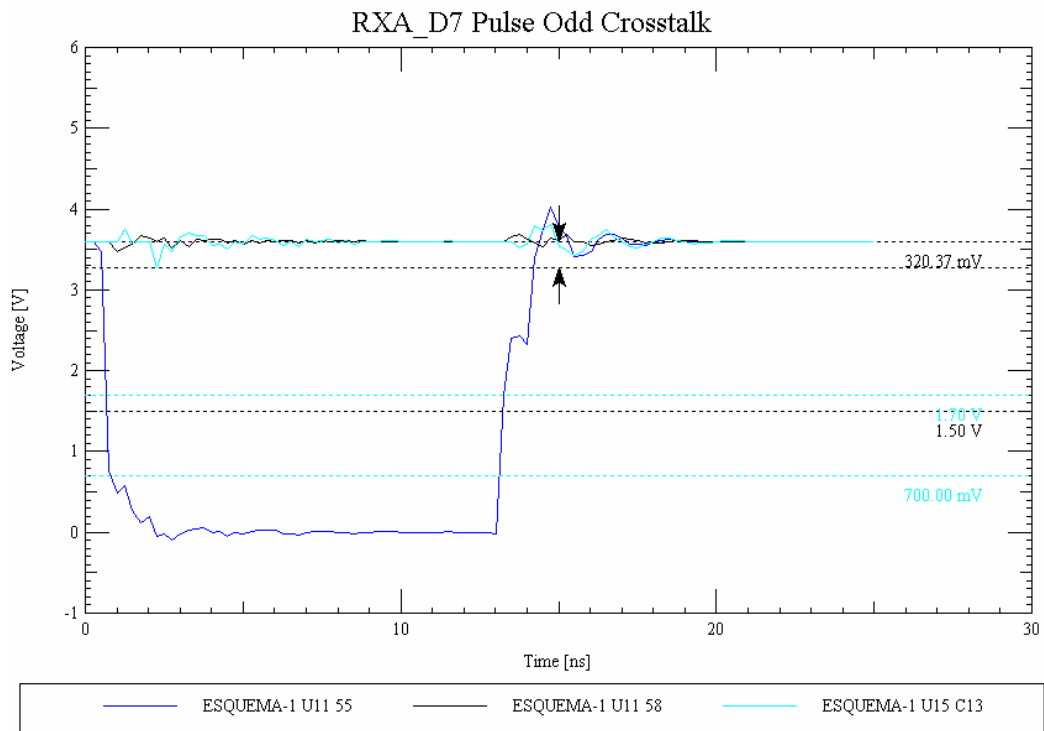


FIGURA 3.74 Simulación del crosstalk a nivel alto para un pulso inverso.

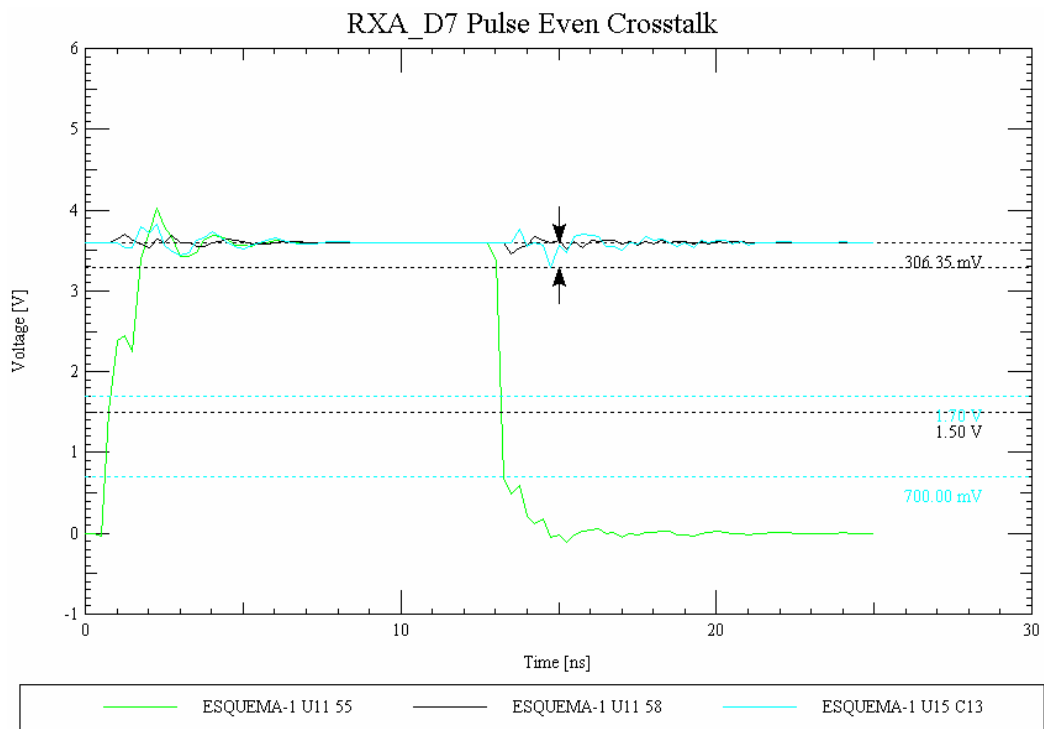


FIGURA 3.75 Simulación del crosstalk a nivel alto para un pulso directo.

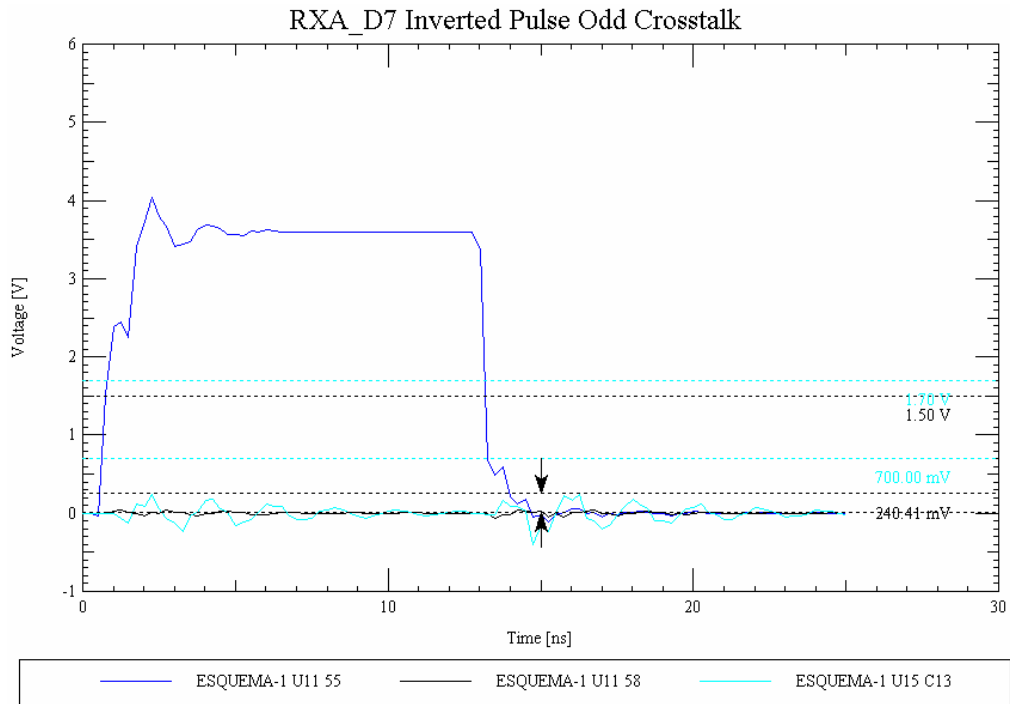


FIGURA 3.76 Simulación del crosstalk a nivel bajo para un pulso inverso.

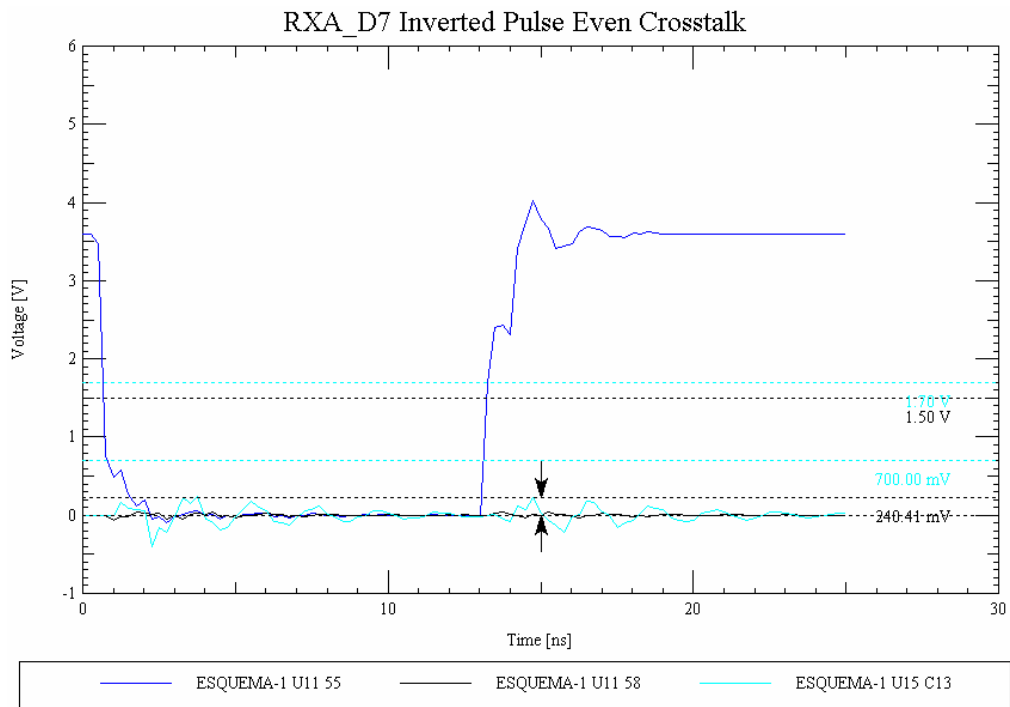


FIGURA 3.77 Simulación del crosstalk a nivel bajo para un pulso directo.

Se puede observar como en ningún caso el nivel de crosstalk supera el margen de ruido establecido para la tecnología LVTTL (400 mV). Con todos estos análisis podemos concluir que nuestro diseño está preparado para su correcto rutado y su posterior testeo. Conociendo de antemano que no aparecerán problemas en las señales debidos a la integridad de la señal en alta velocidad.

5. BIBLIOGRAFÍA

- [1] Castelo, J., et al., *TileCal ROD Hardware and Software Requirements*, ATLCOM-TILECAL-2005-002, 2005.
- [2] Castelo, J., *TileCal ROD Hardware Requirements and LiAr Compatibility*, ROD TileCal Group Internal Note, 2002.
- [3] Castelo, J., *Tile Calorimeter Read Out Driver. Firmware Developments for the Final Prototype*, CERN-LHCC-2004-010, 2004.
- [4] Anderson, K., et al., *Front-End Electronics for the ATLAS Tile Calorimeter*, 4th Workshop on Electronics for the LHC Experiments, Rome, 1998.
- [5] ATLAS Collaboration, *Radiation Test*, TileCal Chicago Group, CERN, http://hep.uchicago.edu/atlas/electr/Rad_testing/Radiation_Testing.html, 1999.
- [6] Anderson, K., et al., *SEE Tests of the TileCal Optical Interface Board*, Enrico Fermi Institute, University of Chicago, 2001.
- [7] Torres, J., Soret, J., González, V., Sanchis, E., Martos, J., Gómez, J. A., *Optical Multiplexer Board for TileCal Data Redundancy*, CERN-LHCC-2004-010, 2004.
- [8] ATLAS Collaboration, *VME Bus*, ATLAS Electronic Co-ordination, CERN, <http://atlas.web.cern.ch/Atlas/GROUPS/FRONTEND/VMEbus/>, 2002.
- [9] Infineon Technologies AG, *V23818-K305-L18 Datasheet*, Gigabit Ethernet Transceiver, <http://www.datasheetarchive.com/datasheet/pdf/68/687493.html>, 2001.
- [10] Agilent, *HDMP-1032A/1034A Datasheet*, Transmitter/Receiver Chip Set, <http://cp.literature.agilent.com/litweb/pdf/5988-3852EN.pdf>, 2001.
- [11] Altera Corporation, *Cyclone Datasheet*, <http://www.altera.com/literature/lit-cyc.jsp>, 2003.
- [12] Altera Corporation, *ACEX Datasheet*, <http://www.altera.com/literature/lit-acx.jsp>, 2003.
- [13] Maxim Dallas Semiconductor, *DS1073-Z80 40 MHz Clock Datasheet*, <http://www.datasheetarchive.com/datasheet/pdf/37/377492.html>, 2003.
- [14] Cadence Design Systems, *Capture CIS v. 15.0*, Allegro Design Entry CIS, http://www.cadence.com/datasheets/capturecis_ds.pdf, 2003.
- [15] Recasens, M^a A., González, J., *Diseño de circuitos impresos con ORCAD Capture y Layout*, Editorial Thomson Paraninfo, 2002.
- [16] Cadence Design Systems, *Allegro PCB SI*, Allegro Design Entry CIS, http://www.cadence.com/products/si_pk_bd/pcb_si/index.aspx, 2003.
- [17] Johnson, W. H., Graham, M., *High-Speed Digital Design*, Prentice Hall, 1993.
- [18] TileCal Valencia Group, *CRC Page*, http://ific.uv.es/tical/rod/crc/xose_crc.html, 2003.
- [19] Sainson, J.-M., *SpecctraQuest Page*, Electronics Applications CERN, <http://support-specctraquest.web.cern.ch/support-specctraquest>, 2003.
- [20] Torres, J., González, V., Sanchis, E., Martos, J., *Diseño Digital de Alta Velocidad*, Editorial Moliner-40, 2004.

VERIFICACIÓN EXPERIMENTAL Y RESULTADOS

1. INTRODUCCIÓN	1
2. DESCRIPCIÓN DE LAS PRUEBAS DEL SISTEMA ROD INICIAL Y RESULTADOS	1
2.1 Descripción del Test Beam de TileCal	1
2.1.1 Desarrollo Experimental	3
2.1.2 Medidas en el Haz del H8	3
2.1.3 Calibración y Desarrollo de los Módulos	4
2.2 Instalación del Sistema ROD Inicial en el Test Beam	5
2.3 Resultados del Sistema ROD Inicial en el Test Beam	7
3. DESCRIPCIÓN DE LAS PRUEBAS DEL SISTEMA ROD FINAL Y RESULTADOS	10
3.1 Descripción de los Tests Previos de la OMB	10
3.2 Instalación del Sistema ROD Final en el Banco de Test	13
3.2.1 Aplicaciones XTestROD y XFilar	14
3.3 Resultados del Sistema ROD Final en el Banco de Test	16
4. BIBLIOGRAFÍA	19

1. INTRODUCCIÓN

Una vez definido el Sistema ROD, sus módulos de preprocesado, sus características, su diseño, sus principales funciones y su implementación, debemos exponer los resultados que se han obtenido en el Test Beam montado en el CERN. En este capítulo describiremos el setup experimental de TileCal usando el haz del pequeño acelerador SPS (Super Proton Synchrotron). A su vez, explicaremos la calibración que se realiza sobre cada uno de los módulos del calorímetro y los parámetros que lo definen.

Explicaremos con detalle la instalación del Read Out Driver y del Módulo de Transición Activo en este Test Beam, el montaje, las pruebas que se realizaron y un análisis de los datos obtenidos. Así mismo, hablaremos de los tests que se realizaron con la Optical Multiplexer Board. Teniendo en cuenta la necesidad de montar un Test paralelo para su análisis.

Estos análisis consisten por una parte en la reconstrucción de la señal recibida, una vez ha sido filtrada con los algoritmos expuestos en capítulos anteriores y por otra parte en la identificación de posibles errores en la transmisión. Dichos errores son ficticios, generados por nosotros, debido a la imposibilidad de observación de los mismos con la luminosidad presente en el Test Beam de TileCal.

Por tanto, hablaremos aquí de dos tests diferenciados. El primero el que se realizó al Sistema ROD inicial, el formado por el RODDemo y el Módulo de Transición Activo, para el cual se usó el Test Beam de TileCal. El segundo el que se desarrolló en el laboratorio del CERN con el Sistema ROD final, formado por la Motherboard del ROD y por la Optical Multiplexer Board, para el cual se usó un Banco de Test creado por nosotros.

2. DESCRIPCIÓN DE LAS PRUEBAS DEL SISTEMA ROD INICIAL Y RESULTADOS

En este punto trataremos todas las acciones realizadas para el test de los módulos que integran el Sistema ROD inicial, módulos que son los prototipos de las versiones finales que se comentan en el siguiente punto. Hablaremos en primer lugar del Test Beam organizado por el CERN, después de la instalación de nuestro sistema en él y por último de los resultados obtenidos.

2.1 DESCRIPCIÓN DEL TEST BEAM DE TILECAL

La comunidad TileCal ha planeado un exhaustivo programa de test [1] para la producción de los módulos del calorímetro. Este programa está pensado principalmente por la necesidad de determinar la respuesta de los módulos a una variedad de test. Así mismo, sirve de base a un test no menos importante, el de la electrónica y la adquisición de datos.

Debido al gran impacto que la tecnología tiene en ATLAS, todos los dispositivos usados en los test deben entenderse perfectamente. A su vez, la mejora tan rápida de los dispositivos electrónicos, hacen que este test deba estar en funcionamiento casi de forma permanente para comprobar si los cambios realizados en la electrónica son satisfactorios para el análisis.

Usando el haz que nos proporciona el SPS (Super Proton Synchrotron), TileCal realiza dos tipos de test principales: una exposición larga a la energía disponible en el haz y la realización de pruebas combinadas con los otros subdetectores de ATLAS. Esto permite un conocimiento claro de cómo se desarrolla el transporte de cada una de las partículas en el detector.

Los objetivos que se buscan en el programa del Test Beam son:

- Calibración de una parte de la producción de los módulos.
- Monitorización de la estabilidad de los dispositivos ópticos y electrónicos.
- Comprensión de la linealidad y respuesta a bajas energías del sistema combinado de los subdetectores.

Basado en la experiencia de otras colaboraciones (CDF y ZEUS), la intercalibración de los módulos del calorímetro se realiza usando fuentes de Cs^{137} . Para obtener resultados fiables en el test de los módulos, el 12% de los módulos producidos (8 barriles centrales y 16 barriles extendidos) se calibrarán en el Test Beam.

La siguiente figura muestra el acelerador SPS usado en el Test Beam y cual es su posición dentro del acelerador LHC.

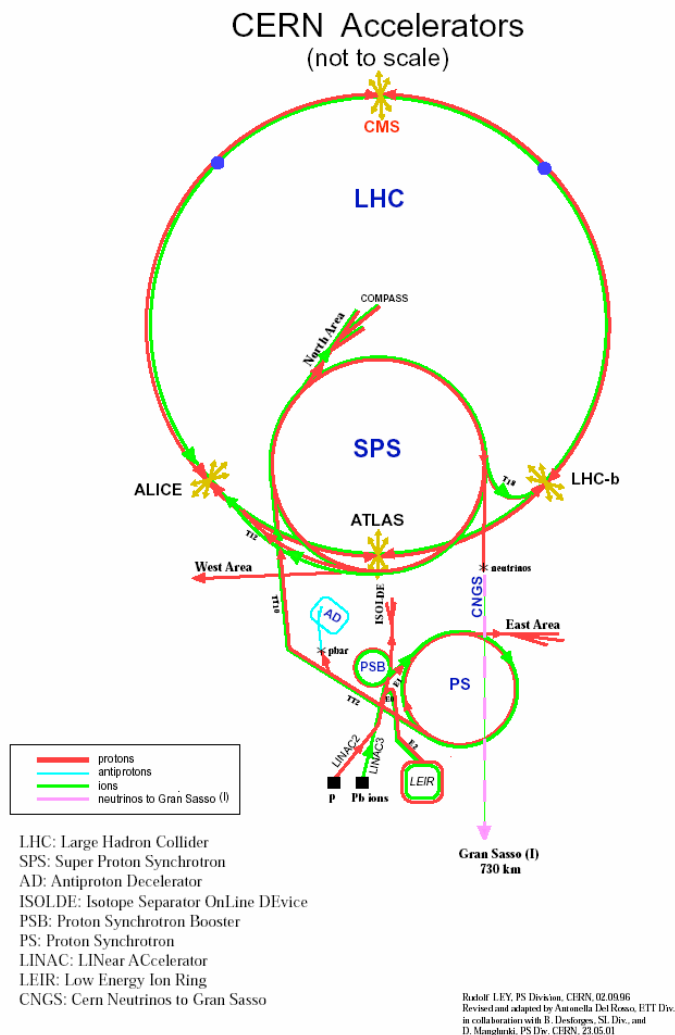


FIGURA 4.1 Posición del SPS usado en el Test Beam dentro del LHC.

2.1.1 Desarrollo Experimental

Las medidas se realizan en la línea del haz construida en el edificio H8 del Área Norte del SPS en el CERN [2]. Es un haz de propósito general de alta energía preparado para realizar los test. En este área el máximo momento del haz de protones es de 450 GeV/c y del haz de electrones de 200 GeV/c. El tamaño del haz es un punto de unos 2 cm de diámetro.

Los módulos del calorímetro están instalados en una mesa que se mueve horizontalmente, manteniendo el módulo perpendicular a la línea del haz. Pueden ser girados verticalmente, permitiendo el cambio de ángulo θ y accediendo a las torres con diferentes valores de η . La mesa también puede ser girada alrededor del eje horizontal, permitiendo tratar el ángulo de incidencia azimutal ϕ .

El programa de instalación usado desde los Test Beams del 2000 se puede observar en la siguiente figura, donde el Módulo del Barril 0 es el módulo inferior que está montado sobre la mesa, un Módulo de Barril de producción se coloca encima del anterior y dos Barriles Extendidos de producción en la parte superior del sistema. Así pues, dos posibles ángulos de incidencia pueden ser observados.

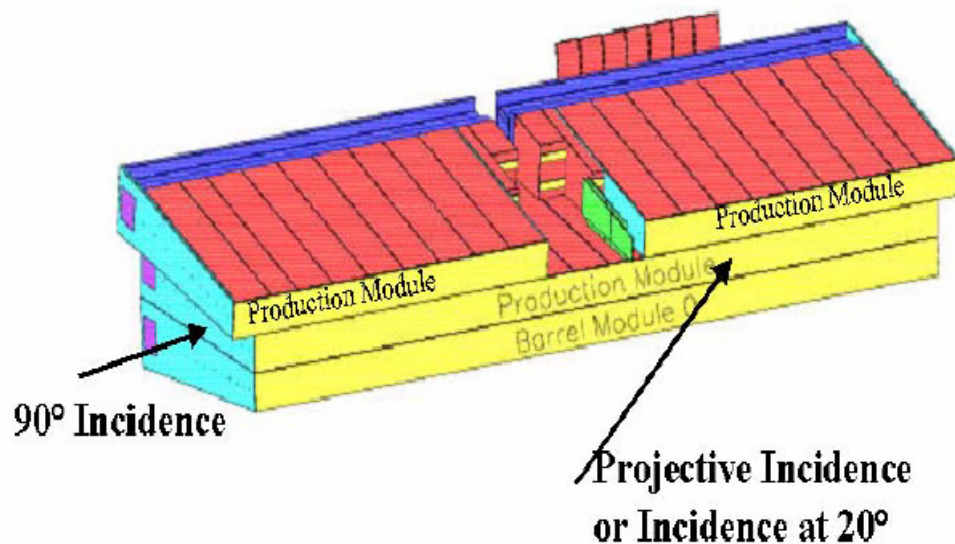


FIGURA 4.2 Instalación de los módulos en el Test Beam.

2.1.2 Medidas en el Haz del H8

El programa básico que se usa durante el Test Beam analiza y mide la respuesta del calorímetro con respecto a los hadrones, electrones y muones en un rango de energía de 10 GeV hasta 180 GeV. El plan de test para un periodo de trabajo incluye los siguientes pasos:

- Fijar la escala EM (ElectroMagnética) del calorímetro para cada celda expuesta al haz de electrones. Normalmente son necesarios electrones con 100 GeV para medir el factor EM.
- Los muones tienen una energía de deposición uniforme en todo el módulo, necesitando una única herramienta para verificar los datos obtenidos de las fuentes de Cs^{137} . Los muones con 180 GeV de energía entran en los módulos de TileCal con 90° de ángulo permitiendo una calibración de las constantes de las fuentes de radioactividad.

- Las señales de los piones y de los electrones son una función que depende de la energía del haz. El parámetro e/h (radio de la señal observada para los componentes del chorro de hadrones) se extrae de estas señales. Este parámetro caracteriza el calorímetro y puede ser útil después en la calibración de la respuesta de los jets (chorros de partículas que se producen en una colisión del acelerador).
- Medidas de los dispositivos electrónicos conectados al módulo, test frente a ruido y frente a un gran número de datos.

2.1.3 Calibración y Desarrollo de los Módulos

El objetivo de TileCal, junto con el de otros calorímetros de ATLAS, es determinar las energías de las partículas y jets que se extraen de las señales medidas en los mismos [3]. El punto principal es conseguir un conjunto de escalas de energía de calibración, como por ejemplo, el factor que relaciona la energía que se libera en el calorímetro con la señal digitalizada que se produce al final de la cadena de adquisición de datos.

Esto no es suficiente para un experimento de esta envergadura, así pues, tendremos que definir diferentes parámetros de calibración para posibles cambios en el desarrollo futuro del experimento. Por esta razón es necesario medir factores de calibración para cada canal de lectura, minimizar las variaciones que se produzcan y monitorizar todos los valores para corregir a tiempo los problemas.

El desarrollo de la calibración en TileCal se basa en diferentes factores y etapas. A continuación se detallan los parámetros más importantes y el diagrama usado en la calibración.

- El factor de calibración general es un parámetro muy importante para determinar el potencial físico del LHC. Para ello debemos determinar una escala absoluta de energía, la cual se realiza a partir de la escala de energía de los jets que se producen en el cruce de haz.
- La escala de energía de los jets se obtiene aplicando factores de corrección. Conceptualmente hay dos aspectos en la calibración de la energía de los jets que debemos tener en cuenta: el primero consiste en calibrar la energía de los jets con medidas experimentales como reproducción de la energía de cada partícula que lo forma, el segundo consiste en deducir la energía del patrón cuya fragmentación ha resultado en la medida del jet.
- La escala EM inicial se obtiene de la energía depositada en cada celda individual del calorímetro por los electrones en el Test Beam. Un sistema dedicado basado en una fuente móvil radioactiva de Cesio es lo que se usa como calibración de los módulos.
- Monitorización. Fundamental para observar los datos que se obtienen en cada una de las celdas.

La siguiente figura muestra el proceso de calibración de un módulo en TileCal.

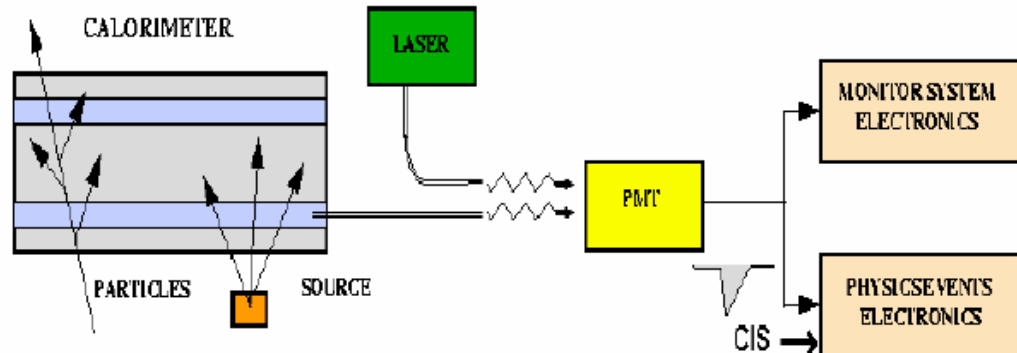


FIGURA 4.3 Diagrama de calibración en TileCal.

Durante la captura de datos en el LHC, se observarán continuamente en pantalla la señal producida en el calorímetro por las colisiones inelásticas protón-protón, también denominado Minimum Bias (explicado en capítulos anteriores). Los centelladores se excitan con emisiones de luz dadas por emisores radiactivos móviles.

Se conecta un láser a cada PMT para producir pulsos que simulan los pulsos del calorímetro. Un sistema de inyección de carga (CIS) genera una señal de carga como la que viene de los PMTs, la cual nos sirve para calibrar correctamente cada celda.

2.2 INSTALACIÓN DEL SISTEMA ROD INICIAL EN EL TEST BEAM

Se describe a continuación el trabajo, las medidas y los resultados obtenidos de la integración del prototipo en el sistema de adquisición de datos del detector TileCal [4].

El proceso de integración se dividió en dos etapas. La primera fue la instalación de todos los componentes del sistema ROD, la segunda fue la adecuación del software de adquisición de datos ya existente para incluir en el mismo el código necesario para el control del módulo, así como para hacer que el flujo de datos procedente del detector pasase por el mismo.

Una vez realizadas estas tareas, se adaptó el prototipo a la estructura del sistema de adquisición empleado con el detector. Como ya se ha comentado, el proyecto ATLAS no tiene prevista su puesta en marcha hasta el año 2007. Por ello, los sistemas de adquisición que actualmente se emplean para las pruebas de los subdetectores no son tal y como serán en el futuro, sino que tienden a su objetivo final a través de sucesivos refinamientos. Por ello, el prototipo tal y como se diseñó no podía ser empleado en el sistema de adquisición existente.

Con el prototipo y el software adaptados a las condiciones del sistema, se tomaron datos del detector y se procesaron obteniéndose los resultados que posteriormente se describen.

El sistema ROD que se instaló en el Test Beam de TileCal es el siguiente:

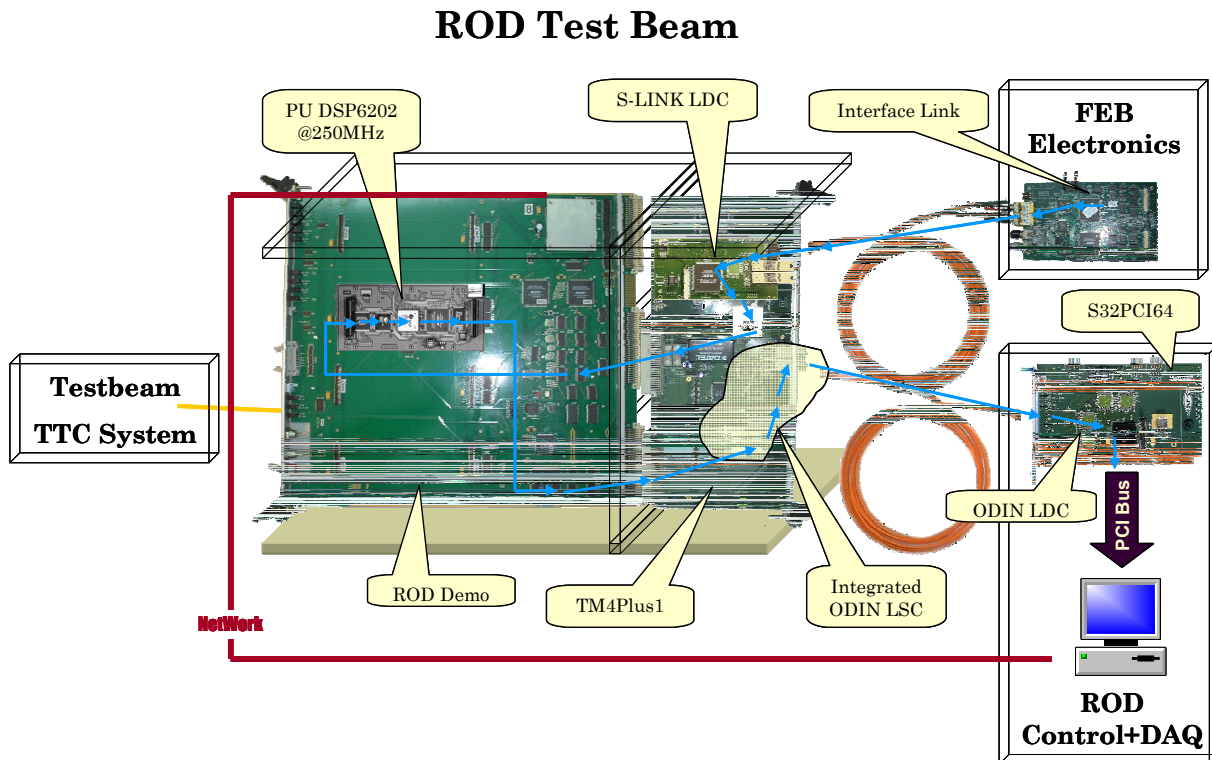


FIGURA 4.4 Sistema RODDemo instalado en Test Beam.

Como se puede ver, sigue la estructura definida en el capítulo 2 de esta Tesis Doctoral con la salvedad de que los datos no son enviados al siguiente nivel de trigger. Los datos de salida se mandan por medio de una tarjeta PCI a un PC en el cual almacenamos los eventos filtrados para obtener resultados que nos confirmen la bondad de nuestro sistema.

Los elementos instalados en el Sistema ROD Inicial para el Test Beam son:

- ROD Controller. Realiza el control de todo el sistema instalado.
- TTCvx, TTCvi. Control del TTC del Test Beam de TileCal.
- ODIN double LDC card. Encargada de la recepción de los datos del front-end.
- 32 bit S-LINK to 64 bit PCI. Recepción y almacenamiento de los datos de salida del ROD.
- DSP6202 PU. Implementación algoritmos de filtrado.
- Módulo de Transición. Objetivo principal de estudio en el capítulo 2.
- RODDemo. Motherboard del sistema ROD.

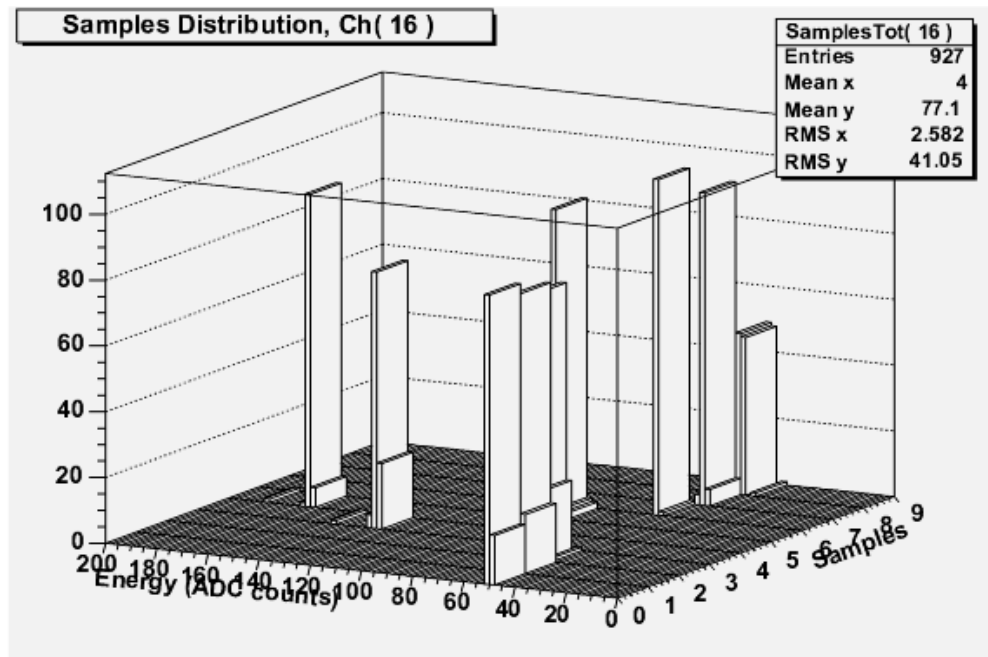
2.3 RESULTADOS DEL SISTEMA ROD INICIAL EN EL TEST BEAM

Son varios los resultados de las pruebas de Test Beam, algunos relacionados con el buen funcionamiento del detector y otros con la funcionalidad del sistema de adquisición.

Desde el punto de vista de la funcionalidad del sistema el principal resultado de las pruebas es la comprobación del correcto funcionamiento del sistema y de la corrección del diseño en vista al prototipo final adecuado para el experimento ATLAS/LHC. Las diferentes pruebas realizadas nos permitieron establecer una serie de requisitos para el prototipo final a instalar en el sistema de adquisición definitivo, como se describe en el siguiente punto. Por otra parte, el módulo implementado permitió la realización de pruebas de diferentes algoritmos de preprocesado de datos tomados directamente del detector y en tiempo real.

En cuanto a los aspectos relacionados con el buen funcionamiento del detector, mediante el sistema descrito en el epígrafe anterior se realizaron una serie de toma de datos en la que en total se han almacenado más de 300 ficheros para su posterior análisis. La información almacenada corresponde a los valores de energía depositada por las partículas a medida que atraviesan el detector.

La figura siguiente nos muestra histogramas típicos de esta distribución en diferentes puntos del detector [5]. Estos histogramas están realizados a partir de datos de inyección de carga.



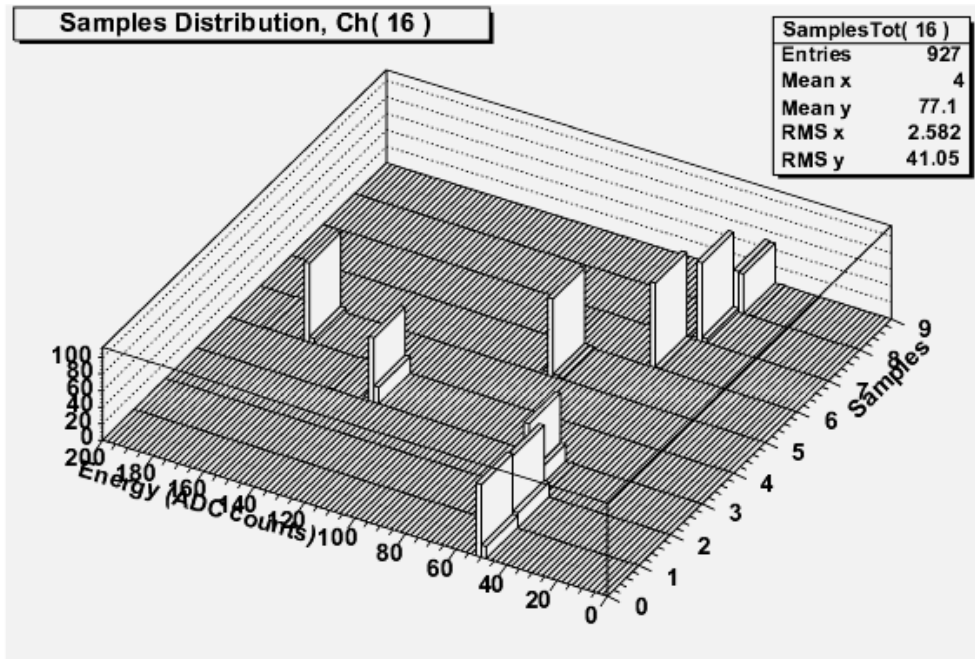


FIGURA 4.5 Distribución de la energía (en GeV) depositada en diferentes puntos del detector.

Así mismo, tenemos datos filtrados con los dos algoritmos explicados en este trabajo. Se trata de reconstrucciones de energía de celdas del calorímetro.

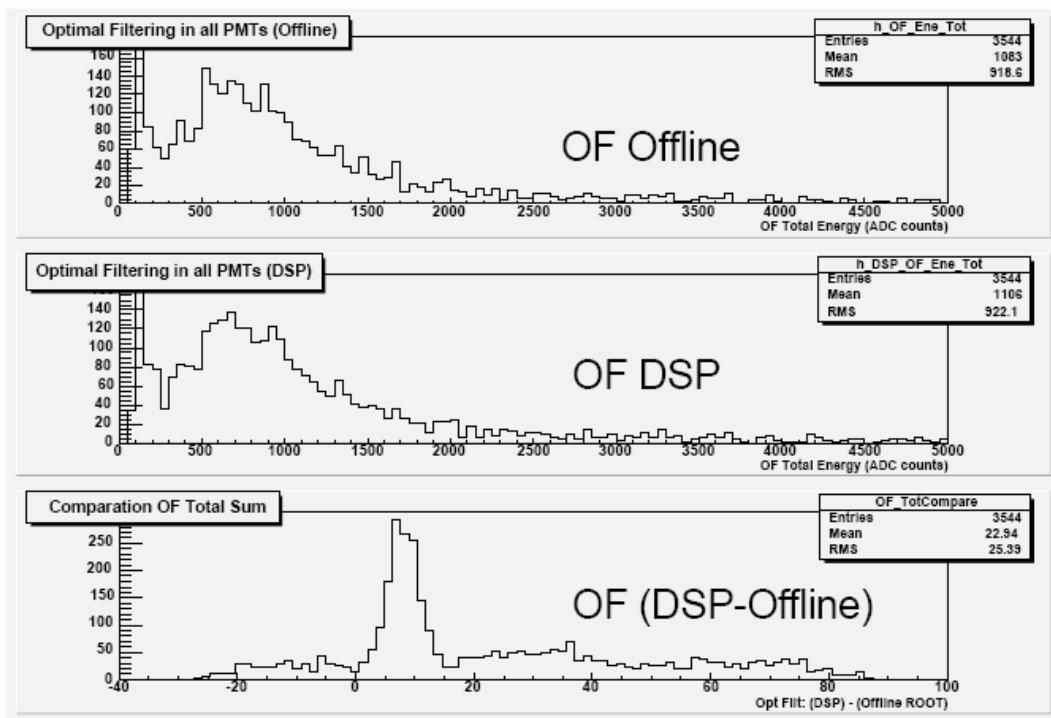


FIGURA 4.6 Reconstrucción de energía con Optimal Filtering.

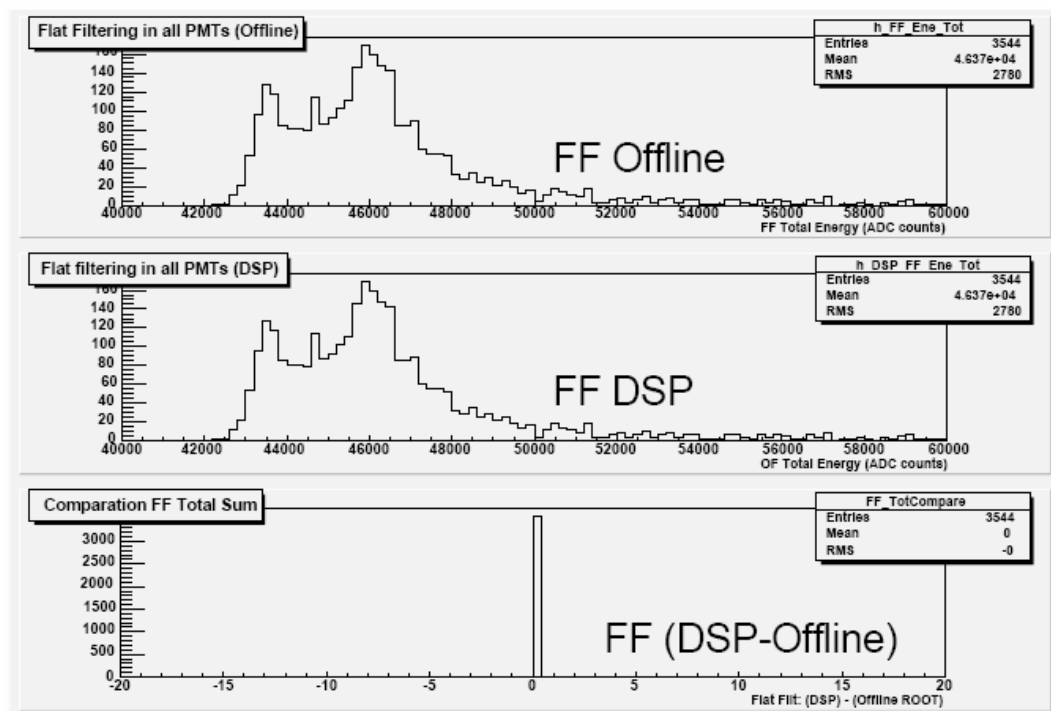


FIGURA 4.7 Reconstrucción de energía con Flat Filtering.

En el caso de Optimal Filtering se puede observar una pequeña diferencia entre las dos reconstrucciones (Offline y Online), lo cual representa un error para la reconstrucción de todos los canales. Este error es asumible y no representa ningún problema para el envío de los datos hacia el siguiente nivel de trigger. Eso sí, es mejorado en la Motherboard final del ROD al usar nuevos DSPs para su tratamiento que nos permiten realizar multiplicaciones de 32 bits.

En el caso de Flat Filtering vemos como no existe ningún error entre la reconstrucción realizada Online y la Offline. Esto es debido a que se trata de un filtrado más sencillo que solo suma samples enteros de 10 bits, con lo cual no se pierde resolución al realizar la reconstrucción en los DSPs.

3. DESCRIPCIÓN DE LAS PRUEBAS DEL SISTEMA ROD FINAL Y RESULTADOS

En este apartado abordaremos el banco de test montado en el CERN para el desarrollo de los tests del Sistema ROD Final. Haremos, como en el apartado anterior, especial hincapié en el Módulo de Preprocesado de Datos; en el caso anterior el Módulo de Transición Activo y en este caso la Optical Multiplexer Board [6].

Para ello, en primer lugar explicaremos los tests previos que se realizaron a la OMB antes de su integración en el banco de test, luego describiremos en qué consiste y qué elementos componen este banco de test. Más tarde hablaremos de la aplicación creada para realizar las comprobaciones necesarias desde un PC y por último de la instalación de nuestros módulos y de los resultados obtenidos.

3.1 DESCRIPCIÓN DE LOS TESTS PREVIOS DE LA OMB

Antes de entrar a describir el banco de test completo donde se realizaron los tests finales, debemos comentar los tests particulares que se realizaron a la Optical Multiplexer Board. Como se ha señalado en el capítulo anterior, el diseño de este tipo de módulos se había realizado hasta este momento de una manera estructural. Se valoraba la necesidad, se realizaba un estudio de componentes, se construía un diagrama de bloques que definiera el módulo y se enviaba su realización al CERN.

Como acabamos de comentar, hasta la realización de la OMB, era en el CERN donde se creaba la tarjeta, se definían sus esquemáticos, se rutaba, se analizaba, se mandaba a fabricación y se testeaba su funcionamiento eléctrico. La programación, su testeo en la cadena de adquisición de datos, su correcto funcionamiento, su integración con el resto de módulos y protocolos y la obtención de resultados volvía a correr de nuestra parte.

Así se ha trabajado con la Motherboard del ROD (tanto en su versión inicial como en su versión final) y con el Módulo de Transición Activo. La gran novedad de este trabajo ha sido la realización completa de la Optical Multiplexer Board. De ahí que antes de describir su integración en el banco de test queremos referirnos a los tests eléctricos y funcionales que se realizaron.

Para ello, antes de colocar la tarjeta en la crate e incorporarla al sistema de test general se alimentó externamente e independientemente. Para ello, en el diseño de la OMB se dejaron conectores externos para este tipo de tests. Tenemos conectores para todas las alimentaciones y conectores para obtener datos de las FPGAs. Algo que nos da una versatilidad en el test muy importante.

Se fueron conectando todas las alimentaciones y se fueron comprobando eléctricamente cada uno de los componentes presentes en la OMB. El resultado favorable de este primer test nos llevó a una primera programación de las FPGAs. Para ello se usó el camino de test y de conexión entre ellas, comprobando que los datos que se generaban en una de ellas eran recibidos y rutados a las demás.

Superados los tests eléctricos y de programación de las FPGAs se decidió su integración en la crate. Comprobando ahora que la alimentación tomada desde VME era correcta y la tarjeta no presentaba problemas eléctricos.

Las siguientes imágenes muestran parte de estos tests:

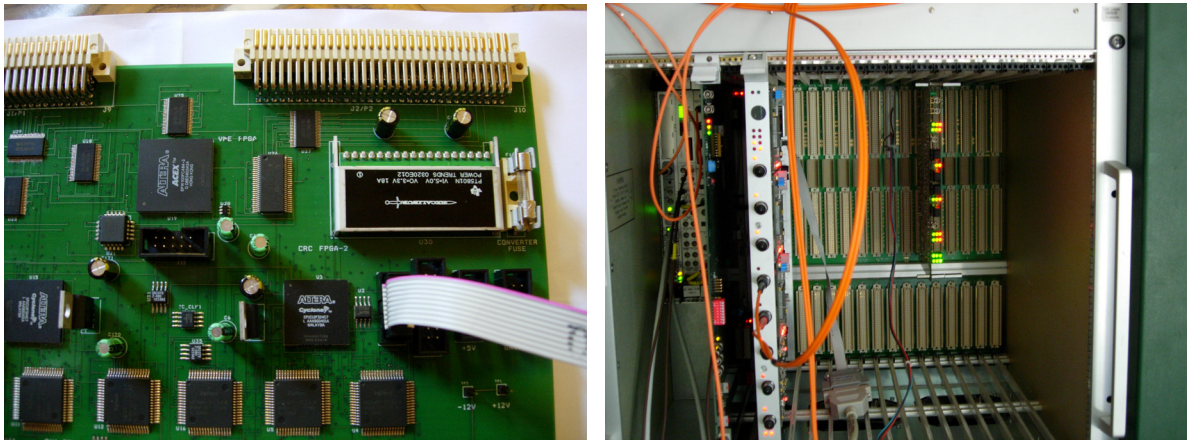


FIGURA 4.8 Fotografías de los primeros tests realizados a la OMB.

El siguiente paso, por tanto, era incluir los protocolos de comunicación en las FPGAs de CRC y VME. Para ello se usó un canal de la tarjeta como inyector de datos y el otro como multiplexor de los mismos. Incluso se utilizó otra OMB (en total se han construido cuatro tarjetas) como inyectora de datos, comprobando así la compatibilidad existente entre todas ellas.

Además, se estableció un nuevo test que nos permitiera conocer la calidad de los datos que se sacaban del deserializador antes de llegar a las FPGAs. Las siguientes imágenes nos muestran las conexiones que se realizaron para este test.

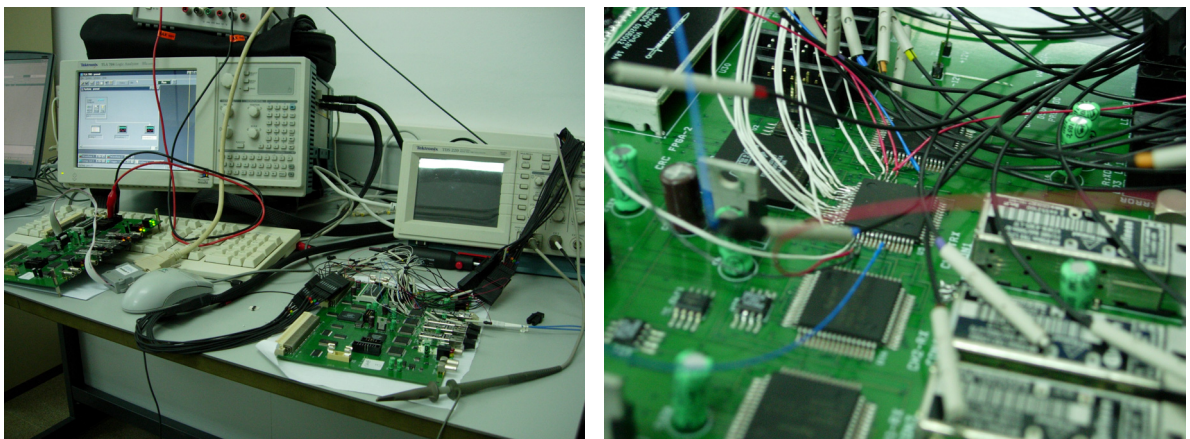


FIGURA 4.9 Fotografías de los tests de datos realizados a la OMB.

Antes de conectarnos a la Motherboard del Sistema ROD se estudiaron los datos mediante la conexión a un analizador lógico. Se tomaron diferentes paquetes de datos comprobando el inicio y final del evento a analizar, así mismo se comprobaron las señales de control y los relojes usados. En la siguiente figura se puede observar la captura de dichos datos en el analizador.

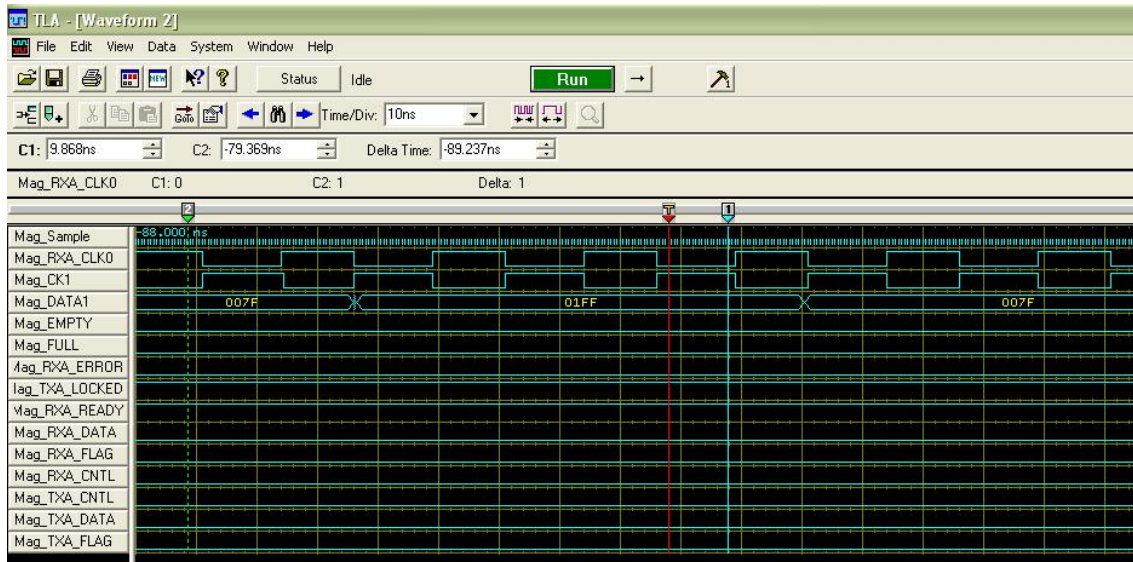


FIGURA 4.10 Datos tomados con el analizador lógico.

Estos análisis de los datos mediante el analizador nos confirmaron lo que en el Capítulo anterior habíamos visto, no existen problemas de integridad de la señal en la Optical Multiplexer Board. No existen problemas de datos corruptos ni retardos en los relojes.

A la vista de los resultados obtenidos con la OMB tanto en modo multiplexor como en modo inyector fuera de la crate, se incorporó la misma al Banco de Test diseñado para comprobar la correcta adquisición de datos por parte de todo el Sistema ROD.

3.2 INSTALACIÓN DEL SISTEMA ROD FINAL EN EL BANCO DE TEST

El proceso de instalación del Sistema ROD Final en el banco de test instalado en el laboratorio del CERN [7] sigue la misma estructura que el instalado para el Sistema ROD Inicial en el Test Beam de TileCal. El hecho de realizar estos tests en el laboratorio y no en el Test Beam fue debido a las necesidades de realizar pruebas en un momento en el que no estaba en funcionamiento el Test Beam.

Estamos a la espera de poder integrar todo este sistema en el Combined Test Beam [8] que está preparando la colaboración para este verano. Ya no será un Test particular de cada uno de los subdetectores de ATLAS sino que se tratará de un primer test de todos los subdetectores. Un paso más hacia el inicio de este experimento.

Debido a la imposibilidad de tomar datos del Test Beam, se hace imprescindible el usar la Optical Multiplexer Board como Inyectora de Datos. Gracias a esta funcionalidad conseguimos enviar eventos hacia otra OMB para que sean tratados y multiplexados y después hacia la Motherboard del ROD para después ser visualizados y almacenados en un PC.

El banco de test desarrollado se asemeja bastante al que hemos explicado en el apartado 2.2 con la salvedad de no recibir datos del Front End de TileCal sino de una Optical Multiplexer Board que simula esta función. Además se han realizado unas aplicaciones informáticas específicas para poder realizar cambios en la adquisición de los datos y para la visualización y comprobación de los mismos. El Sistema ROD instalado es el siguiente.

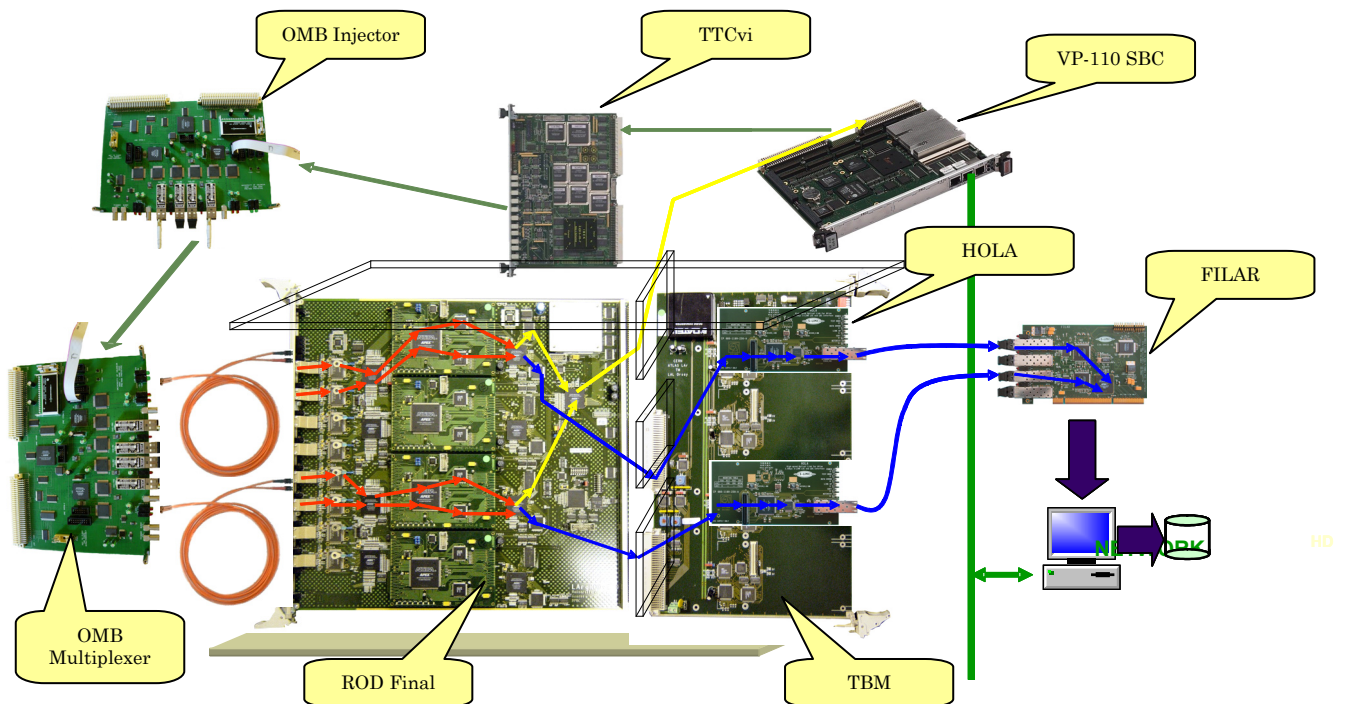


FIGURA 4.11 Sistema ROD Final instalado en el Laboratorio del CERN.

Podemos observar que la estructura de nuestro test sigue el camino definido en el Capítulo 3 de esta Tesis. La única diferencia está en la emisión y recepción de datos, aquí se generan mediante una OMB (trabajando como Front End) y se envían a un PC (trabajando como ROB) donde se almacenan para su posterior estudio.

Los elementos instalados en el Sistema ROD Final para el test de estos módulos son:

- ROD Controller (VP-110 SBC). Realiza el control de todo el sistema instalado.
- TTCvi. Generación de señales de TTC para la sincronización del sistema.
- Optical Multiplexer Board Injector. Encargada de enviar datos como Front End, estos se pueden generar internamente en las FPGAs o se pueden enviar por VME a la OMB.
- Optical Multiplexer Board Multiplexer. Recibe los datos de la otra OMB, los analiza, decide qué fibra lleva datos correctos o si estamos en inyección cómo deben transmitirse y los envía al ROD.
- ROD Final. Motherboard del Sistema ROD.
- PUs. Implementación algoritmos de filtrado.
- Módulo de Transición Pasivo. Incorpora tarjetas HOLA [9] para el envío de datos al siguiente nivel.
- Filar. Recepción y almacenamiento en el PC de los datos de salida del ROD.

El medio de transmisión usado para el envío y recepción de datos es fibra óptica, igual que ocurrirá en TileCal cuando esté en funcionamiento. Con la información guardada en el PC se pueden analizar los datos offline, comparando estos con los enviados.

Tenemos un control total sobre la cadena de datos, pudiendo en cualquier momento verificar lo que está ocurriendo con los mismos. En el caso de estos tests nos ha importado más comprobar que los datos no se pierden ni se corrompen que su posterior análisis físico.

Para ello se han desarrollado unas aplicaciones informáticas que explicamos a continuación.

3.2.1 Aplicaciones XTestROD y XFilar

Para la caracterización eléctrica de los módulos presentes en el Sistema ROD Final y para el test de los mismos se han desarrollado dos aplicaciones informáticas, XTestROD y XFilar [10]. Estos programas han sido desarrollados en C y C++. Nos permiten escribir y leer en los registros de los diferentes módulos y configurar los diferentes modos de funcionamiento del Sistema ROD Final.

Usando este software independiente, la adquisición de datos se dirige a través de del Bus VME y se pueden desarrollar y simular todas las características que nuestro sistema tiene que realizar en TileCal.

XTestROD debe correr en el ROD Controller y nos da acceso a través del Bus VME a todos los módulos presentes en la Crate del Sistema ROD Final.

XFilar debe correr en el PC que se esté usando para adquirir los datos que pasan por el Sistema ROD Final. Nos permite controlar las tarjetas de adquisición de datos montadas en el PC (Filar).

Ambos programas usan el mismo hardware y las librerías de software de bajo nivel desarrolladas en el Sistema de Adquisición de ATLAS. Junto a estas librerías se han desarrollado las librerías necesarias y complementarias para que nuestro sistema funcione de forma autónoma y dentro del futuro Test Combinado del detector.

XTestROD y XFilar están organizados como paquetes estándar de CMT (*configuration management environment*). Se trata de un proyecto académico que da soporte a todo el software desarrollado en el contexto de los experimentos de Física de Altas Energías. Estos paquetes han sido compilados usando la versión 3.2 del compilador GCC y de la versión 01-00-00 del ATLAS Dataflow and Online Software.

XTestROD a su vez también usa los siguientes paquetes desarrollados por nosotros para estos tests.

- *rod_demo_rcc*
- *rod_demo_putius_rcc*
- *bit3_rcc*
- *TileVmeROD*

Una imagen de estas aplicaciones se puede observar a continuación.

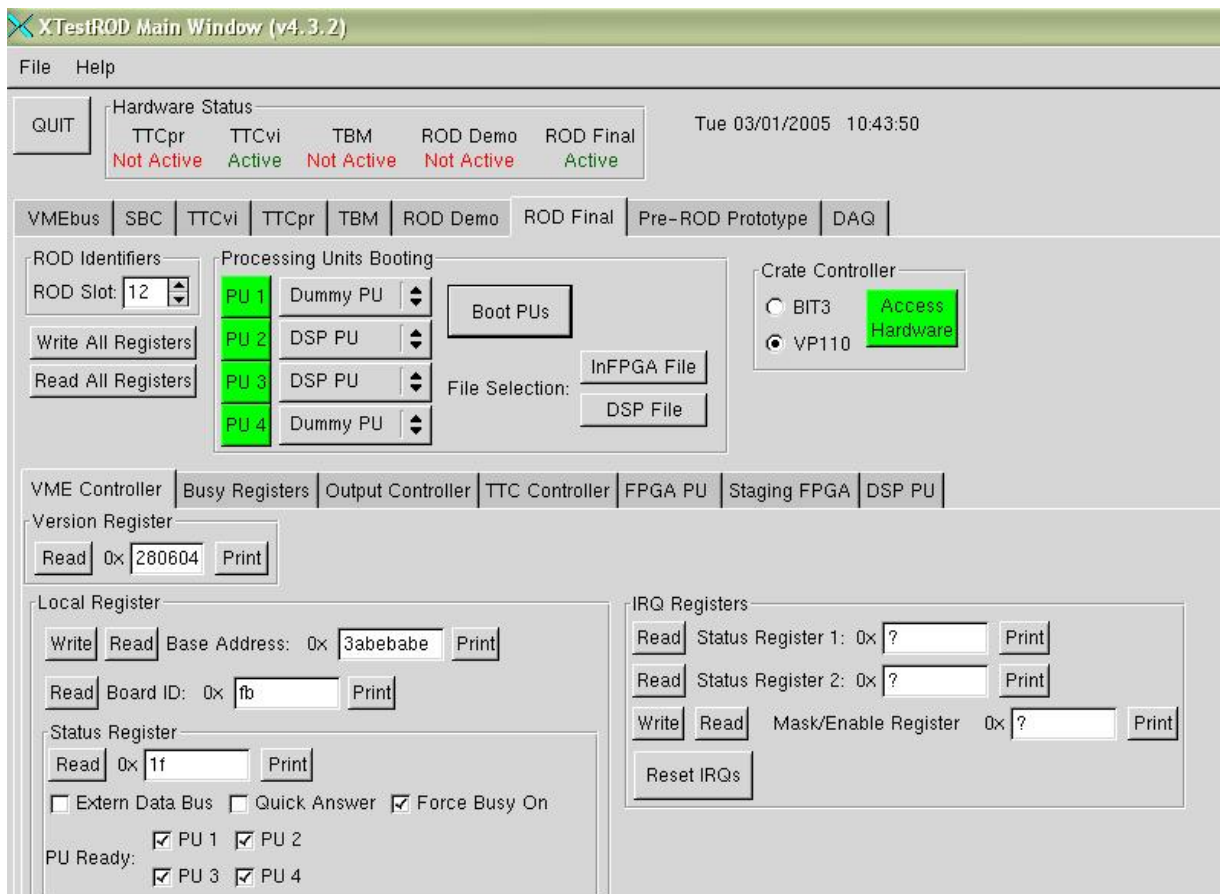


FIGURA 4.12 Aplicación XTestROD.

3.3 RESULTADOS DEL SISTEMA ROD FINAL EN EL BANCO DE TEST

La implementación del módulo de preprocesado y agrupamiento de datos (OMB) demuestra que es posible la realización de un sistema que mejore el rendimiento de la adquisición introduciendo preprocesado en la cadena. En este sentido, las diferentes pruebas realizadas han permitido establecer una serie de requisitos para el prototipo final a instalar en el sistema de adquisición definitivo, como se describe en el próximo capítulo.

Por otra parte, el módulo implementado permite la realización de pruebas mediante el envío de diferentes eventos y el preprocesado de los mismos. Se han introducido datos de diferentes formas, en un principio fueron eventos simples donde no había física y sólo era un contador para tratar y verificar el correcto desarrollo de los mismos por todos los módulos. Así mismo, se verificaba que el formato de los datos mantenía la estructura de ATLAS y se detectaban los inicios y finales de evento correctamente.

Para ello se tomaron varios miles de eventos mediante XTestROD, se almacenaron los datos tomados vía VME y se verificaron si eran correctos o no. La imagen siguiente nos muestra la toma de eventos desde XTestROD.

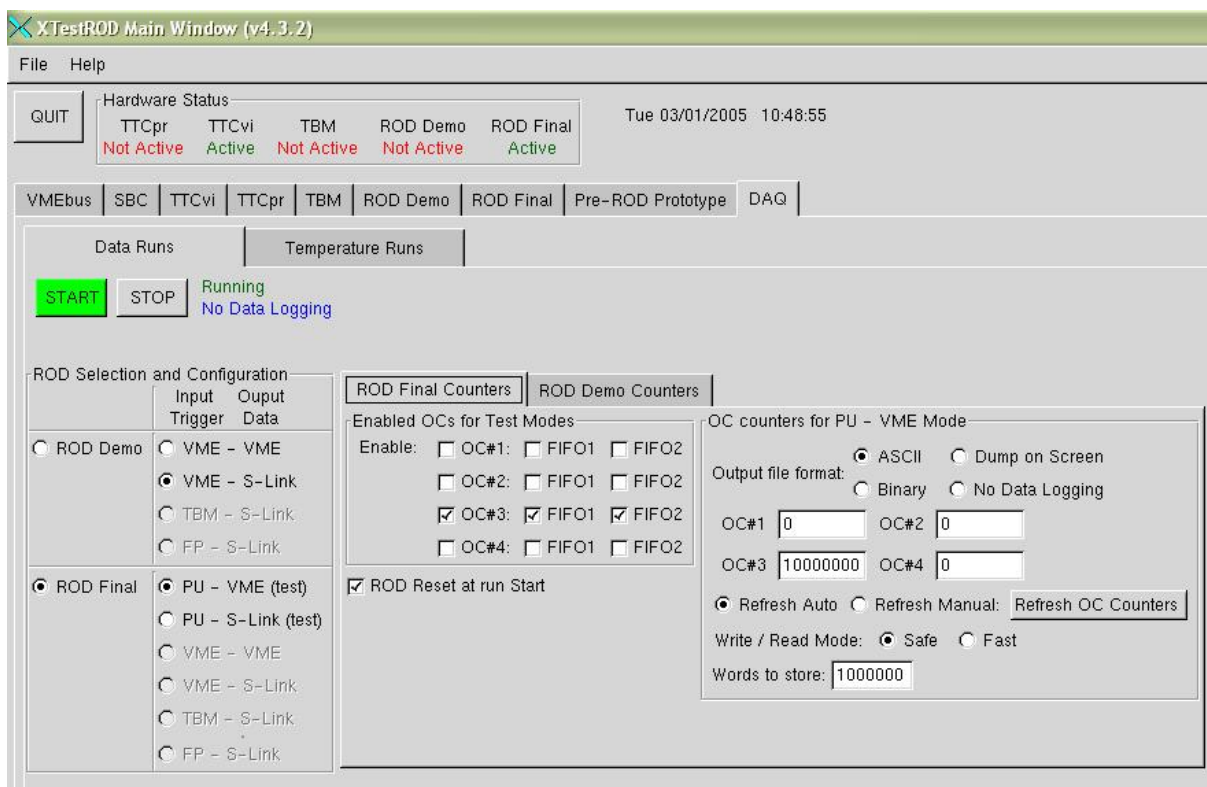


FIGURA 4.13 Datos tomados por VME.

Se puede observar como se tomaron 10 millones de eventos a través de una de las entradas del ROD. Comprobando la no presencia de errores en la transmisión. La verificación se realiza de manera offline en este modo de trabajo.

Una vez observado que no existían errores en los eventos enviados, se procedió a enviar eventos con errores en la paridad, en el CRC-16 y en el CCI-TT-CRC16. De esta forma podíamos comprobar si se detectaban los errores que nosotros introducíamos de manera voluntaria en los eventos.

Así mismo, comprobábamos después si los eventos que llegaban al ROD eran los de las fibras correctas. Para ello se usó la aplicación XFilar que nos permitía comprobar de manera online si los eventos recogidos presentaban o no errores.

Estas dos imágenes nos muestran las dos aplicaciones usadas y cómo se detectaron o no los errores que se introducían.

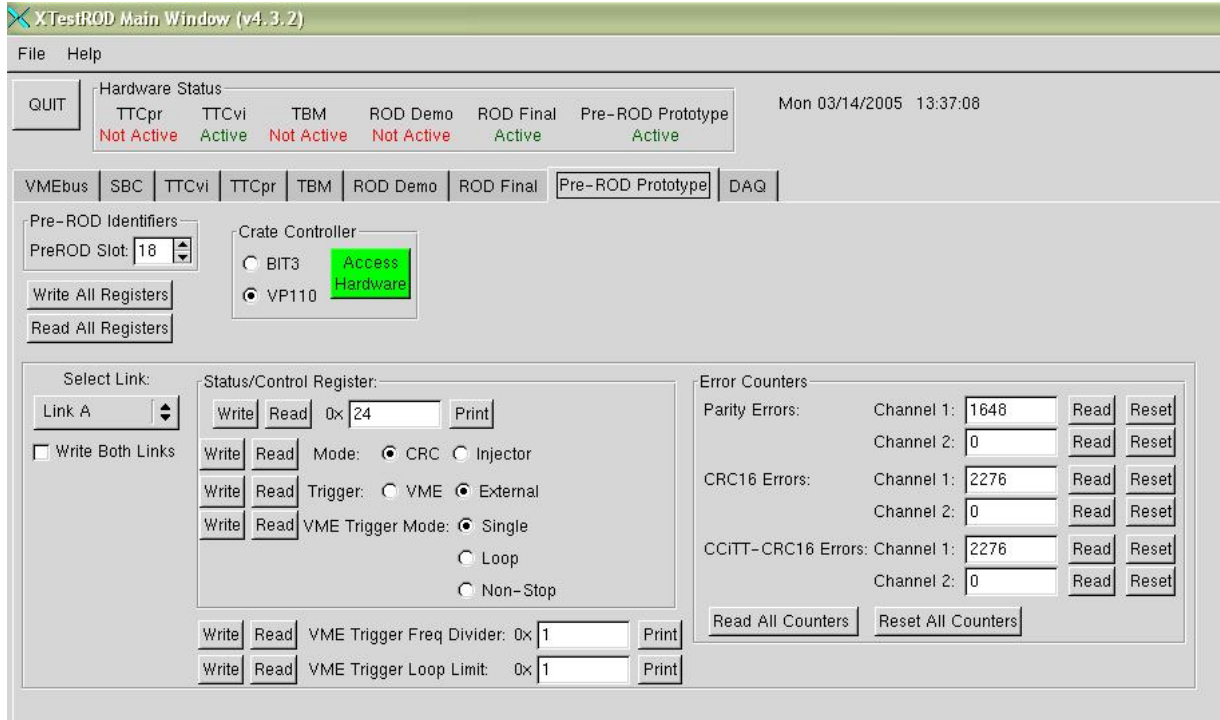


FIGURA 4.14 Comprobación de errores mediante XTestROD.

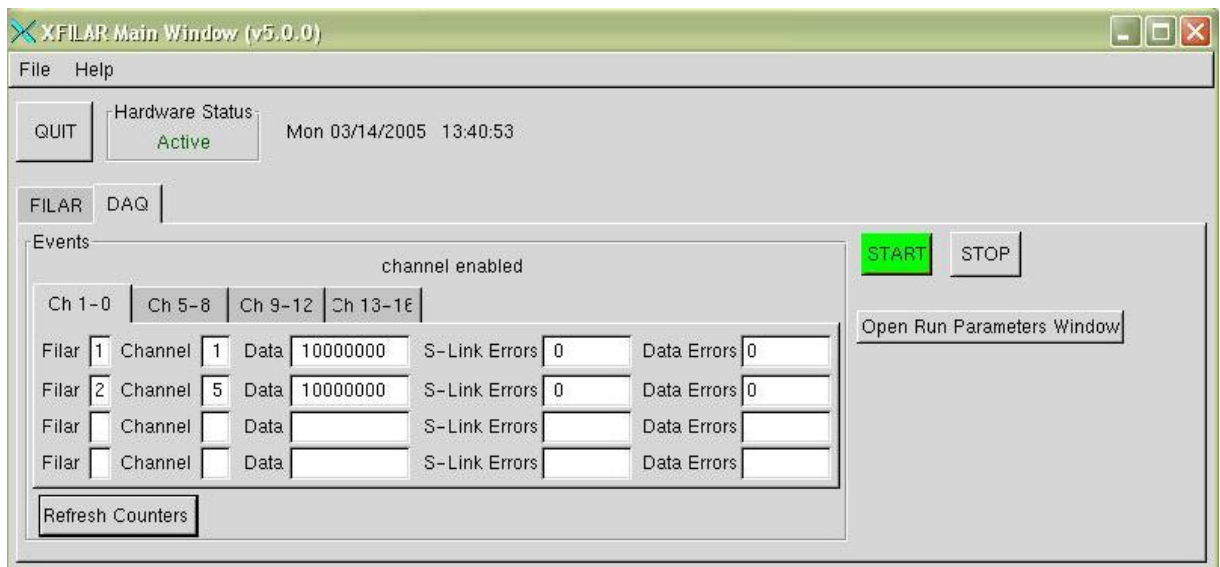


FIGURA 4.15 Comprobación de errores mediante XFilar.

Con la ayuda de XTestROD y XFilar, se ha podido estudiar la Optical Multiplexer Board como Inyectora y Multiplexora de datos dentro del Sistema ROD Final. Ante los resultados satisfactorios que se han obtenido, se ha comenzado a desarrollar la OMB 9U, tarjeta final con funcionalidad para tratar 8 canales a la vez.

Por tanto, esta tarjeta estará presente dentro del sistema DAQ/Trigger de ATLAS. Los resultados de la simulación proporcionan estadísticas sobre los errores que se tienen en el envío de los datos, los errores de paridad y CRC en los canales y del funcionamiento general del sistema.

Observando los datos tomados y los resultados obtenidos con las aplicaciones informáticas se comprueba la viabilidad de la Optical Multiplexer Board y su necesidad dentro del Sistema ROD Final, a su vez proporciona información de gran valor para el diseño y desarrollo de la OMB 9U Final.

El trabajo de modelización ha de continuar con el objetivo de refinar los algoritmos utilizados, e incluir nuevos estudios sobre los datos de Física junto al resto de detectores que se integrarán dentro de ATLAS.

Este desarrollo se realizará gracias al Test Beam Combinado de este verano.

4. BIBLIOGRAFÍA

- [1] TileCal Collaboration, *TileCal Technical Design Report*, CERN/LHCC 96-42, 1996.
- [2] ATLAS Collaboration, *Summary Setup*, ATL-COM-LARG-97-021, 1996.
- [3] Norriella, O., *Response of the ATLAS Tile Calorimeter to a radioactive source and muons*, Trabajo de Investigación, 2003.
- [4] Castelo, J., *Results of RODdemo testbeam installation*, General TileCal Meeting, <http://agenda.cern.ch/fullAgenda.php?ida=a032221>, 2003.
- [5] TileCal Collaboration, *TileCal Analysis Manual*, The Hadron Performance Section, 1999.
- [6] Torres, J., Soret, J., González, V., Sanchis, E., Martos, J., Gómez, J. A., *Optical Multiplexer Board for TileCal Data Redundancy*, CERN-LHCC-2004-030, 2004.
- [7] Castelo, J., et al., *TileCal ROD Hardware and Software Requirements*, ATL-COM-TILECAL-2005-002, 2005.
- [8] The ATLAS CTB Team, *The 2004 ATLAS Combined Test Beam*, ATL-ENEWS-2004-021, 2004.
- [9] ATLAS Collaboration, *HOLA S-LINK Hardware Specification*, ECP Division. CERN, <http://hsi.web.cern.ch/HSI/s-link/devices/hola/>, 2003.
- [10] Poveda, X., *Standalone Software for TileCal ROD Characterization and System Tests*, ATL-COM-TILECAL-2004-012, 2004.

CAPÍTULO
5

CONCLUSIONES Y TRABAJO FUTURO

1. CONCLUSIONES	1
2. TRABAJO FUTURO	2
2.1 Tests Combinados del Sistema ROD Final	3
2.2 Tarjeta Multiplexora 9U	4

1. CONCLUSIONES

Esta Tesis comenzó tras la observación de que en el sistema de procesado de datos del subdetector TileCal se necesitaba un sistema intermedio de adquisición de datos que fuese fiable, que pudiese procesar los datos y que trabajase en tiempo real.

Así nació el prototipo del Sistema Read Out Driver (ROD) y así empezó el desarrollo de los módulos que lo componen (Motherboard y Módulo de Transición). Del desarrollo del Módulo de Transición y de los algoritmos implementados en él, hemos centrado nuestra primera parte de la Tesis.

Con la experiencia adquirida en la realización de este Sistema ROD Inicial se ha desarrollado el Sistema ROD Final. Nuestro trabajo se ha desarrollado casi en exclusiva en el Módulo de Preprocesado de Datos, la Optical Multiplexer Board. Módulo que nació desde la necesidad del subdetector TileCal, manteniendo la estructura del Módulo de Transición y que ha finalizado con los tests y desarrollos explicados en los capítulos 3 y 4 de esta tesis.

Las principales conclusiones del trabajo se han ido presentando a lo largo del mismo conforme se iban obteniendo. No obstante, podemos ahora resumirlas de manera más concreta en los siguientes puntos:

- Se han desarrollado los prototipos que componen el RODDemo de TileCal, sus características, su necesidad, su funcionamiento y su desarrollo. Así mismo, hemos visto su localización dentro del experimento LHC.
- Se ha realizado un estudio de todos los elementos que rodean la adquisición de los datos para conseguir incluir el módulo sin grandes cambios en la estructura principal. Adecuándonos en cada momento a las exigencias que un proyecto de este calibre nos han dado.
- Se han obtenido las expresiones que nos permiten realizar un filtrado de los datos obtenidos y ofrecer un formato de salida de los mismos fiable y comprensible para el siguiente elemento de la estructura de adquisición de datos.
- Se ha introducido una mejora en la recepción y en el envío de los datos gracias a la introducción de las FPGAs. Su versatilidad y capacidad nos han permitido adaptar perfectamente los prototipos y las versiones finales en el sistema.
- Para la verificación experimental de estos resultados se ha montado el prototipo en el Test Beam del CERN, en concreto en el del subdetector TileCal que se está construyendo para el detector ATLAS.
- Se han realizado las pruebas en los bancos de test disponibles actualmente, y se ha comprobado que la introducción de este módulo introduce una mejora en la reconstrucción de la energía depositada en las celdas del calorímetro, lo que valida el modelo desarrollado. Además, este módulo ha demostrado cumplir con la funcionalidad requerida para el subdetector en las fases de test y calibración del mismo antes de la construcción final.

2. TRABAJO FUTURO

A la vista de los resultados el trabajo futuro se plantea en las siguientes direcciones:

- Desarrollar un test combinado con el resto de subdetectores del sistema ROD final con la experiencia adquirida en este desarrollo y mejorando la integración de los componentes así como, la inclusión de zonas de test para verificar con facilidad las implementaciones realizadas en los dispositivos del sistema.
- Desarrollar nuevas implementaciones de los algoritmos desarrollados en estos sistemas para mejorar la eficiencia y rapidez de los mismos. En vista de los resultados obtenidos, su utilización en el sistema de adquisición de datos final del detector será fundamental.
- Profundizar en los desarrollos de los protocolos de comunicación del CERN, incluyendo las nuevas tecnologías y los nuevos dispositivos. Además de estudiar y desarrollar una programación de las FPGAs vía VME, lo cual nos permitiría trabajar en cualquier nueva implementación sin tener que trasladarnos al CERN ni tener que parar el desarrollo de los RODs.
- Realizar una nueva aplicación que nos permita obtener una visión remota del funcionamiento de cada uno de los componentes de la Optical Multiplexer Board. Una manera de realizar un test profundo de la tarjeta de manera externa.
- Estudiar y desarrollar una nueva Optical Multiplexer Board de mayor tamaño para el multiplexado de los datos recibidos y la comprobación de la redundancia de datos como paso previo a la llegada de los mismos a la Motherboard del ROD.

Los siguientes dos apartados muestran las líneas futuras ya en fase de investigación y desarrollo.

2.1 TESTS COMBINADOS DEL SISTEMA ROD FINAL

ATLAS ha desarrollado un Combined Test Beam para analizar qué ocurrirá cuando se ponga en funcionamiento el detector. Hasta estos momentos, cada uno de los subdetectores de ATLAS ha trabajado de manera independiente y será en estos Tests Combinados donde se podrá observar la fiabilidad de cada uno de los sistemas.

Una porción de cada subdetector del barril central y una del detector de muones que forma la tapa de ATLAS se colocará para su testeo. Los objetivos que se persiguen son los de tener una pre-comisión de los elementos finales y un estudio del funcionamiento del detector en un entorno más realista.

La figura 5.1 nos muestra la disposición de estas porciones de los subdetectores de ATLAS.

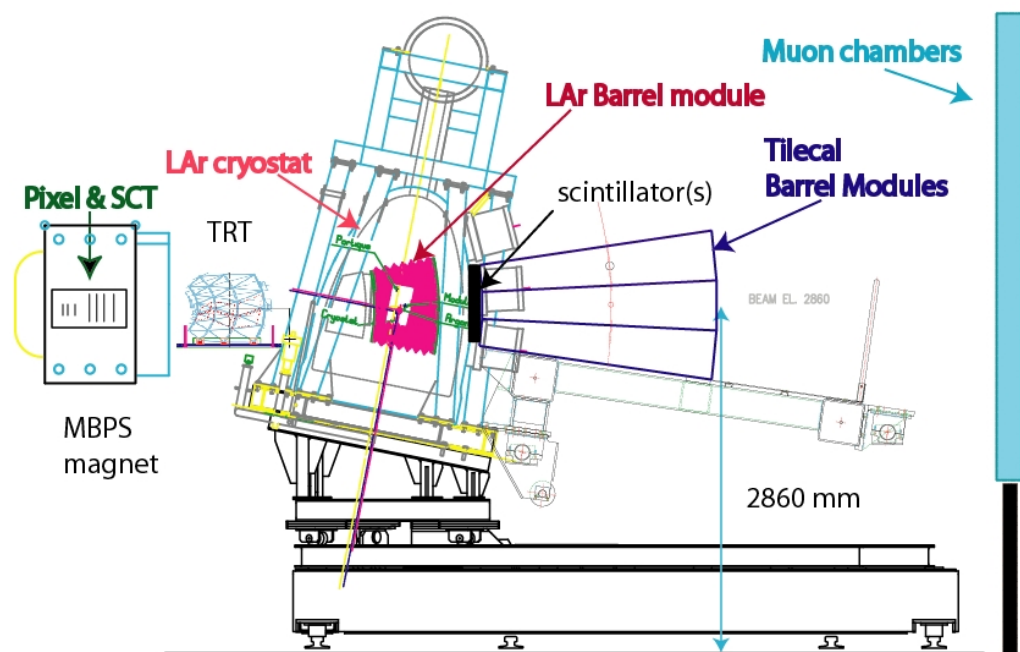


FIGURA 5.1 Diagrama del Combined Test Beam.

Este sistema será testeado con diferentes haces de partículas (piones, electrones, muones y fotones) con diferentes energías y polaridades, el rango será desde 1GeV hasta 350 GeV. Nos proporcionará una oportunidad única para evaluar el funcionamiento del detector ATLAS, la identificación de las partículas y las medidas que se realicen sobre ellas. Es una manera de comprender en mayor profundidad el detector.

Hablando en términos del trabajo desarrollado en esta Tesis, nos permitirá observar el funcionamiento de nuestro Sistema ROD Final en un entorno lo más parecido a las condiciones finales donde actuará. Será una prueba definitiva donde se consolidará todo nuestro proyecto. Será fundamental el trabajar con todos los sistemas en conjunto para probar el sincronismo de los módulos.

2.2 TARJETA MULTIPLEXORA 9U

La Tarjeta Multiplexora 9U surge ante la necesidad de leer un amplio número de canales y como éxito de la Optical Multiplexer Board comentada en el capítulo 3.

Esta solución podría implicar la reducción del número de RODs finales y en consecuencia una disminución considerable en el coste final del proyecto. El siguiente diagrama de bloques muestra cómo sería esta tarjeta.

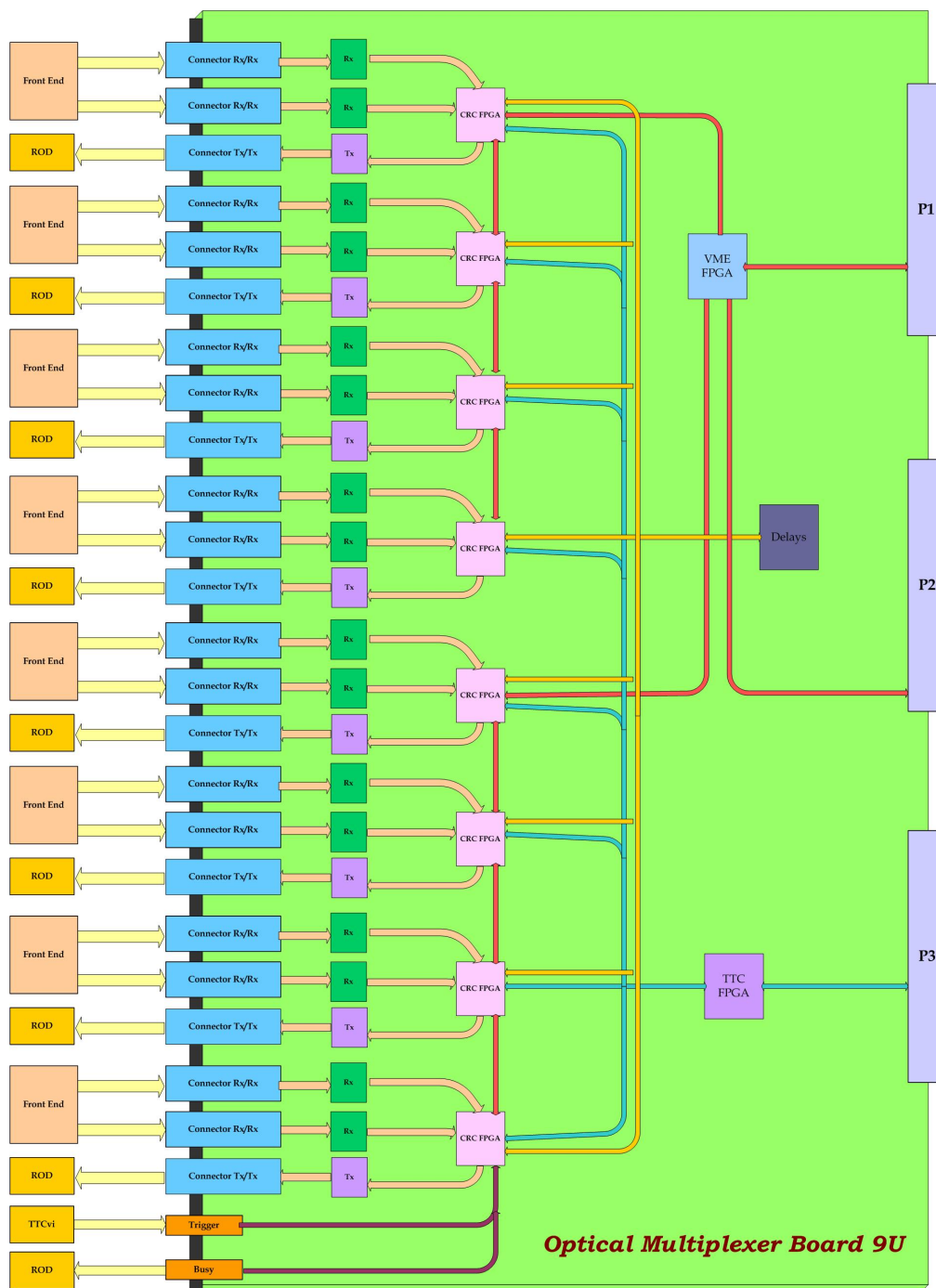


FIGURA 5.2 Diagrama de bloques de la Optical Multiplexer Board 9U.

Las diferencias y mejoras que se pretenden introducir y que ya se están desarrollando son las siguientes:

- Análisis de 8 canales de datos, 16 entradas y 8 salidas. Permitiendo una compatibilidad tarjeta a tarjeta con la Motherboard del ROD (8 canales de entrada por módulo).
- Incorporación del TTC (Timing and Trigger Controller) para una sincronización correcta con ATLAS.
- Desarrollo de nuevos programas para mejorar la eficiencia en el tratamiento de los datos y su envío a la Motherboard del ROD.
- Estudio de nuevas alternativas para generar delays internos que permitan un test más real de las Motherboards del ROD.
- Implementación de la programación de las FPGAs desde VME, permitiéndonos abordar cualquier modificación o problema en las mismas de forma remota.

Está prevista su finalización para Diciembre de este año y su testeo y posterior inclusión en el Combined Test Beam del año que viene.