

TESIS DOCTORAL

2-10-1998



UNIVERSITAT DE VALÈNCIA
REGISTRE GENERAL
ENTRADA
18 JUN. 1998
N.º 57.817
HORA
OFICINA AUXILIAR I

UNIVERSITAT DE VALÈNCIA

Departamento de Informática y Electrónica

Físicas

339

T.D.

TESIS DOCTORAL

PROCESADO Y FUSIÓN JERARQUICA DE DATOS EN REDES DE SENSORES DISTRIBUIDOS. APLICACIÓN AL EXPERIMENTO ATLAS/LHC DEL CERN

presentada por

D. Vicente González Millán

dirigida por

Dr. Enrique Sanchis Peris

UNIVERSITAT DE VALÈNCIA
Biblioteca



80001792271

Valencia, 1998

UMI Number: U607746

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI U607746

Published by ProQuest LLC 2014. Copyright in the Dissertation held by the Author.
Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against
unauthorized copying under Title 17, United States Code.



ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

TESIS DOCTORAL

2-10-1998



UNIVERSITAT DE VALÈNCIA
REGISTRE GENERAL
ENTRADA
18 JUN. 1998
N.º 57.817
HORA
OFICINA AUXILIAR I

UNIVERSITAT DE VALÈNCIA

Departamento de Informática y Electrónica

Físicas

339

T.D

TESIS DOCTORAL

PROCESADO Y FUSIÓN JERARQUICA DE DATOS EN REDES DE SENSORES DISTRIBUIDOS. APLICACIÓN AL EXPERIMENTO ATLAS/LHC DEL CERN

presentada por

D. Vicente González Millán

dirigida por

Dr. Enrique Sanchis Peris

Valencia, 1998

UNIVERSITAT DE VALÈNCIA

FACULTAD DE CIÈNCIES FÍSQUES

Reunit el Tribunal que subscriu, en el dia de la data,
acorda d'atorgar, per unanimitat, a aquesta Tesi Doctoral
d'En/Na/N' VICENTE GONZÁLEZ MILLÁN
la qualificació d' SOBRESALIENTE "CUM LAUDE"

València a 2 d' OCTUBRE de 1998.

El Secretari,

El President,



A handwritten signature in blue ink, appearing to be 'Juan J.', written over a horizontal line.



UNIVERSITAT DE VALÈNCIA
Departament d'Informàtica i Electrònica
Facultat de Física
Avda. Dr. Moliner, 50.
46100 Burjassot, València. Spain

Enrique Sanchis Peris, Doctor en Física y Profesor Titular de Tecnología Electrónica del Departamento de Informática y Electrónica de la Universitat de València,

HAGO CONSTAR:

Que el Ingeniero de Telecomunicación D. Vicente González Millán ha realizado bajo mi dirección, en el Departamento de Informática y Electrónica de la Universitat de València, el trabajo de investigación titulado **“Procesado y Fusión Jerárquica de Datos en Redes de Sensores Distribuidos. Aplicación al Experimento ATLAS-LHC del CERN”**, que se presenta para optar al grado de Doctor por la Universitat de València.

Que D. Vicente González Millán queda facultado para presentar esta Memoria de Tesis Doctoral, para que pueda ser oportunamente defendida y juzgada públicamente según establecen las disposiciones vigentes.

Y para que así conste, firmo el presente documento en Valencia a 18 de Junio de mil novecientos noventa y ocho.

Prof. Dr. Enrique Sanchis Peris

FISIQA

UNIVERSITAT DE VALÈNCIA
BIBLIOTECA CIÈNCIES

Nº Registre 12413

DATA 26-X-98

SIGNATURA T. D. 339

Nº LIBRE: j1998506X

A Maite

AGRADECIMIENTOS

Ante todo quisiera agradecer al Dr. Enrique Sanchis, director de esta su primera tesis, todo el esfuerzo que ha realizado para la concepción y buen desarrollo de la misma. Bien sabemos los dos los días y noches, de los veranos de estos últimos tres años en el CERN, que ha costado. Gracias también por todo el apoyo recibido no sólo en esta tesis sino en todo cuanto con ella estaba relacionado. También gracias por todos los consejos recibidos que han hecho posible que ahora esté donde estoy.

Al Dr. Antonio Ferrer por su inestimable apoyo, ayuda y colaboración en todo lo referente a la integración del prototipo en el proyecto ATLAS/LHC, en la disponibilidad de medios en el CERN y su *savoir faire* dentro del mismo.

A Alfonso Ríos, JuanBa Romance, Tomàs Pons por toda la ayuda prestada en el diseño del software para el prototipo, la integración del mismo en el sistema de adquisición de CERN y las pruebas. Gracias por todas las noches que pasamos en el labo desesperados porque las cosas no iban. Especialmente por las pruebas del último TestBeam donde se sacrificaron muchas horas de sueño para llegar a tiempo.

Gracias al Dr. Ángel Sebastiá, al Dr. Francisco Mora y a Ricardo Colom por toda la ayuda prestada con la integración del prototipo en el anillo SCI y por los buenos momentos pasados durante las estancias en el CERN.

ÍNDICE

Prólogo	vi
<u>CAPÍTULO 1 REDES DE SENSORES: ORGANIZACIÓN Y PROCESADO</u>	
1. SENSORES	1-1
1.1 Atributos de los sensores	1-2
2. EVOLUCIÓN DE LOS SISTEMAS SENSORES	1-4
3. PROCESADO DE SENSORES	1-6
3.1 Adquisición	1-7
3.2 Procesado	1-7
3.3 Integración	1-8
3.3.1 Necesidad	1-8
3.3.2 Mecanismos de integración de sensores	1-9
3.4 Análisis	1-10
4. ORGANIZACIÓN DE SISTEMAS DE PROCESADO DE SENSORES	1-11
4.1 Colección de sensores	1-11
4.2 Sistemas jerárquicos	1-11
4.3 Procesado de sensores en árbol	1-13
4.4 Integración multisensor	1-14
4.5 Procesado distribuido de sensores	1-15
4.5.1 Descomposición de tareas	1-15
4.5.2 Procesado local	1-16
4.5.3 Integración	1-17
4.5.4 Análisis	1-18
4.6 Redes de sensores distribuidos	1-18
5. ARQUITECTURAS PARA EL PROCESADO DE SENSORES	1-19
5.1 Topología de la red	1-19
5.1.1 Organización en comité	1-19
5.1.2 Organización jerárquica	1-20
5.1.3 Organización en árbol plano	1-20
5.2 Modelo de sincronización	1-23
5.2.1 Modelo de reloj	1-23
5.2.2 Comportamiento del reloj y aspectos de sincronización	1-23
5.2.3 Resincronización de reloj	1-23

5.3 Rutado	1-24
5.3.1 Restricciones en la comunicación	1-25
5.3.2 Restricciones debidas al retardo	1-25
5.3.3 Restricciones debidas a la fiabilidad	1-25
6. OBJETO DE LA TESIS	1-26
7. BIBLIOGRAFÍA	1-27

CAPÍTULO 2 EVALUACIÓN DE ARQUITECTURAS PARA EL PROCESADO DE SENSORES

1. INTRODUCCIÓN	2-1
2. CLASIFICACIÓN DE LAS ARQUITECTURAS	2-1
3. ARQUITECTURAS MIMD	2-5
3.1 Clasificación	2-5
3.1.1 Sistemas multiprocesador	2-5
3.1.2 Sistemas multicomputador	2-7
3.2 Parámetros de excelencia de los sistemas MIMD	2-8
3.2.1 Escalabilidad	2-8
3.2.2 Incremento de velocidad y eficiencia de un sistema multiprocesador	2-9
4. REDES DE INTERCONEXIÓN EN SISTEMAS MIMD	2-12
4.1 Sistemas multiprocesador	2-13
4.2 Sistemas multicomputador	2-16
4.3 Tecnologías de redes de interconexión	2-18
4.3.1 Scalable Coherent Interface (SCI)	2-18
4.3.1.1 Características	2-18
4.3.1.2 Protocolo de transmisión	2-19
4.3.2 Topologías interconexión	2-20
4.3.2.1 Integrados para el estándar SCI	2-21
4.3.3 Asynchronous Transfer Mode (ATM)	2-22
4.3.3.1 Aspectos y conceptos básicos de ATM	2-23
4.3.3.2 El protocolo ATM	2-24
4.3.3.3 Arquitectura genérica de un conmutador	2-25
4.3.4 Fibre Channel (FC)	2-27
4.3.4.1 Topología Fibre Channel	2-27
4.3.4.2 Estructura del protocolo FC	2-28
5. ARQUITECTURAS PARA LA INTEGRACIÓN DE SENSORES	2-31
5.1 Problemas con elevada tasa de datos	2-31
5.1.1 Calidad requerida de un sistema	2-33
5.1.2 Parametrización de las arquitecturas paralela y jerárquica	2-34
5.1.3 Evaluación de la calidad requerida	2-37
5.2 Introducción de elementos de preprocesado	2-40
5.2.1 Regiones de Interés	2-40
5.2.2 Agrupamiento de datos	2-46
5.3 Desaleatorizadores	2-49
6. BIBLIOGRAFÍA	2-51

CAPÍTULO 3 APLICACIÓN DE TÉCNICAS DE REGIONES DE INTERÉS Y AGRUPAMIENTO DE DATOS AL EXPERIMENTO ATLAS/LHC DEL CERN

1. INTRODUCCIÓN	3-1
2. EL EXPERIMENTO ATLAS/LHC	3-1
2.1 Descripción	3-3
2.2 Requerimientos de la aplicación. Posibles soluciones	3-5
2.2.1 Solución paralela	3-6
2.2.2 Solución jerárquica	3-7
2.2.2.1 El primer nivel	3-8
2.2.2.2 El segundo nivel	3-10
2.2.2.3 El tercer nivel	3-11
2.2.2.4 Evaluación de la arquitectura jerárquica	3-13
3. INCORPORACIÓN DE REGIONES DE INTERÉS Y AGRUPAMIENTO DE DATOS	3-13
3.1 Regiones de interés (RoI)	3-14
3.2 Agrupamiento de datos (ROD)	3-18
3.3 Opciones tecnológicas para la implementación de los niveles	3-21
3.3.1 El primer nivel	3-21
3.3.2 El segundo nivel	3-22
3.3.3 El tercer nivel	3-25
4. BIBLIOGRAFÍA	3-26

CAPÍTULO 4 VERIFICACIÓN EXPERIMENTAL

1. INTRODUCCIÓN	4-1
2. DISEÑO DE UN MÓDULO DE AGRUPAMIENTO DE DATOS	4-1
2.1 El detector TileCal	4-1
2.2 Especificaciones generales del módulo	4-4
2.3 Especificaciones particulares para TileCal	4-5
2.4 Diagrama de bloques del prototipo	4-5
2.4.1 La tarjeta DBV44	4-7
2.4.2 La tarjeta interface SCI	4-8
2.4.3 El módulo interface S-LINK	4-9
2.4.3.1 Solución adoptada. Diseños alternativos	4-12
2.4.3.2 Cálculos	4-15
2.4.3.3 DISEÑO DE LA PLACA	4-17
3. DESCRIPCIÓN DE LAS PRUEBAS Y RESULTADOS	4-20
3.1 Determinación del número de DSPs	4-20
3.2 Integración del prototipo	4-25
3.2.1 Sistema StandAlone	4-25
3.2.2 Sistema TestBeam	4-29
3.2.2.1 Descripción del TestBeam para TileCal	4-30
3.2.2.2 Inclusión del prototipo	4-34
3.2.3 Resultados	4-37
4. BIBLIOGRAFÍA	4-38

CAPÍTULO 5 CONCLUSIONES Y TRABAJO FUTURO

1. CONCLUSIONES 5-1

2. TRABAJO FUTURO 5-2

ANEXO I ESQUEMAS DE LA TARJETA INTERFACE S-LINK I-1

PRÓLOGO

El objeto de la tesis es el estudio de las arquitecturas para el procesado y fusión jerárquica en redes de sensores distribuidos y la introducción de mejoras en las mismas.

Las redes de sensores distribuidos son de utilidad en sistemas de detección y sensado remoto, tales como sistemas de vigilancia aérea, control de procesos, etc. La característica principal de este tipo de sistemas es que, por lo general, los sensores utilizados no son homogéneos, es decir, no miden todos la misma magnitud. Esto es debido a dos factores:

- puede ser que, para la identificación del fenómeno físico estudiado, se necesite evaluar más de una de sus características.
- la medición de variables diferentes puede mejorar el rendimiento del sistema y aumentar su fiabilidad.

Por ello, a menudo, los sistemas se componen de un cierto número limitado de tipos de sensores diferentes, pero todos ellos replicados en la cantidad adecuada para cubrir todo el espacio de observación.

El problema, desde el punto de vista de la adquisición de datos, es cómo integrar de forma coherente la información dispar ofrecida por los sensores. A este respecto, existen diversas soluciones, pero cuando se trata de problemas en los que, debido al gran número de sensores y a la elevada frecuencia con la que se toman los datos, el volumen de información a tratar es grande, se suelen emplear arquitecturas jerárquicas para la integración de los mismos.

Desde el punto de vista tecnológico, los sistemas jerárquicos conducen a soluciones multiprocesador del tipo MIMD según la clasificación de Flynn. Este tipo de arquitecturas presenta un flujo de datos múltiple (procedentes de los diferentes sensores) y un flujo múltiple de instrucciones (aquellas necesarias para procesar la información de cada sensor).

Además del elemento de procesado empleado en la arquitectura, una parte fundamental de la misma, que afecta a su rendimiento, es la red de interconexión que se emplee. Para el tipo de sistemas con el que se trata, las redes deben poseer grandes prestaciones en cuanto a velocidad de transmisión, ancho de banda y latencia. Actualmente, existen tres redes que destacan por esos parámetros y cuya aplicación es posible en los sistemas MIMD: el Scalable Coherent Interface (SCI), el Asynchronous Transfer Mode (ATM), y el FibreChannel (FC).

Las arquitecturas jerárquicas, como ya se ha dicho, se emplean en este tipo de sistemas. Sin embargo, no se ha encontrado en la literatura una justificación sobre el

porqué es así y tampoco acerca de cuántos niveles debe poseer la misma, sino que se actúa por la experiencia acumulada. Por ello, se presenta en la tesis un estudio de la comparación de este tipo de arquitecturas con la que podría ser una solución alternativa válida: una arquitectura totalmente paralela.

La comparación se ha limitado a evaluar los elementos de procesado empleados y no ha tenido en cuenta los medios por los que éstos se interconectan. Para la realización de esta comparación se ha definido en primer lugar un factor de calidad requerida de la unidad de procesado definido como el producto de su ancho de banda por su capacidad de procesado, con unidades de MBytes/s x MIPS (millones de instrucciones por segundo). Este factor, definido de esta manera, asegura que las unidades con mayor factor de calidad requerida presentarán mayor ancho de banda y/o mayor capacidad de procesado, lo que, en el fondo, implica un mayor coste económico.

Se ha estudiado después el factor de calidad requerida en la asociación serie y paralelo de procesadores en función del factor de calidad requerida de cada uno de los que componen este tipo de sistemas.

Con la ayuda de esta función se ha evaluado el sistema paralelo y el sistema jerárquico y se han comparado las expresiones obtenidas. El resultado es que bajo ciertas condiciones, el sistema jerárquico posee un factor de calidad requerida inferior al paralelo. Esta condición es función del número de niveles del sistema jerárquico, del factor de calidad requerida de cada unidad de procesado tanto en el sistema paralelo como en el jerárquico, del número de unidades que se empleen en ambos casos y de la probabilidad de detección del fenómeno estudiado.

A la vista de los resultados obtenidos, se presenta una mejora sobre el sistema jerárquico consistente en la introducción de una técnica de localidad de datos utilizada en el procesado de imágenes, conocida como región de interés, y otra de agrupamiento selectivo de datos.

La región de interés define físicamente un subespacio dentro del espacio de observación donde, según las primeras estimaciones, existe la posibilidad de localizar el dato buscado. Esto se traduce en que sólo aquellos sensores correspondientes a la región de interés deberán ser procesados, con lo que la capacidad de cálculo requerida disminuye. Sin embargo, la introducción de regiones de interés implica añadir un nuevo elemento al sistema de adquisición de datos: el constructor de región de interés. Este elemento será el encargado de definir el tamaño de la región en función de los datos procedentes de los sensores.

Se ha comparado el sistema con y sin regiones de interés para observar en qué situación el factor de calidad requerida del primero es inferior a la del segundo. El resultado depende, obviamente, de la reducción en los factores de calidad de las unidades de procesado de los niveles de la arquitectura jerárquica y del aumento del factor de calidad requerida debida a la introducción del constructor.

En el estudio anterior, se supone que los datos a procesar en cada unidad se encuentran en la memoria del elemento. Sin embargo, esto, en la mayoría de los casos no es así. Por ello, la segunda mejora supone que se puede lograr ese caso ideal o, por lo menos, acercarse a él, realizando un agrupamiento selectivo de datos. Este agrupamiento se puede dar de dos formas: dinámica o estáticamente.

En el caso del agrupamiento dinámico, los datos se concentran de forma que una o varias regiones de interés completas vayan al mismo procesador. De esta manera se puede llegar a la solución óptima pero a costa de aumentar el tiempo de procesado ya que antes de comenzar hay que rutar los datos siguiendo las indicaciones del constructor.

El caso del agrupamiento estático sólo es posible cuando existe uno cierto sesgo en el fenómeno observado que le hace más propenso a aparecer en cierta región del espacio antes que en otra. Si esto es así, se pueden agrupar los sensores de forma que

constituyan subespacios del tamaño de una región de interés, de manera que se haga máxima la probabilidad de que ésta contenga el suceso. En este caso, la solución no es óptima pero a cambio tenemos un sistema más rápido pues no hace falta la información del constructor para el agrupamiento.

La comparación del sistema con regiones de interés y agrupamiento y del mismo sin agrupamiento demuestra que el factor de calidad requerida aumenta, debido a que no estamos en el caso ideal, y que este aumento depende de la probabilidad de que las unidades de procesado requieran datos no incluidos en sus memorias.

Para la verificación experimental de estos resultados se podría tomar cualquier aplicación de fusión de datos procedentes de sensores, como un sistema distribuido de control, un sistema de sensado remoto, etc. En nuestro caso, se ha escogido una aplicación donde tradicionalmente se ha empleado la arquitectura jerárquica para la adquisición de los datos, los experimentos de física de altas energías. En ellos, se emplean detectores de partículas constituidos hasta por centenares de millones de sensores cuya información debe ser filtrada de forma adecuada puesto que la probabilidad del suceso buscado es muy pequeña. En nuestro caso, se ha escogido el experimento ATLAS/LHC en preparación en el Laboratorio Europeo de Física de Partículas (CERN) en Ginebra. Se presenta la evaluación del sistema de adquisición antes y después de la introducción de regiones de interés y agrupamiento de datos, comprobándose cómo se reduce el factor de calidad requerida del sistema.

Se ha diseñado un módulo de agrupamiento y preprocesado de datos para uno de los subdetectores de este experimento con el objeto de validar experimentalmente las conclusiones acerca de la introducción de este mecanismo en el sistema de adquisición. Dicho módulo está basado en procesadores digitales de señal adaptados a las especificaciones concretas del subdetector.

Se describen también las pruebas realizadas con ese módulo en el subdetector y los resultados obtenidos que verifican los teóricos expuestos anteriormente y demuestran la utilidad del módulo en el sistema de adquisición. A la vista de los resultados, 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 un análisis que permita la modificación de la función de coste para que incluya los aspectos de coste de la comunicación ahora no contemplados.

CAPÍTULO

1

REDES DE SENSORES: ORGANIZACIÓN Y PROCESADO

1. SENSORES	1
1.1 Atributos de los sensores	2
2. EVOLUCIÓN DE LOS SISTEMAS SENSORES	4
3. PROCESADO DE SENSORES	6
3.1 Adquisición	7
3.2 Procesado	7
3.3 Integración	8
3.3.1 Necesidad	8
3.3.2 Mecanismos de integración de sensores	9
3.4 Análisis	10
4. ORGANIZACIÓN DE SISTEMAS DE PROCESADO DE SENSORES	11
4.1 Colección de sensores	11
4.2 Sistemas jerárquicos	11
4.3 Procesado de sensores en árbol	13
4.4 Integración multisensor	14
4.5 Procesado distribuido de sensores	15
4.5.1 Descomposición de tareas	15
4.5.2 Procesado local	16
4.5.3 Integración	17
4.5.4 Análisis	18
4.6 Redes de sensores distribuidos	18
5. ARQUITECTURAS PARA EL PROCESADO DE SENSORES	19
5.1 Topología de la red	19
5.1.1 Organización en comité	19
5.1.2 Organización jerárquica	20
5.1.3 Organización en árbol plano	20

5.2 Modelo de sincronización	23
5.2.1 Modelo de reloj	23
5.2.2 Comportamiento del reloj y aspectos de sincronización	23
5.2.3 Resincronización de reloj	23
5.3 Rutado	24
5.3.1 Restricciones en la comunicación	25
5.3.2 Restricciones debidas al retardo	25
5.3.3 Restricciones debidas a la fiabilidad	25
6. OBJETO DE LA TESIS	26
7. BIBLIOGRAFÍA	27

1. SENSORES

En el estudio científico de los fenómenos naturales que nos rodean, una de las primeras tareas a realizar consiste en el análisis detallado de las variables físicas del fenómeno para obtener la máxima información acerca del mismo. Para realizar este análisis se emplean sensores capaces de medir la variable física y convertir esa medida en información útil para el estudio.

Un sensor es un dispositivo capaz de responder ante una variable física de forma predecible. Los sensores pueden ser mecánicos, eléctricos, electromecánicos, electrónicos, magnéticos, electromagnéticos u ópticos, por nombrar algunos. El que la variación producida en el sensor sea predecible, es decir, siga una determinada ley conocida, es fundamental para después estimar el valor de la variable física en función del valor obtenido por el sensor. A esta ley se le denomina función de transferencia del sensor, y nos asegura que el valor a la salida obtenida de un sensor es análogo al de la variable física sensada [1].

Los sensores pueden ser de forma muy variada incluso aquellos que miden la misma variable. Sin embargo, cualquier sensor puede ser estudiado bajo dos aspectos: físico y funcional.

El aspecto físico se refiere a cómo está hecho el sensor o a cuál es su forma. Los términos sensor físico o concreto se refieren a los dispositivos que sensan la variable física de interés, como por ejemplo un barómetro, un radar, un termómetro, etc.

El aspecto funcional se refiere a lo que se supone que debe hacer el sensor y/o a cuál es su abstracción. Los términos sensor abstracto o lógico se usan para referirse a una abstracción de la lectura tomada por un sensor concreto. Existen diferentes abstracciones posibles. Podemos denotar la lectura de un sensor como un simple número o como un intervalo en la recta real \mathfrak{R} . Por ejemplo, la lectura de un barómetro situado en un medio E para un instante t puede expresarse de las siguientes maneras:

1. $P[E, t] = v$
2. $P[E, t] = v \pm \Delta v$
3. $P[E, t] \in v_{inf} \dots v_{sup}$

La primera abstracción expresa que v es el valor leído. La segunda indica la existencia de una tolerancia Δv . La tercera asegura que el valor verdadero se halla en un intervalo cuyos límites inferior y superior son v_{inf} y v_{sup} . Hay que hacer notar que el valor verdadero no tiene por qué situarse en el centro del intervalo, en cuyo caso la segunda y la tercera abstracción serían idénticas.

En la mayoría de los casos los sensores llevan siempre asociados un elemento transductor que convierte las variaciones de las características del mismo en una señal eléctrica útil. El hecho de que la conversión sea a una señal eléctrica se debe a la facilidad con la que se trabaja en ese campo a la hora de hacer operaciones con la información.

Puesto que la propia existencia de un fenómeno implica una variación en algunos o todos los parámetros a él asociados, la señal eléctrica que se obtiene presentará una cierta variación con el tiempo, directamente relacionada con la variación de la magnitud medida.

Así pues, en el estudio del fenómeno se ha pasado de un espacio $(n+1)$ dimensional de magnitudes físicas (n magnitudes y el tiempo) a n espacios bidimensionales de magnitudes eléctricas (amplitudes y tiempos), cada una correspondiente a una de las magnitudes física a medir.

La ventaja obtenida con esta transformación es que en el campo de las señales eléctricas disponemos de toda una serie de herramientas y tecnologías que permiten

el análisis y tratamiento de las mismas, cosa que muchas veces no es posible directamente sobre las magnitudes físicas del fenómeno.

Para tratar la información obtenida por los sensores es necesario un sistema de procesado de los mismos que sea capaz de operar adecuadamente con los datos de cada sensor, los interprete y obtenga el resultado buscado. Este sistema será más o menos complejo en función del número de sensores que se empleen, tanto iguales como diferentes, cosa que a su vez depende del fenómeno que se esté estudiando.

Antes de entrar en el estudio de los sistemas de procesado de sensores, veamos algunas características generales referidas a los sensores.

1.1 ATRIBUTOS DE LOS SENSORES

El diseño de un sistema de procesado de sensores depende de la elección de una serie de sensores cuyos atributos se ajusten a los requerimientos de la aplicación. A continuación se presenta una relación de atributos de los sensores:

- **Accesibilidad**

La accesibilidad se refiere a la porción del medio a la que el sensor tiene acceso. El acceso parcial se produce cuando el medio a sensar es demasiado grande para ser manejado por un solo sensor. Por ejemplo, aunque es teóricamente posible tener un sistema de vigilancia aérea para todo el territorio español, en la práctica hay varios sistemas de vigilancia con jurisdicciones solapadas. En este caso cada sensor tiene un acceso parcial del espacio aéreo. También si lo que queremos es conocer la distribución de temperatura en una sala, emplearemos varios sensores distribuidos espacialmente, de forma que cada uno toma la temperatura de una zona cercana al mismo.

- **Dimensión de la medida**

Se dice que un sensor es lineal, escalar o uno-dimensional si puede ser modelado por un valor escalar. Por ejemplo, un sensor térmico mide la temperatura que es un valor escalar. Se dice que un sensor es multidimensional si se puede modelar por un vector. Por ejemplo, un sensor de posición es un sensor tridimensional puesto que obtiene un vector de coordenadas espaciales (x, y, z) .

- **Rango de operación**

Asociado con cada sensor existe un rango de operación. Algunos sensores tienen un rango de operación preferido (corta, media o larga distancia); algunos son adecuados sólo para el día o para la noche; algunos pueden funcionar durante todo el día. Un termómetro clínico está diseñado para operar cerca de 37°C que es la temperatura normal para la mayoría de las personas. En contraste, los termómetros para monitorizar la temperatura interna de la vasija de un reactor nuclear tienen un rango de operación centrado a una temperatura mucho mayor.

- **Tipo de dato**

El sensor puede proporcionar datos continuos o discretos. Los datos pueden ser binarios, numéricos o alfanuméricos. Los datos pueden ser enviados a otro sensor y/o a una unidad de procesado. El sensor puede o no tener un buffer asociado a él.

Además, el sensor puede tener que transmitir los datos a través de una red de comunicación a un ordenador central para el procesado de los mismos. Los datos pueden organizarse en mensajes o en paquetes. La cantidad de datos transmitidos a través de la red determina el ancho de banda requerido.

Si el volumen de datos a transmitir es elevado, la solución es emplear técnicas de compresión de datos o incluir elementos de procesado local antes de la transmisión, como se verá más tarde.

- **Sensibilidad**

En la construcción de un sensor se tienen en cuenta ciertas especificaciones tales como la precisión y la sensibilidad. La sensibilidad se puede definir como la variación más pequeña de la señal sensada que es posible detectar mediante un sensor determinado, mientras que la precisión es el error que comete el sensor en la lectura de la variable física. Este error se expresa como una tolerancia asociada a cada medida. Ambos factores pueden degradarse a lo largo del tiempo o en presencia de condiciones adversas (temperatura, humedad, suciedad del ambiente, etc.).

- **Localización**

La situación del sensor puede o no ser la misma que la localización del sistema de procesado. Si no es la misma, el sensor se llama *remoto*. El sensado remoto se emplea, por ejemplo, en la obtención de imágenes por satélite, vigilancia aérea, instrumentación y control. Los datos recogidos remotamente junto con un identificador de la localización del sensor que los ha obtenidos deben ser transmitidos al centro de procesado para su elaboración.

- **Inteligencia**

Existen varias nociones de inteligencia. La inteligencia es la habilidad para realizar abstracción y conceptualización. En el contexto de sistemas inteligentes autónomos, la inteligencia se refiere a la habilidad para recoger la información, hacer decisiones e implementar acciones que conduzcan hacia el logro de un determinado objetivo [1].

- **Sensores activos y pasivos**

Un sensor activo es aquel que opera emitiendo energía de algún modo y después detectando e interpretando la energía devuelta. Ejemplos de sensores activos son el sonar, el radar, etc.

Se dice que un sensor es pasivo si simplemente interpreta los cambios de alguna propiedad del medio que lo rodea. No emite energía como parte del proceso de sensado. Ejemplos de sensores pasivos incluyen los termopares, barómetros y fotocélulas, etc.

Un conjunto de sensores puede ser totalmente activo, pasivo o mixto. Si existe más de un sensor activo, la sincronización de los sensores activos es crucial para el rendimiento del sistema.

- **Sensores de contacto y de no contacto**

Cuando la salida de un sensor se produce como consecuencia del contacto físico del mismo con algo, se dice que el sensor es de contacto. Como ejemplos tenemos los sensores de desplazamiento, colisión, presión, etc.

Si el sensor opera sin contacto físico con el objeto sensado, se dice que es de no contacto. Ejemplos típicos son las células fotoeléctricas, sensores de proximidad capacitivos, sensores de luz, etc.

Ciertos sensores pueden usarse bien como de contacto o de no contacto en función de la aplicación. Los termopares se pueden considerar como sensores de contacto para a mayoría de las aplicaciones porque requieren contacto directo con el medio cuya temperatura está siendo medida. Sin embargo, los termopares pueden usarse también como sensores de no contacto en algunas aplicaciones de alta temperatura cuando se exponen a la energía radiante del objeto sensado.

- **Tecnología**

Las tecnologías actuales de sensores incluyen la optoelectrónica, radar, ondas milimétricas, detección de radiación de neutrones, tomografía de rayos X, etc.

2. EVOLUCIÓN DE LOS SISTEMAS SENSORES

La evolución de los sistemas sensores se puede dividir en varias épocas cada una caracterizada por la mejora de las prestaciones de los sensores de la anterior y por el hecho de que vienen a solucionar problemas que insolubles con los sistemas de etapas anteriores. Podemos describir dicha evolución en cinco etapas cada una representada por un tipo diferente de sensor, aunque una nueva etapa no implica la desaparición de los sensores utilizados en la anterior.

- **Sistemas de un solo sensor**

Un ejemplo de sistema de un solo sensor lo tenemos en un radar. Este equipo envía una señal de radio de frecuencia y dirección determinadas, y recibe la señal reflejada en los objetos que se encuentran en el camino del haz enviado. De la diferencia de tiempos entre el envío de la señal y la recepción de la onda reflejada, se puede estimar la distancia al objeto. Otro ejemplo de sistema de un solo sensor es el sonar. Este equipo puede usarse para determinar la profundidad del suelo marino.

Antes de advenimiento de la microelectrónica, los sistemas de sensores tendían a ser grandes y/o caros. Así pues, este tipo de sistemas usaba un único sensor para ahorrar espacio y/o dinero. Las ventajas de los sistemas de un solo sensor son la facilidad de construcción y el análisis de los datos.

Sin embargo, también tienen sus desventajas. La primera es que sus aplicaciones y usos son limitados. Por ejemplo, un robot móvil autónomo necesita varios sensores (táctiles, cámaras CCD, etc.), y, por lo tanto, no puede construirse como un sistema de un solo sensor. En segundo lugar, los sistemas de un solo sensor no son robustos ni soportan fácilmente los fallos para poder ser utilizados en aplicaciones críticas. La tercera desventaja es que un sistema de un solo sensor no puede garantizar que la información que está dando sea exacta todo el tiempo debido a su rango de operación y a la relación señal/ruido. Asociado con cada sensor existe un límite que define su rango de operación útil. Por ejemplo, el espectro electromagnético incluye rayos gamma, rayos X, ultravioletas, luz visible, infrarrojos, microondas y radio; no existe un sensor capaz de cubrir todo el espectro. Por otra parte, cualquier señal está expuesta a la influencia del ruido; si este es elevado puede perturbar la medida, y sin otra referencia, hacer que no pueda conocerse la forma en que ésta es afectada.

- **Sensores replicados**

Asociada con cada sensor, existe una curva en forma de U que muestra la probabilidad de que el dispositivo falle en función del tiempo. Lo más normal es que el sensor falle bien al principio de su operación, bien al final más que en el interin [2]. Así pues, en el diseño de un sistema de un solo sensor se suele usar uno que haya pasado ya su periodo inicial de funcionamiento. Pese a esta táctica, no existe la garantía de que el sensor no falle en algún momento. Si no existen requerimientos de velocidad muy exigentes, el sistema puede y debería validar las lecturas del sensor mediante el empleo de varios sensores. Basándose en sus especificaciones, la salida puede ser ignorada si no es realista. Por ejemplo, si tenemos un sistema de medición de temperatura en un reactor nuclear, interesa conocer en todo momento si existe el peligro de fisión en cadena. Como esta decisión puede tener consecuencias nefastas, es necesario asegurarse de que la medición realizada es buena y que los sensores no están funcionando de forma errónea. Para ello, basta con disponer más de un sensor y comparar las mediciones.

¿Deberíamos usar dos o tres sensores replicados? Si el coste es un problema, dos sensores son más baratos que tres; si no lo es, tres son más seguros que dos, ya que con dos no puede saberse cuál de ellos está dando una lectura incorrecta. La redundancia mejora la fiabilidad al tolerar un número determinado de fallos en los sensores.

Cuando existen más de dos sensores, se debe emplear alguna técnica para obtener el valor correcto de la lectura. Existen varias técnicas para lograr este objetivo: la primera consiste en tomar la lectura que aparece igual (dentro de un cierto margen) en un número mayor de sensores; también se puede realizar una media de todas las lecturas; por último, si los valores de los sensores son continuos, se puede asignar un peso a cada sensor y realizar una media ponderada según esos factores.

- **Sensores diferentes**

El sistema con un solo sensor no era fiable. El uso de sensores replicados mejoraba la tolerancia a fallos, pero, incluso en este caso, se tiene un espacio de sentido limitado, y, a menudo, no puede sentir el fenómeno completo que se está estudiando.

Por ello, el empleo de varios sensores diferentes soluciona el problema de poder sentir magnitudes diferentes de un mismo fenómeno, necesarias para la total interpretación del mismo.

Por ejemplo, un robot móvil autónomo está equipado con sensores diferentes: visión, sonar, tacto, infrarrojos, etc. necesarios para obtener una aprehensión completa de su entorno y poder evolucionar a través de él.

Durrant-Whyte [3], entre otros, discutieron las ventajas de la integración de sensores diferentes. Entre ellas destacaron:

- Incremento de la fiabilidad.
- Mejora de la tolerancia a fallos.
- Mejora en la detección y eliminación del ruido.

Este último punto se explica por el hecho de que el ruido recogido por sensores diferentes tiende a no estar correlacionado, mientras que la señal de interés si lo está. Mas aún, los sensores pueden escogerse de tal manera que unos complementen a los otros.

- **Sensores distribuidos espacialmente**

Algunas aplicaciones requieren que las observaciones de un objeto se tomen simultáneamente desde dos o más puntos. Existen varios grados en los que los sensores pueden distribuirse espacialmente: desde un área limitada, hasta una región o incluso todo un país.

El tipo de sensor usado puede ser cualquiera de los vistos anteriormente, incluso una combinación de los mismos. La peculiaridad de estos sistemas es que ahora la información varía tanto espacial como temporalmente, por lo que se requerirá un sistema de procesamiento más complejo que tenga en cuenta esos dos aspectos.

- **Sensores inteligentes**

Si empleamos un número elevado de sensores, ya sean replicados o diferentes, puede ser que el volumen de información a tratar crezca de manera que el problema tenga difícil solución. Una posibilidad es emplear sensores inteligentes en la medida de la variable física.

Un sensor inteligente es aquel que incluye cierta circuitería lógica capaz de abstraer información con un contenido semántico mayor que el de la señal eléctrica obtenida de la variable física. Por ejemplo, en un sistema para detectar el paso de personas en un recinto puede ofrecer sólo un pulso eléctrico cuando pasa una persona, o tener la lógica necesaria para ofrecer un valor numérico representativo del número de personas que han pasado a través del sistema. En este caso, el sensor se hace inteligente al ofrecer una información más elaborada que la puramente eléctrica.

La abstracción de la información lleva consigo una reducción de la información que repercute en el diseño del sistema de procesado, al aligerar la carga computacional y el ancho de banda necesario del mismo.

3. PROCESADO DE SENSORES

El procesado de los datos procedentes de los sensores es un problema importante y central en los sistemas de sensores. Por su propia naturaleza, se trata de una materia interdisciplinar que requiere conocimientos de campos como la física, la electrónica y la informática. De él depende que la información obtenida por los sensores sea adecuadamente elaborada para obtener una decisión sobre aquello que se está sensando.

El procesado de sensores puede verse como un procedimiento consistente en cuatro actividades: adquisición, procesado, integración y análisis (figura 1.1).



FIGURA 1.1 Sistema de procesado de sensores típico.

No todos los sistemas de procesado de sensores incluyen las cuatro actividades, algunos puede que omitan algunas o las traten muy superficialmente.

Veamos con un ejemplo didáctico en qué consisten cada una de las actividades del sistema de procesado de sensores. Imaginemos un sistema que obtiene el volumen de un recipiente en el que existe un gas de manera indirecta. La forma de realizar esto es a través de la medición de la temperatura y la presión del gas. Existiría una etapa de adquisición en la que se tomarían los valores de presión y temperatura con los sensores apropiados. La etapa de procesado podría consistir en la corrección de los valores obtenidos de acuerdo con la linealidad del sensor, el ruido eléctrico del sistema u otros factores. Tras esta corrección los valores de presión y temperatura se operarían de la manera adecuada siguiendo las leyes físicas para obtener un resultado para el volumen del recipiente. Por último, esta información podría

emplearse en una etapa de análisis para comprobar si el recipiente tiene el volumen adecuado.

Para sistemas de un solo sensor no son necesarias ni la integración ni el procesado. En sistemas con sensores replicados, el procesado podría ser mínimo, pero la integración es crucial. Si el sistema se basa en sensores diferentes, el procesado es necesario para hacer que las lecturas de los diferentes sensores sean compatibles y la integración lo es para obtener una información global del entorno.

Sin embargo, la mayoría de los sistemas de procesado de sensores realizarán las cuatro actividades. La variable física se sensorará en la actividad de adquisición. Los datos obtenidos se procesarán de forma adecuada (por ejemplo un escalado o un formato) para poder ser llevados a la actividad de integración. La salida de esta actividad se lleva a la de análisis de donde se obtendrá una decisión. El mecanismo de obtención de la decisión puede ser determinista, estocástico o empírico.

Veamos a continuación con un poco más detalle cada una de las cuatro actividades del sistema de procesado de sensores.

3.1 ADQUISICIÓN

La primera actividad del sistema de procesado de sensores es adquirir los datos del sistema físico o del entorno que se analiza. Los datos adquiridos pueden ser un conjunto de medidas de la variable física tomadas en un determinado punto a lo largo del tiempo o un conjunto de medidas de la variable física tomadas en diferentes puntos en un mismo instante. En general, los datos son función de la variable física a lo largo del tiempo y del espacio.

En la adquisición, los datos pueden provenir de uno o varios sensores a través de sus correspondientes transductores. Si estos sensores se usan para decidir sobre la presencia o no de un determinado objeto o suceso, entonces reciben el nombre de detectores.

Todas la información adquirida por los sensores consiste en dos partes: la parte útil o *señal* y aquella que no es útil o *ruido*. La mayoría de las señales van acompañadas de ruido blanco que se supone no correlacionado y gaussiano.

Los datos recogidos pueden ser continuos o discretos, dependiendo de los posibles valores que puedan presentar. En general, la salida del transductor es una señal analógica con valores de amplitud continuos. Sin embargo, la práctica común es muestrear y cuantizar esa información analógica para transformarla en información binaria más fácilmente tratable. El muestreo viene determinado por su periodo, mientras que la cuantización lo es por sus niveles. Estos dos parámetros determinan la cantidad de información que pasa hacia la actividad de procesado, y que viene expresada como una tasa binaria.

3.2 PROCESADO

La segunda actividad en el sistema de procesado es conocida también como procesado local. Si a esta actividad le sigue alguna forma de procesado global, también recibe el nombre de preprocesado.

Existen varias razones por las que podemos querer procesar los datos que obtenemos de los sensores. Quizá las dos más importantes son:

- los datos contienen una componente de ruido tanto blanco como puede que espúreo inducido por otros sistemas cercanos. Para mejorar la relación señal-ruido podemos usar un filtro que suprima o minimice el ruido.
- algunas aplicaciones necesitan un procesado local para organizar los distintos grupos de datos (a veces sin estructura) de una determinada forma para resaltar una *característica (feature)*. El procesado de los datos puede obtener estructuras con contenido sintáctico y semántico acerca de los datos leídos.

3.3 INTEGRACIÓN

La tercera actividad del sistema de procesado es la integración, también conocida como fusión (figura 1.2). Esta actividad no es necesaria en un sistema con un solo sensor, pero es extremadamente importante en sistemas con múltiples sensores.



FIGURA 1.2 Fusión de datos procedentes de varios sensores.

3.3.1 Necesidad

Si volvemos al ejemplo del sistema de temperatura de un reactor nuclear donde existen varios sensores de temperatura, necesitamos un procedimiento para conocer cual es la medida real a partir de las que nos ofrecen los sensores. Esto es así porque puede que existan sensores defectuosos o que ya no funcionan y, por lo tanto, están ofreciendo lecturas erróneas. En general, para obtener la medida correcta se pueden adoptar tres posturas:

- Si lo que deseamos es obtener un solo número, se puede obtener la media aritmética de los valores adquiridos.
- Si deseamos un intervalo de valores, entonces tomaremos los valores máximo y mínimo de las lecturas obtenidas.
- Si el número de muestras es elevado, se pueden despreciar los valores mínimo y máximo y después obtener la media de los restantes. De esta forma se elimina el sesgo de los datos
- Si un valor es muy diferente de los otros, podemos despreciarlo como no válido y promediar el resto.

De este ejemplo, vemos como en determinadas situaciones se debe emplear algún mecanismo para fusionar la información y obtener un solo resultado.

Otro ejemplo, aún más cercano, es el sistema sensorial del ser humano compuesto por cinco sentidos: vista, oído, olfato, tacto y gusto. En el proceso de aprendizaje y conocimiento, y en cualquier momento, varios de estos sentidos están actuando simultáneamente y es en nuestro cerebro, donde todos los datos se fusionan y ofrecen un resultado global sobre el entorno. Es más, los sentidos se complementan entre sí: ¿no es cierto que no disfrutamos de una buena comida si tenemos mucho frío o mucho calor? Además, la falta de uno de ellos es suplida (con las obvias limitaciones) por los demás, como evidencian las personas discapacitadas.

De los ejemplos anteriores podemos ahora definir la integración de sensores como el proceso mediante el cual se integran, combinan o fusionan las lecturas procedentes de varios sensores independientes.

En la integración de las medidas de los diferentes sensores debemos resolver varios problemas:

- Puesto que la información obtenida puede no ser homogénea, hay que decidir cómo representarla e interrelacionarla. El primer paso es obtener una abstracción de la medida física, un conjunto de valores, adecuadamente elaborados, que proporcionan un medio común en el que integrar los valores de los diferentes

sensores. En este caso hablamos de sensores abstractos para diferenciarlos de los sensores concretos, término que se refiere al sensor físico utilizado. Así pues, cada sensor concreto tiene asociado un sensor abstracto.

- Cualquier medida va a estar afectada en mayor o menor medida por el ruido. Si se emplea un solo sensor no podremos estar seguros de que el valor obtenido sea el correcto. Podemos reducir el error si realizamos múltiples medidas, aunque entonces podemos tener contradicciones entre ellas. Por ello, es necesario establecer el mecanismo adecuado para obtener la medida "correcta" a partir de ese conjunto de medidas.
- Los sensores pueden destruirse, fallar en su funcionamiento o violar sus límites operativos. Además, puede que su número varíe con el tiempo. Necesitamos una técnica robusta para la integración de los sensores en estos casos.

De todas estas consideraciones se derivan una serie de métodos para integrar la información procedente de varios sensores. Estos métodos se implementan tanto al nivel del hardware como del software del sistema de procesado.

3.3.2 Mecanismos de integración de sensores

Los sensores pueden proporcionar cuatro tipos de datos:

- Competitivos (redundantes).
- Cooperativos (parciales y sin solape).
- Complementarios (parciales y con solape).
- Independientes (no relacionados).

Veamos ejemplos de los cuatro tipos de datos para aclarar sus conceptos:

- Supongamos que usamos tres termómetros para medir la temperatura del vaso de un reactor nuclear. En este caso, las medidas de temperaturas se pueden considerar como datos competitivos.
- Pensemos ahora en una región que dividimos en subregiones o particiones disjuntas, y supongamos que tenemos un sensor que mide una determinada variable en cada una de las subregiones. En este caso, las lecturas de los sensores se pueden considerar como cooperativas, pues no solapándose contribuyen de manera parcial al conocimiento de la información de toda la región.
- Si, a diferencia del caso anterior, quisiéramos dividir una región rectangular en subregiones circulares, aparecerían zonas solapadas, por lo que las lecturas de los sensores se considerarían como complementarias.
- Dos variables físicas pueden o no estar relacionadas. Si están relacionadas, es posible determinar una con la otra, sino decimos que son independientes. ¿Qué significado tiene el integrar dos datos independientes? Hay dos respuestas a esta pregunta: la primera es que puesto que son independientes no tiene sentido integrarlas; la segunda sugiere tomar los datos para añadirlos a registros de una base de datos, tarea que se realizarían en la actividad de integración.

Independientemente del tipo de datos, los sensores pueden ofrecer datos contradictorios o erróneos. Es imperativo, por lo tanto, que los datos se combinen de modo coherente y sistemático, tarea que realiza el *fusionador* dentro del integrador.

Para integrar los fragmentos de esta información de manera adecuada necesitaremos uno o más de los siguientes métodos de integración de sensores.

- **Integración competitiva**

La información redundante o competitiva nos permite la construcción de un resultado más fiable y tolerante a fallos a partir de las lecturas de los sensores. Este tipo de integración requiere las lecturas de los sensores replicados, que idealmente será idénticas, pero que en realidad puede estar afectadas por el ruido. Incluso en el caso de que existan sensores defectuosos, la información de los mismos puede ser ignorada, utilizando los mecanismos apropiados, durante el proceso de integración.

- **Integración cooperativa**

En el caso de disponer de información parcial y no solapada (cooperativa), la integración considerará cada uno de los fragmentos para obtener la información global, como en el caso de la integración complementaria, pero, a diferencia de ésta, si uno o varios sensores ofrecen información errónea, el resultado será erróneo, al no existir posibilidad de recuperar aquella información perdida por no existir solape.

- **Integración complementaria**

Si disponemos de información parcial y solapada (complementaria) de los sensores, la integración deberá reunir la misma para obtener la información completa. Si alguno de los sensores está estropeado, el resultado de la integración puede ser erróneo. La integración complementaria supone la combinación de la información parcial en una única visión global.

- **Integración independiente**

En el caso de sensores independientes, la integración consiste en añadir nuevos campos a los registros de la base de datos donde se almacenan los resultados.

- **Integración temporal**

La integración temporal implica la integración de la información a lo largo del tiempo. Los datos, muestreados a intervalos de tiempo conocidos, deben procesarse para obtener una secuencia temporal.

- **Integración espacial**

En este caso la integración se realiza a lo largo de un espacio dado. Por ejemplo, para percibir un objeto en una escena, tomaremos una serie de imágenes desde diferentes puntos de vista y después integraremos la información de manera adecuada.

3.4 ANÁLISIS

La cuarta y última actividad del sistema de procesado se conoce como *análisis, interpretación de datos o toma de decisión*. Veamos un ejemplo para mostrar la función de la toma de decisiones.

Una imagen digital es un patrón bidimensional de valores de intensidad. El objetivo de un sistema de visión es la interpretación de la imagen o el análisis de la escena. En este caso, el análisis puede ser responder a preguntas como ¿hay un objeto de tipo X presente en la escena? o ¿qué objetos están presentes en la escena?

Para responder a estas preguntas, el sistema de procesado deber evaluar la imagen, procesarla, y en función de esos resultados y de determinadas condiciones sobre los mismos, elaborar una decisión. Esta última tarea es la que se engloba en la actividad de análisis.

4. ORGANIZACIÓN DE SISTEMAS DE PROCESADO DE SENSORES

Como ya vimos, el procesamiento de sensores consistía en cuatro actividades: adquisición, procesamiento, integración y análisis. Existen varias formas de organizar el sistema de procesamiento de sensores. En función de las características del problema a resolver deberemos elegir la forma de organizar el sistema de procesamiento, como veremos ahora, para adecuarlo a nuestras necesidades.

4.1 COLECCIÓN DE SENSORES

Una colección de sensores es un conjunto de sensores dispuestos de una determinada manera. Éstos se pueden disponer en serie, en paralelo o en combinaciones mixtas de estas dos formas básicas como se muestra en la figura 1.3.

La elección de la configuración particular dependerá de la aplicación en la que se vaya a emplear. La integración de la información se va realizando de forma progresiva a través de los distintos sensores hasta obtener un resultado final.

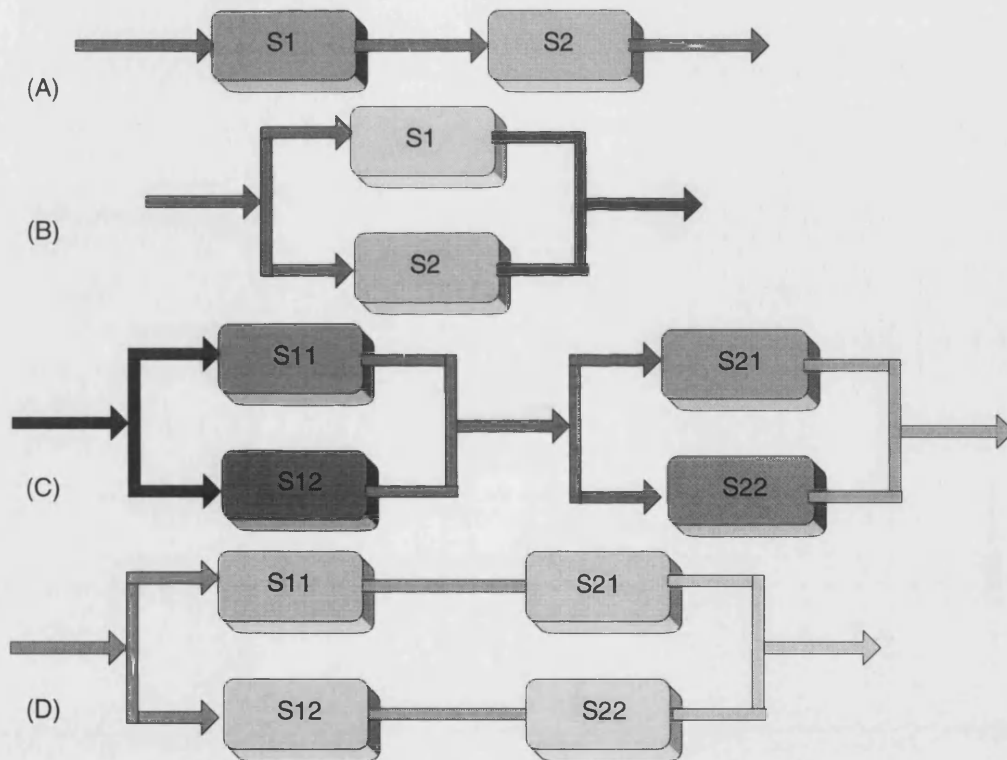


FIGURA 1.3 Colección de sensores. (A) Serie. (B) Paralelo. (C) Serie/Paralelo. (D) Paralelo/Serie.

4.2 SISTEMAS JERÁRQUICOS

En un sistema centralizado, los datos procedentes de los sensores se transmiten a un procesador central para ser combinados. Si el volumen de datos es elevado, esta organización puede requerir anchos de banda considerables. Para estos casos, el sistema de procesamiento de sensores se puede organizar como una jerarquía de subsistemas.

Como caso especial podemos considerar un sistema organizado como un árbol binario completo (figura 1.4), en el cual existe un nodo raíz y una serie de nodos

intermedios cada uno de los cuales tiene dos hijos; asumimos que cada nodo es inteligente (es decir, consiste en un procesador y los sensores asociados). Puesto que cada nodo es inteligente, podemos reducir los requerimientos de comunicación integrando los datos de los sensores en cada nodo intermedio y pasando al nivel superior sólo el resultado de esa integración. El resultado producido por el nodo raíz es la integración de los resultados de integrar los datos de todos los sensores.

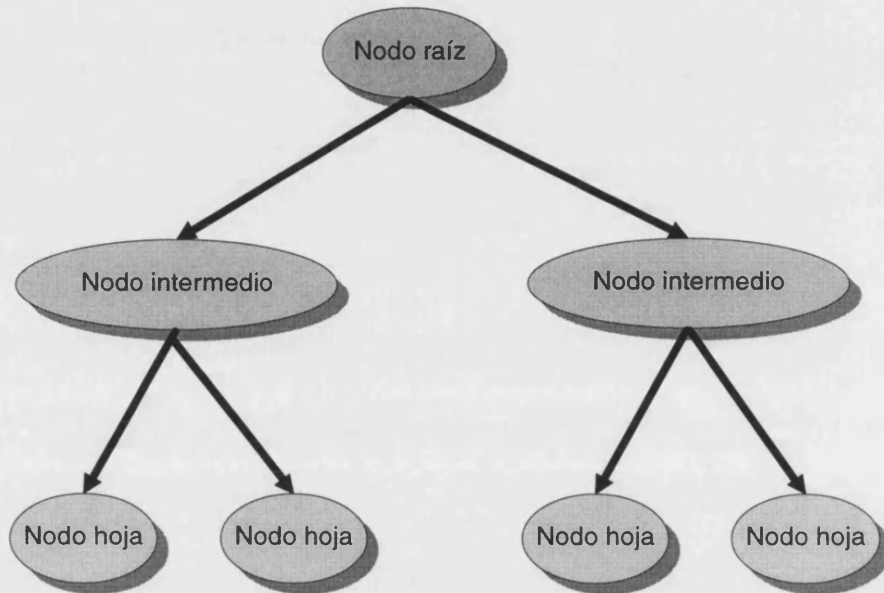


FIGURA 1.4 Sistema de procesamiento jerárquico de sensores organizado en árbol binario. La integración se produce en cada nodo intermedio y en el nodo raíz.

Es habitual tener tres niveles: bajo, intermedio y alto. En cada uno de los niveles tiene lugar un procesamiento diferente sobre los datos, cuyo resultado es pasado al nivel superior, de forma que el último nivel obtiene la decisión. Normalmente la operación que se realiza en cada uno de los niveles es, respectivamente, integración-fusión de datos, integración-fusión de características e integración-fusión de decisiones. Las características se refieren a los datos en cuanto representan una información con mayor sentido semántico de los mismos. A partir de ellas se realizan decisiones parciales mediante las reglas determinadas en cada caso. La decisión final se obtiene de las parciales también a través de procesos de decisión perfectamente especificados.

Por ejemplo, en el caso de procesamiento de imágenes, el nivel bajo convierte los niveles de intensidad (pixels) en características como líneas o bordes. El nivel intermedio realiza el reconocimiento de las formas de la imagen a partir de la integración de las características. Por último, el nivel alto identifica el objeto de la escena.

Consideremos el ejemplo de la figura 1.5. Los datos D_1 y D_2 se combinan en la etapa de integración de datos en la característica F_{12} . Del mismo modo, D_3 y D_4 , D_5 y D_6 y D_7 y D_8 lo hacen en las características F_{34} , F_{56} y F_{78} respectivamente. Las características F_{12} y F_{34} se combinan en la etapa de integración de características en una decisión local *Dec 1-4*, y F_{56} y F_{78} lo hacen en *Dec 5-8*. Por último, las decisiones locales se combinan en la etapa de integración de decisiones en una decisión global *Dec 1-8*.

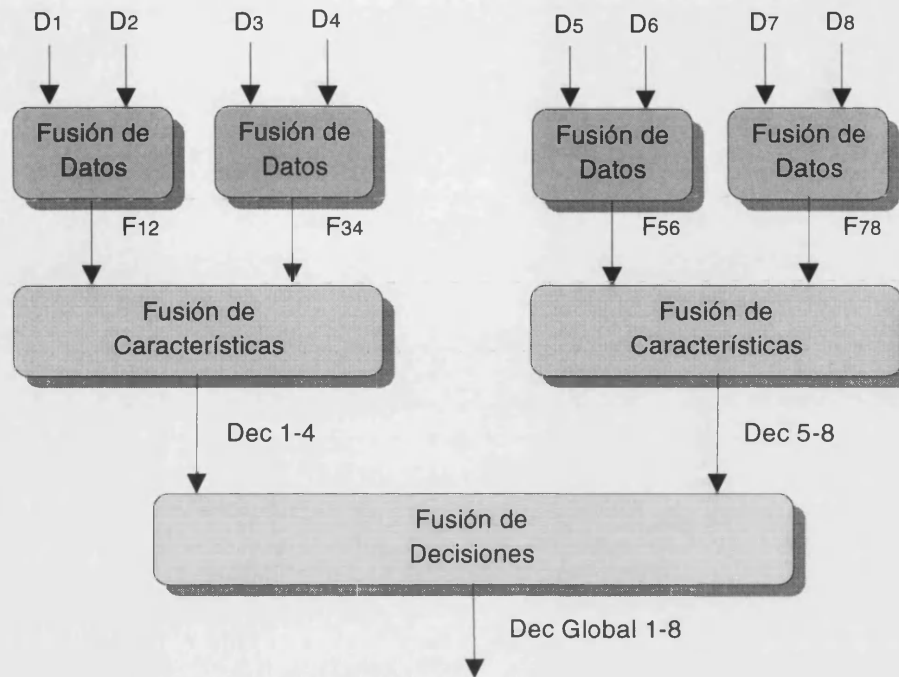


FIGURA 1.5 Integración jerárquica de sensores.

El aspecto interesante de esta organización es que, como se verá más tarde, un incremento en el "tamaño" del problema no resulta en un incremento semejante en la organización del sistema de procesamiento. Es decir, el sistema no crece linealmente con el problema. Como se verá esto es cierto siempre que se cumpla que la fusión de datos y características reduce el volumen de información.

4.3 PROCESADO DE SENSO RES EN ÁRBOL

Otra variante de sistema de procesamiento de sensores es la que se muestra en la figura 1.6, donde los sensores s_1 a s_7 (del mismo tipo y especificaciones) se disponen en forma de árbol binario. Hay cuatro nodos hoja, dos nodos intermedios y un nodo raíz. Los nodos intermedios y el raíz son inteligentes en el sentido de que pueden realizar procesamiento local sobre tres valores de entrada (de los dos hijos y suyo propio) y pasar la salida hacia arriba en el árbol. Así pues, la salida del nodo raíz depende de la función F usada para el procesamiento local.

El procesamiento de los siete sensores requiere tres pasos ejecutados de forma secuencial:

1. $t_1 = F(s_1, s_2, s_3)$
2. $t_2 = F(s_4, s_5, s_6)$
3. $t_3 = F(t_1, t_2, s_7)$

La salida final es $t_3 = F(F(s_1, s_2, s_3), F(s_4, s_5, s_6), s_7)$. Hay que hacer notar que los pasos 1 y 2 pueden realizarse de forma paralela. En general, el procesamiento de n sensores dispuestos en árbol binario requiere un tiempo orden $\log n$.

Obsérvese la diferencia con el sistema jerárquico en el que la totalidad de los sensores es procesada en el primer nivel, mientras que en esta organización los sensores son progresivamente integrados en los sucesivos niveles del árbol.

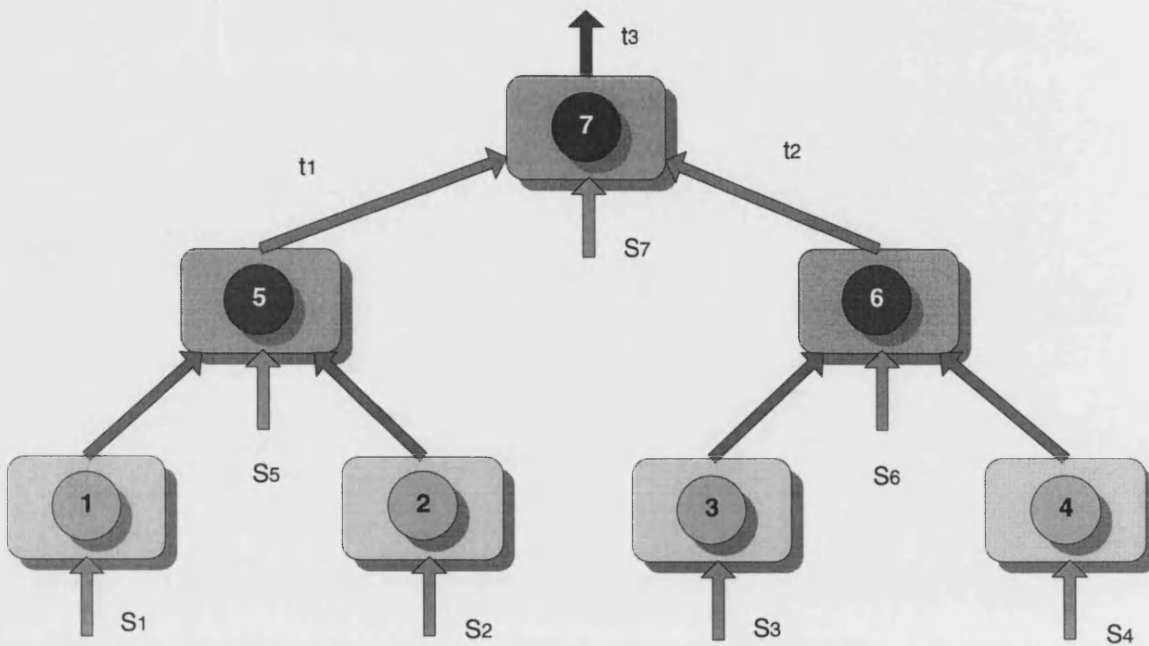


FIGURA 1.6 Integración de sensores dispuestos en árbol binario.

4.4 INTEGRACIÓN MULTISENSOR

La figura 1.7 muestra la integración de varios sensores s_1 , s_2 , s_3 y s_4 (no todos del mismo tipo). Asumimos que s_1 y s_2 son del mismo tipo, s_3 de un segundo tipo y s_4 de un tercero. En este caso, la integración de la información ha de hacerse para asegurar que los datos de los sensores son compatibles.

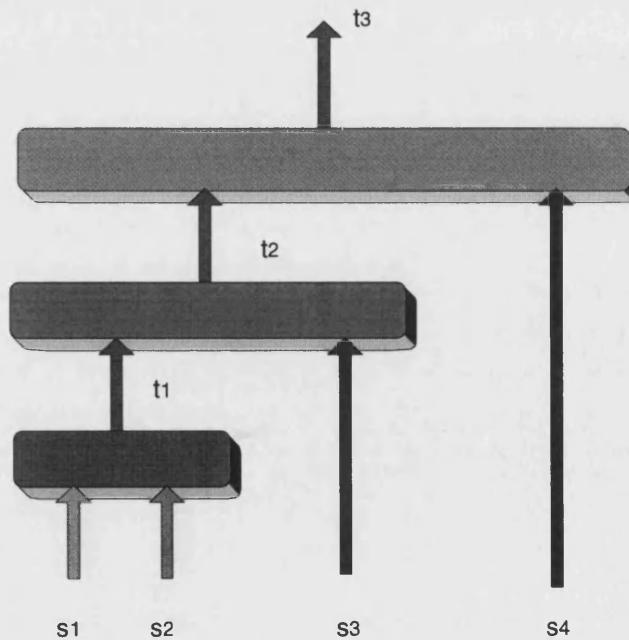


FIGURA 1.7 Integración multisensor con tres tipos diferentes de sensores.

En este caso, el procesamiento de las cuatro lecturas ha de realizarse secuencialmente en tres fases:

1. $t_1 = F_1(s_p, s_2)$
2. $t_2 = F_2(t_p, s_3)$
3. $t_3 = F_3(t_p, s_4)$

La salida final es $t_3 = F_3(F_2(F_1(s_p, s_2), s_3), s_4)$. Hay que aclarar que la función F_1 usa datos competitivos, procedentes de los dos sensores, mientras que F_2 y F_3 usan datos complementarios.

4.5 PROCESADO DISTRIBUIDO DE SENSORES

En los siguientes puntos nos centraremos en aspectos computacionales del procesamiento de sensores en general y, en particular, nos centraremos en el procesamiento destinado a sistemas de sensores donde de las cuatro actividades del sistema de procesamiento de sensores una o más tienen lugar de forma distribuida, sistemas conocidos como de procesamiento distribuido de sensores.

Para comprender mejor los diferentes aspectos de procesamiento de sensores veamos un ejemplo: consideremos un sistema de procesamiento distribuido de sensores para aplicaciones de defensa o vigilancia aérea (figura 1.8). El principal objetivo del sistema es detectar los objetos (o blancos) presentes en el espacio observado. Otros ejemplos pueden ser los sistemas de control distribuido, los sistemas sensoriales para robots, etc.

4.5.1 Descomposición de tareas

Asumamos que existe un número finito de recursos (sensores y procesadores) en el sistema de procesamiento distribuido de sensores. Consideremos un sistema en el que hay N sensores (S_1 a S_N) y P procesadores (EP_1 a EP_p). Los N sensores, por ejemplo, pueden seguir objetos en el espacio de observación y asumimos que son todos del mismo tipo (radares de microondas, radares láser o infrarrojos), por lo que se trata de un sistema basado en sensores físicamente replicados. Tenemos los sensores organizados en P grupos o *clusters*, por ejemplo 3, de N/P sensores cada uno. En nuestro ejemplo, hay tres grupos, cada uno con tres sensores y un procesador para controlarlos.

La tarea principal, T , es seguir los objetos que puedan atravesar el espacio de observación. Consideremos dos posibilidades:

- El espacio de observación es demasiado amplio y por lo tanto no puede ser eficientemente cubierto por ninguno de los grupos de sensores.
- El espacio de observación puede ser cubierto por todos los grupos de sensores, pero se requiere una respuesta en tiempo real para el seguimiento de los blancos.

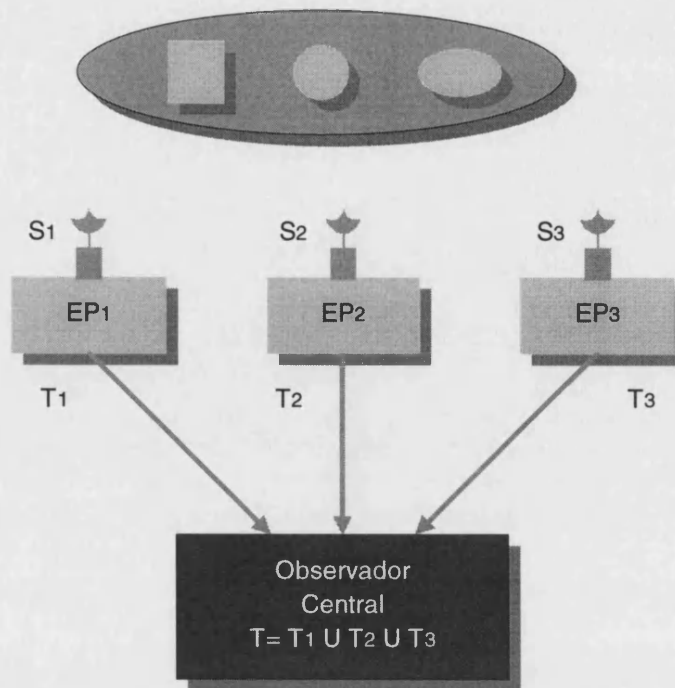


FIGURA 1.8 Ejemplo de sistema de procesamiento distribuido de sensores. La tarea principal T se descompone en tres subtareas T_1 , T_2 y T_3 . Cada subtarea se asigna a un elemento de procesamiento EP_i que controla un grupo de sensores.

En el primer caso, podemos asignar a cada grupo de sensores una zona del espacio. De manera colectiva, cubrirán todo el espacio de observación. En el segundo caso, podemos asignar a cada grupo el seguimiento de un número concreto de blancos; idealmente, cada grupo debería efectuar el seguimiento de un solo blanco.

Así pues, el ejemplo del problema de la vigilancia aérea es tratable bajo el paradigma de *divide y vencerás*. Un problema complejo puede resolverse dividiéndolo primero en un número determinado de subproblemas de manejo y resolución más simples, y después combinando las soluciones parciales en una única solución global.

En nuestro ejemplo, el sistema de procesamiento distribuido de sensores descompone la tarea principal T en P subtareas; a esta operación se la conoce como *descomposición de tareas*. El objetivo de cada subtarea T_i es detectar y seguir el objeto i -ésimo del espacio de observación. Cada tarea tiene asignado un elemento de procesamiento EP_i que controla los tres sensores del grupo.

4.5.2 Procesado local

Cada grupo de sensores replicados dispone de un procesador local. El procesador se encarga del procesamiento local y del control; puede controlar los sensores asignados a él y obtener de ellos los valores obtenidos. Idealmente, los sensores del grupo deberían obtener siempre un valor común, pero en la práctica aparecen diferencias representables mediante distribuciones estadísticas.

Supongamos que cada grupo puede observar sólo parte del espacio, pero los blancos pueden moverse hacia cualquier parte dentro de este espacio. En este caso, haría falta un sistema de comunicación entre los procesadores locales para compartir la información acerca del objeto y conocer cuándo éste cambia de un área a otra.

La figura 1.9 muestra el procesado local que involucra a los sensores S_1 a S_3 , a los procesadores EP_1 a EP_3 y las salidas de los sensores SO_1 a SO_3 .

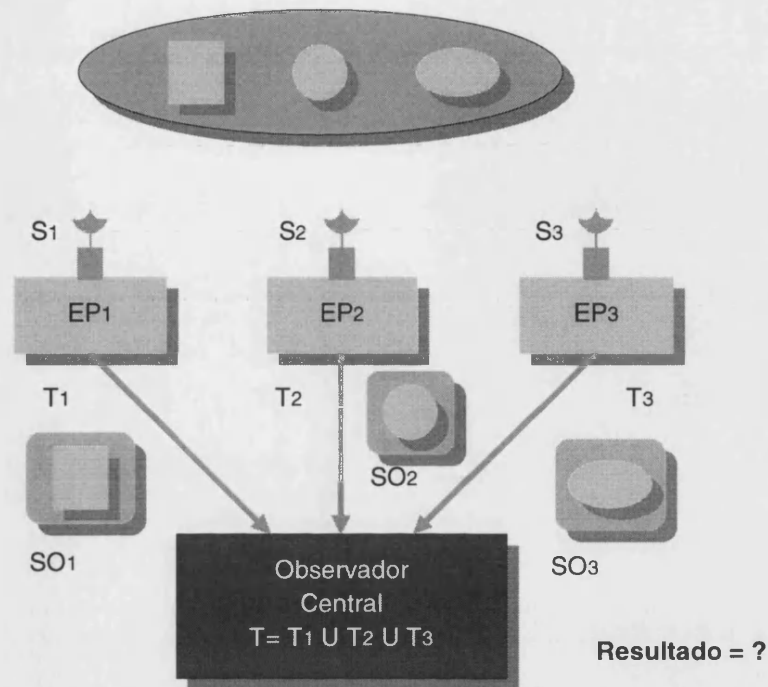


FIGURA 1.9 Ejemplo de procesado local. Las lecturas de los sensores S_1 a S_3 se procesan para obtener las salidas SO_1 a SO_3 .

4.5.3 Integración

El observador central, integrador o fusionador, es el responsable de combinar los datos procedentes de los sensores y/o sus abstracciones. Hay que hacer notar que comenzamos con nueve sensores. Hay tres grupos de tres sensores cada uno. Los tres sensores de cada grupo proporcionan información redundante. El procesado de cada grupo combina la información redundante para obtener una solución de un subproblema – ¿qué objeto es el que el grupo está observando?

De esta manera, al observador central le llegan tres conjuntos de datos, cada uno proveniente de un grupo de sensores. Con esos datos, el observador determina que hay tres objetos en el espacio de observación. En ningún momento se ha hecho mención de la proximidad del observador central a los sensores, y, en general, se comprueba que suelen estar separado por cierta distancia por lo que el resultado del procesado local deberá ser transmitido de alguna manera hasta el observador central.

En nuestro ejemplo, el resultado que obtiene el procesador central es un mapa de los objetos presentes en la totalidad del área de interés. La figura 1.10 muestra la fase de integración que involucra los sensores S_1 a S_3 , los procesadores EP_1 a EP_3 y las salidas procesadas de los sensores SO_1 a SO_3 .

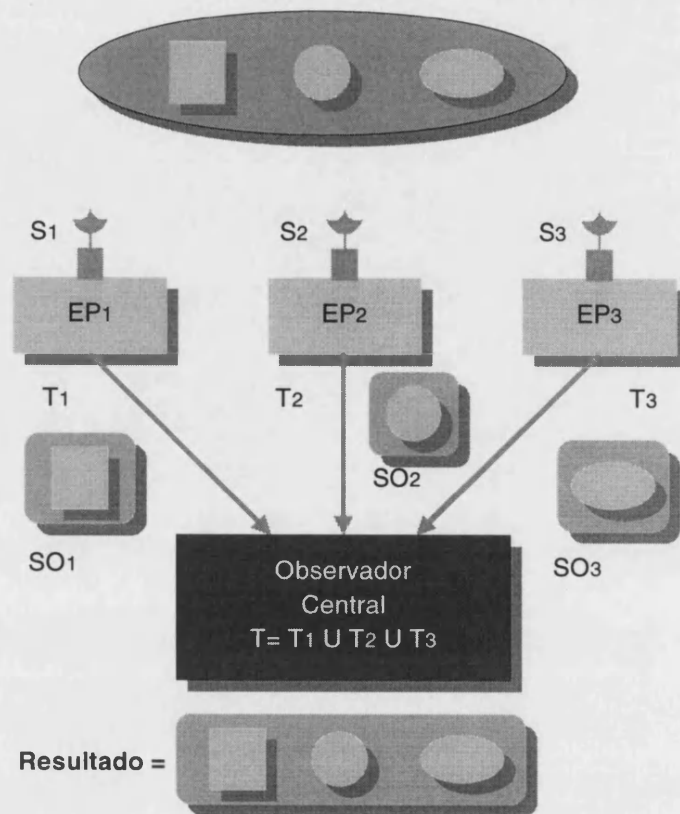


FIGURA 1.10 **Ejemplo de integración.** El observador central integra SO1 a SO3 para obtener un resultado.

4.5.4 Análisis

Se supone que el sistema de procesado tiene una base de conocimiento mediante la cual puede analizar e interpretar los datos y tomar la acción apropiada en función del resultado obtenido. En nuestro caso, el sistema interpreta que hay tres objetos de forma geométrica que ocupan el espacio de observación; depende de la base de conocimiento el que el sistema reaccione de una manera o de otra.

Los sistemas de vigilancia reales no sólo tienen que distinguir e identificar los objetos, sino que también deben obtener la posición, velocidad y trayectoria de los blancos y ofrecer suficiente información para planificar medidas de respuesta efectivas.

4.6 REDES DE SENSORES DISTRIBUIDOS

El empleo de sensores diferentes, inteligentes y distribuidos espacial y geográficamente ha ido creciendo constantemente en aplicaciones tales como la robótica, los experimentos de física de partículas, las imágenes médicas, el seguimiento por radar, la navegación aérea y los sistemas de control de actividades en cadenas de montaje, por nombrar algunas. Estos sistemas, y otros de las mismas características, reciben el nombre de *Redes de Sensores Distribuidos* o *DSN*. De otra manera, podríamos definir una red de sensores distribuidos como un conjunto de sensores inteligentes distribuidos espacialmente y diseñados para obtener datos del medio que les rodea, abstraer de ellos la información relevante e inferir de la misma el objeto observado, derivándose de todo ello la actuación apropiada según el caso.

La red de sensores distribuidos constituye el paso lógico en la evolución de los sistemas de procesamiento de sensores. El diseño y la implementación de este tipo de redes no hubiera sido posible si los avances en la tecnología, sobretodo en los procesadores y las comunicaciones.

En este tipo de redes, a menudo se emplean técnicas de resolución distribuida de problemas o de procesamiento distribuido. Las primeras se refieren a la búsqueda de métodos de resolución de problemas que incluyan descomposición de tareas, hipótesis, integración de datos, etc. en dominios distribuidos. Las segundas tratan de aspectos de arquitectura de computadores, distribución de carga computacional, planificación de la ejecución de los procesos, interconexión de procesadores, etc.

5. ARQUITECTURAS PARA EL PROCESADO DE SENSORES

El diseño de la arquitectura para un sistema de adquisición de datos multisensor depende de varios factores. Entre ellos podemos destacar los siguientes:

1. Topología de la red
2. Modelo de sincronización
3. Comunicación
4. Rutado
5. Requerimientos algorítmicos

Veamos con un poco más de detalle cada uno de estos aspectos.

5.1 TOPOLOGÍA DE LA RED

Un grafo $G=(V, E)$ es un conjunto V de vértices o nodos y otro E de flechas, donde cada flecha conecta un par de vértices. Dependiendo de las restricciones sobre las flechas, podemos tener varios tipos de grafo: no dirigidos, dirigidos, árboles, etc. A menudo, a los grafos en general se les conoce como redes.

Una red de sensores distribuidos se puede modelar como un grafo, donde V es el conjunto de nodos inteligentes (sensor más procesador) y E es el conjunto de flechas (conexiones entre nodos).

Por topología de la red entendemos el modo en que los nodos sensor/procesador se conectan. Para las DSN existen tres topologías representativas de las que discutiremos sus ventajas e inconvenientes.

- Organización en comité
- Organización jerárquica
- Árbol plano

5.1.1 Organización en comité

La organización en comité fue descrita por primera vez por Wesson et al. [4] en el contexto de un sistema distribuido. En este tipo de organización, cada nodo (sensor más procesador) es autónomo, por lo que, a veces, también es conocida como *organización anárquica*. Cada nodo envía información a todos los otros nodos de la red, y si todos ellos están interconectados, la organización es isomórfica a un grafo completamente conectado (figura 1.11).

Si el número de nodos es N , este tipo de organización necesita del orden de N^2 interconexiones para formar un grafo completo. La ventaja de este número grande de conexiones es que el sistema es muy robusto a los fallos, ya que los mensajes de un nodo a otro pueden ser rerutados para evitar los enlaces que no funcionan. Para

valores de N pequeños, la organización puede usarse en redes de sensores, sin embargo, si N es grande, el coste se hace enorme.

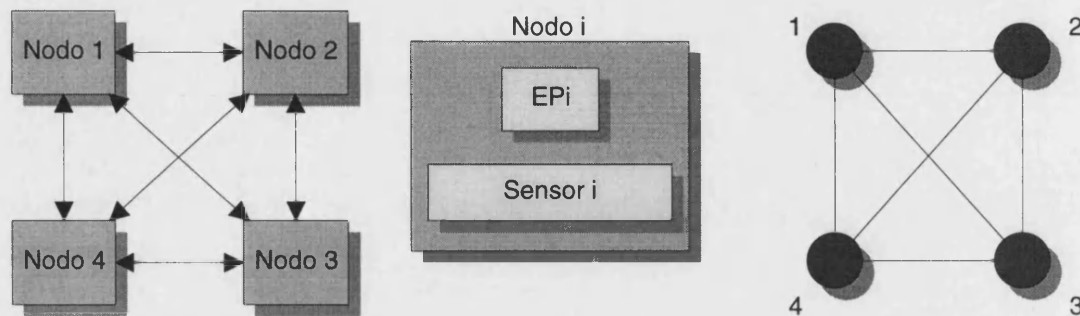


FIGURA 1.11 Organización en comité para cuatro nodos. Cada nodo está formado por un elemento de procesado EP_i y un sensor.

5.1.2 Organización jerárquica

En este tipo de organización los nodos (sensores más procesador) se disponen en niveles jerárquicos y es isomórfica a un árbol.

En cada uno de los niveles del árbol, los nodos reciben la información de sus inferiores, la integran de acuerdo a su función dentro de la jerarquía y envían los resultados hacia el nodo correspondiente en el nivel superior. El nodo a más alto nivel es el nodo raíz, y es el encargado de controlar los nodos de la jerarquía (figura 1.12).

En un sistema con N nodos, se requieren del orden de N interconexiones, por lo que el coste de comunicación de este tipo de organización es menor que en el caso de comité. Sin embargo, puede producir estimaciones inexactas por dos razones. La primera es que no se permite compartir los datos entre los sensores en el nivel inferior. En segundo lugar, los errores, si se producen, se acumulan a medida que se sube en la jerarquía.

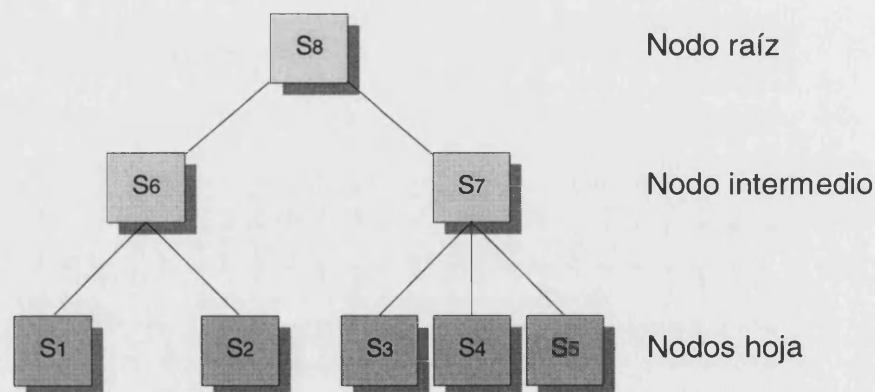


FIGURA 1.12 Organización jerárquica de 8 nodos. S_1, \dots, S_5 son los nodos hoja, S_6 y S_7 son los nodos intermedios, S_8 es el nodo raíz.

5.1.3 Organización en árbol plano

Puesto que tanto la organización en comité como la jerárquica tienen ventajas y limitaciones, sería normal preguntarse si una organización híbrida de ambas lograría incluir las ventajas de ambas.

Una de las posibles organizaciones es la organización en árbol plano, isomórfica a un conjunto de árboles binarios cuyos nodos raíz se interconectan totalmente [5][6] (figura 1.13). Este tipo de organización tiene las siguientes características:

- Los sensores se agrupan en bloques llamados *clusters*.
- Cada nodo sensor/procesador consiste en un procesador y uno o varios sensores asociados a él. El sensor mide la variable física bajo el control y la dirección del procesador.
- Cada *cluster* está organizado como un árbol binario formando así una jerarquía de nodos.
- Existe más de un nodo raíz por la definición misma de este tipo de organización.
- Cada procesador en un nodo se comunica con sus hijos o sus padres a través de canales mediante mensajes.
- No existe un reloj global que sincronice los procesadores del sistema, empleando cada uno su propio reloj local. Sin embargo, es necesaria la sincronización en los sensores para la realización de la medida.
- Cada canal de comunicación dispone de un buffer para almacenar los mensajes.
- Cada nodo raíz controla su propio *cluster*. Actúa combinando la información de los nodos para obtener un solo resultado.
- Todos los nodos raíz están interconectados entre sí.
- Obtiene las ventajas de la organización en comité y jerárquica al limitar el número de canales de comunicación requeridos y, al mismo tiempo, limitando el crecimiento de la complejidad de la información.
- Permite la integración de sensores en tiempo real.
- Tras la lectura de la información por parte del sensor, el procesador asociado la traduce en una estimación abstracta, añade la información temporal adecuada y la pone en el buffer de mensajes asociado.
- Esta organización no es muy robusta. Se puede demostrar que el resultado global obtenido por la red varía cuando se intercambia un sensor correcto con uno que ha fallado dentro de un *cluster*, pero no varía si ese intercambio se hace con otro [7].

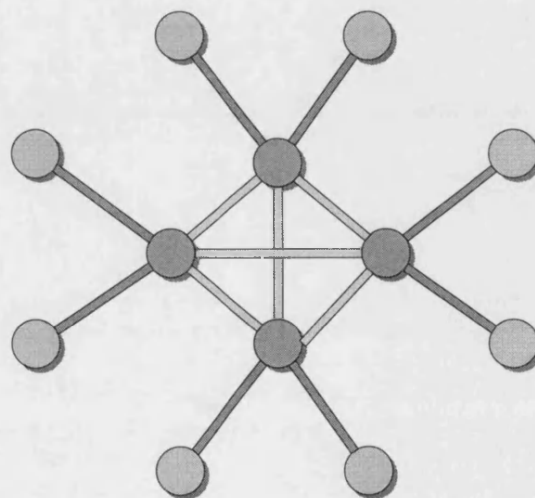


FIGURA 1.13 Diagrama de una estructura de árbol plano compuesta de 4 *clusters* de tres nodos.

Con la excepción del nodo raíz de cada *cluster*, cada nodo funciona como un sensor abstracto para producir una estimación abstracta (un rango de valores dentro del cual estamos seguros de que reside el valor de la variable física) que envía al nodo predecesor en el *cluster*. Esta estimación puede ser un intervalo, en el caso de un nodo hoja, o tres intervalos en el caso de un nodo intermedio (los dos intervalos de los nodos hoja por debajo de él y el suyo propio).

Para esta organización hacemos dos suposiciones:

- La DSN es un árbol binario completo en cada *cluster* y en el que la información fluye unidireccionalmente desde los nodos más bajos a los más altos.
- Los sensores están distribuidos espacialmente en un rango máximo de unos 100 metros. Se supone esto para simplificar la sincronización de la toma de datos de los sensores con un reloj central y los problemas de resincronización.

Para que el árbol plano funcione correctamente, la relación entre el número de sensores/procesadores en cada *cluster* con respecto al número total de sensores/procesadores debe ser pequeña, es decir, el número de *clusters* puede ser elevado, pero cada *cluster* dispone de pocos sensores/procesadores. Debido al tamaño limitado de los *clusters*, los retardos y el intercambio de mensajes entre los procesadores no reducen las prestaciones del sistema.

Si llamamos N al número total de sensores y P el número medio de sensores en cada *cluster*, habrá en media N/P *clusters*. El número de interconexiones necesarias entre los nodos raíz de cada *cluster* es del orden de N^2/P^2 , ya que al tener pocos nodos en cada *cluster*, el número de interconexiones viene marcado por la comunicación total entre los N/P nodos raíz de cada *cluster* que es de ese orden. Como $N < N^3 < N^2/P^2$ si $N < P^2$, cosa que se cumple al introducir la condición de un número pequeño de nodos por *cluster*, la organización de árbol plano requiere menos ancho de banda que las dos anteriores.

Sin embargo, un aumento del número de *clusters* también conlleva un aumento del número de niveles y el retraso en la señal, así como el número de mensajes necesarios para la comunicación en los *clusters*. Por otra parte, un valor bajo de P mejora el rendimiento de cada *cluster* pero incrementa los requerimientos de ancho de banda. Así pues, la elección de P es un compromiso entre ancho de banda y eficiencia de la red y por lo tanto se trata de un parámetro crítico.

5.2 MODELO DE SINCRONIZACIÓN

Puesto que la salida del sensor varía con el tiempo, es necesario que la integración se realice con valores temporalmente próximos de las estimaciones de los sensores. Esto se logra incluyendo en cada estimación una marca de tiempo, es decir, una información acerca del momento en que el sensor realizó la estimación. Si las estimaciones de los diferentes sensores deben sincronizarse para obtener una estimación final en cada nodo raíz, los sensores deben muestrearse aproximadamente al mismo tiempo.

5.2.1 Modelo de reloj

En un sistema distribuido, no hay reloj de sincronización central que regule la actividad de cada nodo. En vez de esto, cada nodo está bajo el control de su reloj local. Puesto que el sensor responde a actividades en tiempo real, es conveniente que el reloj de cada nodo proporcione información de la hora actual para poder incluir esta información con la estimación realizada. Más aún, puesto que las estimaciones de los diferentes sensores han de integrarse, es conveniente hacer que los tiempos proporcionados por los nodos sensores estén próximos los unos a los otros.

El reloj en cada nodo puede no ser preciso debido a una gran variedad de razones, como la deriva o los cambios en la temperatura. Cada reloj por lo tanto, debe ser sincronizado periódicamente con otro reloj más preciso. Asumiremos la existencia de

un servidor de tiempo central situado en algún elemento de procesado de la red que, proporcional el tiempo $C(t)$ cuando se le pide en el instante t .

5.2.2 Comportamiento del reloj y aspectos de sincronización

Sea $C_p(t)$ el tiempo proporcionado por el reloj situado en el elemento de procesado p en el instante t . Supongamos que en un instante t_1 , el tiempo $C_p(t_1)$ proporcionado por p es mayor que $C(t_1)$ proporcionado por el servidor central.

Hagamos ahora que p se sincronice con el servidor central de tiempo en el instante t_2 . Es posible que ahora el nuevo tiempo $C_p(t_2)$ sea menor que $C_p(t_1)$ por lo que una estimación enviada más tarde que el instante $C_p(t_2)$ parecerá como si hubiera sido enviada antes que la de $C_p(t_1)$.

Si las estimaciones no se integran de modo FIFO en el nodo (es decir, se van poniendo en una cola sin ser ordenadas cronológicamente), puede que se creen problemas. Por lo tanto, un requerimiento para el reloj situado en un procesador, es que el tiempo proporcionado siempre ha de incrementarse monótonamente. Esta monotonocidad puede lograrse adelantando o retardando el reloj cada vez que se requiere una corrección del servidor central [8].

5.2.3 Resincronización de reloj

Puesto que los relojes en servidor central y en cada elemento de procesado derivan, deben ser ajustados periódicamente. En [9] se puede encontrar la demostración de que el periodo entre resincronizaciones tiene que ser:

$$T_c \leq \frac{\varepsilon - 2\sigma T_s - (\varepsilon_{\max} - \varepsilon_{\min})}{k} \quad (1.1)$$

donde T_s es el periodo entre resincronizaciones del servidor central, σ es la deriva máxima permitida del servidor central, ε es la desviación máxima permitida en el reloj de cada elemento de procesado, k es la deriva máxima permitida en cada elemento de procesado, ε_{\min} y ε_{\max} son los valores mínimo y máximo del retardo en la recepción del mensaje enviado por el servidor central en cada elemento de procesado. La ecuación anterior expresa que el tiempo entre resincronizaciones, T_c , ha de provocar un error debido a la deriva del reloj del elemento de procesado menor que el propio del reloj menos un valor que depende del error del servidor central y del tiempo que tarda en llegar el valor desde ese servidor al elemento de procesado.

5.3 RUTADO

El rutado es el proceso de enviar una información de un nodo fuente a un nodo destino a través de una red de interconexión. Los algoritmos de rutado son elementos esenciales para el correcto funcionamiento de una red de procesadores.

Para las redes de sensores distribuidos, el problema no es menor, sobretudo en lo que se refiere a la comunicación entre los procesadores [10], no en vano, el componente principal de una DSN es el procesado coordinado entre los procesadores de la red. Cada sensor actúa como una fuente de información que se comunica con algunos o todos los nodos de la red para iniciar el proceso de tratamiento de la información. La interacción entre las fuentes es una operación costosa en sistemas distribuidos, por lo que es importante minimizar la comunicación entre procesadores y diseñar modos eficientes de encaminar la información en la red.

Cada procesador asociado a uno o varios sensores recibe un bloque de datos a intervalos regulares y con una tasa generalmente fija. El nodo procesa los datos y envía la información a otro u otros nodos de la red, en función de la técnica empleada en la resolución del problema. Puesto que la generación de datos es repetitiva, es obligado rutar la información de manera eficiente.

Los requerimientos para el rutado de la información en una DSN son los siguientes:

- Toda la información generada por un sensor debe formar un paquete; de otra forma, la pérdida o retraso en la recepción de varios paquetes puede llevar a rechazar todo el dato por inválido. La razón de este requerimiento es que de ese modo se acelera el proceso de inferencia y se reduce el tamaño de las colas. Este modo de trabajo es contrario al empleado en las redes de comunicación donde la información se divide en varios paquetes que pueden ser encaminados por caminos distintos y reordenados en el destino con la única penalización de un tiempo de transferencia total mayor.
- En la mayoría de aplicaciones con DSN, los datos de los sensores se generan y transmiten en cada ciclo de sensado. Puesto que el intercambio de datos es casi continuo, los protocolos de comunicación deben diseñarse de forma que no se genere un reconocimiento (*acknowledge*) por cada paquete de información que se envía. Esta forma de operar aligera el tráfico de la red teniendo en cuenta el tamaño de la DSN.
- Al no usar mensajes de reconocimiento, pueden existir paquetes anteriores que aún no han sido procesados cuando llega uno nuevo. En este caso podemos ignorar el paquete antiguo que podría identificarse gracias a la marca temporal. Sin embargo, es necesario asegurar que no se pierden datos al ignorar paquetes anteriores.
- En aquellos casos en los que la DSN ha de trabajar en ambientes hostiles como reactores nucleares, ambientes radioactivos, etc. , es muy conveniente el empleo de protocolos de comunicación punto a punto debido a su fiabilidad. Los resultados de los estudios de redes de computadores son de gran utilidad a la hora de adaptar estas redes a las peculiaridades de nuestra DSN.

Cada uno de los nodos de la DSN se tiene que diseñar, en los aspectos relacionados con el rutado, de acuerdo con la repuesta a dos preguntas fundamentales. Por una parte hay que plantearse qué información ha de transmitir el nodo; la respuesta a esta pregunta es compleja y muy dependiente del problema concreto de que se trate. Por otra parte cómo tiene que comunicarse esa información; la respuesta a esta pregunta está sujeta a la consideración de dos restricciones de comunicación importantes: retardos y fiabilidad. Veámoslas a continuación con un poco más de detalle.

5.3.1 Restricciones en la comunicación

El rendimiento de cualquier sistema de procesadores que trabaje de forma coordinada depende de la velocidad de procesado de cada uno de ellos, y de la de los enlaces de comunicación que los unen. La comunicación es, de los dos, el mayor problema puesto que el tiempo necesario para la comunicación es grande si la red de interconexión también lo es. En las DSN, los sensores generan datos a intervalos regulares de forma repetitiva. Es, por lo tanto, esencial asegurar que cada dato es enviado y recibido en el nodo destino en un tiempo finito, antes que los datos del siguiente ciclo lleguen a ese nodo. Por lo tanto, existe un límite superior para el máximo retardo permisible en la red. Si, además, la red trabaja en un ambiente hostil, las consideraciones de fiabilidad de la red también serán importantes.

5.3.2 Restricciones debidas al retardo

En una DSN, un nodo n_i ha de transmitir una unidad de información a otros nodos en un tiempo t_d , que representa el retardo de transmisión máximo permisible. La pregunta es si es posible con una determinada topología de la red asegurar esta restricción.

Los factores que afectan al retardo de transmisión en la DSN incluyen la capacidad del canal, el número de líneas en la ruta, retardos de propagación y tamaño

del paquete. De los resultados que se deducen de la teoría de colas [11], se tiene una aproximación al retardo medio total T :

$$T = \frac{L}{C(1-\rho)} + t_p \quad (1.1)$$

donde L es la longitud media de los paquetes en bytes, C la capacidad del canal en bytes/s, t_p es el retardo de propagación y ρ la carga en la línea, es decir, un coeficiente que indica la disponibilidad del canal para nuestra transmisión.

Para cualquier red, los parámetros L , C y ρ son generalmente conocidos con un grado de aproximación aceptable. En redes dinámicas, es decir, aquéllas en las que el camino que siguen los datos de un origen o un destino dados puede variar en diferentes instantes de una misma transmisión, será necesario evaluar t_p de alguna manera. Para ello se puede tomar el máximo retardo en la red, que coincidirá con el camino más largo entre dos nodos.

Si modelamos la red como un grafo $G(V, E)$ como ya se ha visto anteriormente, podemos definir $d(u, v)$ como la longitud del camino más corto entre los nodos u y v ; $u, v \in V(G)$, medida como el número de arcos en ese camino. El diámetro $D(G)$ de un grafo G es la longitud del mayor $d(u, v)$ para todo $u, v \in V(G)$, es decir:

$$D(G) = \max_{u, v \in V(G)} d(u, v) \quad (1.2)$$

Así pues, el diámetro del grafo correspondiente a la red nos da una medida del máximo retardo de transmisión.

5.3.3 Restricciones debidas a la fiabilidad

Una consideración vital para el correcto funcionamiento de la DSN es la supervivencia de la red a lo largo del tiempo en condiciones adversas. Los nodos y los enlaces de la DSN pueden fallar debido a diversos factores considerando el hecho de que la DSN trabaja en un ambiente hostil. Por lo tanto es necesario mantener en todo momento el flujo de información entre los nodos.

Cuando falla un nodo o un enlace, es necesario encontrar un camino alternativo para rutar los mensajes. Esto implica que la red tenga el grado de conectividad necesario para poder hallar ese camino. Este requerimiento ha de tenerse en cuenta en el momento de diseñar la topología de la red. Supongamos que limitamos el número de enlaces que pueden fallar sin que la red deje de funcionar, y llamemos L a este número. En ese caso, cada nodo de la red deberá tener $L+1$ enlaces para asegurar que la red continúa funcionando aún en el caso en que fallen L enlaces, es decir, el grafo deberá ser $(L+1)$ -conexo. De igual modo, para que la red siga funcionando cuando fallan k nodos, deberá ser $(k+1)$ -conexa. Por lo tanto, es posible mantener la red conectada a pesar de los fallos en los enlaces o en los nodos sin más que incrementar de manera adecuada la conectividad de la misma. Sin embargo, el aumento de la conectividad lleva asociado un incremento en el diámetro de la red y por lo tanto en los retardos.

Supongamos que $D(G)$ es el diámetro de un grafo G correspondiente a una determinada topología de la red de interconexión. Sea k la conectividad del grafo y n su orden (el número de nodos del grafo), y supongamos que permitimos como mucho m nodos fallidos en cualquier instante. Según se deduce en [11], el valor de k que, con m nodos fallidos en la red, proporciona un diámetro D' para el cual los retardos de transmisión no supera t_p , es:

$$D' \leq \frac{n-m-2}{k-m} + 1 \quad (1.3)$$

De la ecuación anterior podemos deducir la conectividad k del grafo, para un valor fijo de m , tal que su diámetro D' asegure que el retardo está dentro de los valores deseados y que además los fallos en los nodos no provoquen retardos inaceptables. En [11] también puede encontrarse la expresión para el caso de fallos en los enlaces.

6. OBJETO DE LA TESIS

El objeto de la tesis es el estudio de las arquitecturas para el procesado y fusión jerárquica en redes de sensores distribuidos y la introducción de mejoras en las mismas. El capítulo 2 presenta, en primer lugar, una clasificación de las arquitecturas multiprocesador adecuadas para este tipo de sistemas, una descripción de las tecnologías de redes de interconexión empleadas en los sistemas para obtener elevadas prestaciones y un estudio analítico de las arquitecturas jerárquicas y su comparación con las arquitecturas paralelas en los casos donde la probabilidad de detección de un determinado suceso es muy baja. La comparación de esas dos opciones se realiza mediante un factor de calidad requerida definida con tal fin.

Tras ese estudio, se presenta una modificación de la arquitectura jerárquica, destinada a disminuir la función de coste, basada en la localidad de los datos y su agrupamiento. Se demostrará que en esos casos el sistema posee una función de coste menor que cuando no se aplican.

El tercer capítulo presenta la introducción de las técnicas descritas en una aplicación particular: el sistema de adquisición de un detector de partículas de un experimento de física de altas energías, que reúne los requisitos de un sistema de sensores distribuidos donde el volumen de información es elevado y la probabilidad de detección del suceso muy baja. Se presenta una comparación entre el sistema jerárquico de adquisición empleado normalmente y el modificado con la introducción de las aportaciones del capítulo 2.

El capítulo 4 presenta la verificación experimental de los resultados del capítulo 2. Para ello, se ha desarrollado un prototipo para el agrupamiento y preprocesado de datos y se ha adaptado a uno de los subdetectores del experimento presentado en el capítulo 3. Se describen las pruebas realizadas, así como los resultados obtenidos de las mismas.

El capítulo 5 expone las conclusiones obtenidas del trabajo realizado, así como el trabajo futuro previsto como continuación del mismo.

7. BIBLIOGRAFÍA

- [1] Iyengar, S. S. y Parameswaran. *Autonomous intelligent systems*. 1995.
- [2] Myers, R. H. et al. *Reliability engineering for electronic systems*. Ed. John Wiley and sons Inc., Nueva York, 1964.
- [3] Durrant-Whyte, H. F. *Sensor models and multisensor integration*. I. J. Robotics Research, 7(6), págs. 97-113, 1988.
- [4] Wesson, R. et al. *Network structures for distributed situation assessment*. IEEE Trans. on System, Man, Cybernetics, SMC 11(1), págs. 5-23, 1981.
- [5] Iyengar, S. S. et al. *A tree architecture for sensor fusion problems*. Proc. of SPIE Technical symposium on sensor fusion, Orlando (Florida), 1990.

- [6] Iyengar, S. S. y Thomas, D. *A distributed sensor network structure with fault-tolerant facilities*. Proc. of SPIE Symposium on advances in intelligent systems, 1989.
- [7] Jayasimha, D. N. et al. *Information integration and synchronization in distributed sensor networks*. IEEE Trans. on systems, man, cybernetics, 21(5), págs. 1032-1043, 1991.
- [8] Gusella, R. y Zatti, S. *The accuracy of the clock synchronization achieved by TEMPO in berkeley UNIX 4.3 BSD*. IEEE Trans. on software engineering, págs. 847-853, 1989.
- [9] Iyengar, S. S. et al. *Advances in distributed sensor integration. Application and theory*. Prentice Hall, 1995.
- [10] Yemini, Y. *Distributed sensor networks: an attempt to define the issues*. Proc. Distributed sensor network workshop, págs. 53-66, Carnegie Mellon University, Pittsburgh, 1978.
- [11] Bond, J. y Pyerat, C. "*Diameter vulnerability in networks*" en Graph theory with applications to algorithms and computer science, págs. 125-149. Ed. John Wiley and sons, New York, 1985.

CAPÍTULO

2

EVALUACIÓN DE ARQUITECTURAS PARA EL PROCESADO DE SENSORES

1. INTRODUCCIÓN	1
2. CLASIFICACIÓN DE LAS ARQUITECTURAS	1
3. ARQUITECTURAS MIMD	5
3.1 Clasificación	5
3.1.1 Sistemas multiprocesador	5
3.1.2 Sistemas multicomputador	7
3.2 Parámetros de excelencia de los sistemas MIMD	8
3.2.1 Escalabilidad	8
3.2.2 Incremento de velocidad y eficiencia de un sistema multiprocesador	9
4. REDES DE INTERCONEXIÓN EN SISTEMAS MIMD	12
4.1 Sistemas multiprocesador	13
4.2 Sistemas multicomputador	16
4.3 Tecnologías de redes de interconexión	18
4.3.1 Scalable Coherent Interface (SCI)	18
4.3.1.1 Características	18
4.3.1.2 Protocolo de transmisión	19
4.3.2 Topologías interconexión	20
4.3.2.1 Integrados para el estándar SCI	21
4.3.3 Asynchronous Transfer Mode (ATM)	22
4.3.3.1 Aspectos y conceptos básicos de ATM	23
4.3.3.2 El protocolo ATM	24
4.3.3.3 Arquitectura genérica de un conmutador	25
4.3.4 Fibre Channel (FC)	27
4.3.4.1 Topología Fibre Channel	27
4.3.4.2 Estructura del protocolo FC	28

5. ARQUITECTURAS PARA LA INTEGRACIÓN DE SENSORES	31
5.1 Problemas con elevada tasa de datos	31
5.1.1 Calidad requerida de un sistema	33
5.1.2 Parametrización de las arquitecturas paralela y jerárquica	34
5.1.3 Evaluación de la calidad requerida	37
5.2 Introducción de elementos de preprocesado	40
5.2.1 Regiones de Interés	40
5.2.2 Agrupamiento de datos	46
5.3 Desaleatorizadores	49
6. BIBLIOGRAFÍA	51

1. INTRODUCCIÓN

En el capítulo anterior hemos presentado diferentes aspectos de los sistemas de procesado de sensores y las redes de sensores distribuidos en las cuales nos centraremos. En este caso, ya se apuntó la necesidad de emplear sistemas de procesado distribuido para la integración de la información procedente de los sensores.

En este capítulo se va a estudiar las soluciones posibles desde el punto de vista de la arquitectura de computadores al problema de la implementación del sistema de procesado distribuido de sensores, para después centrarnos en un modelo de arquitectura jerárquica para la integración de sensores y proponer una mejora basada en la introducción de elementos de preprocesado entre los niveles de la misma.

2. CLASIFICACIÓN DE LAS ARQUITECTURAS [1]

Existen varios esquemas de clasificación de las arquitecturas de computadores en función de las variables que se tomen para su realización.

La clasificación de Flynn (1966) [2] se basa en la multiplicidad de flujos de instrucciones y datos en un sistema computador. El esquema de Feng (1972) [3] se basa en la confrontación procesamiento serie frente a procesamiento paralelo. La clasificación de Händler (1977) [4] viene determinada por el grado de paralelismo y encauzamiento en los diferentes subsistemas del computador.

Existen otras clasificaciones, como las propuestas por Reddi y Feurstel (1976) [5], Skillicorn (1988) [6] y Dasgupta (1990) [7]. Sin embargo, ninguna de todas ellas ha sido tan ampliamente usada como la de Flynn.

Clasificación de Flynn. Multiplicidad de flujos de instrucciones y datos

En general, los computadores pueden clasificarse en cuatro categorías de acuerdo con la multiplicidad de los flujos de instrucciones y datos. El término flujo indica una secuencia de elementos (instrucciones o datos) que ejecuta o sobre los que opera un único procesador. Un flujo de instrucciones es una secuencia de instrucciones ejecutadas por la máquina; un flujo de datos es una secuencia de datos que incluye los datos de entrada y los resultados parciales o totales solicitados o producidos por el flujo de instrucciones.

Atendiendo a la multiplicidad del hardware provisto para atender a los flujos de instrucciones y datos, Flynn realizó una ordenación de las arquitecturas en cuatro clases (figura 2.1):

- Flujo de instrucciones simple - flujo de datos simple (SISD)
- Flujo de instrucciones simple - flujo de datos múltiple (SIMD)
- Flujo de instrucciones múltiple - flujo de datos simple (MISD)
- Flujo de instrucciones múltiple - flujo de datos múltiple (MIMD)

Veamos a continuación una pequeña descripción de cada uno de los tipos de arquitecturas enumeradas.

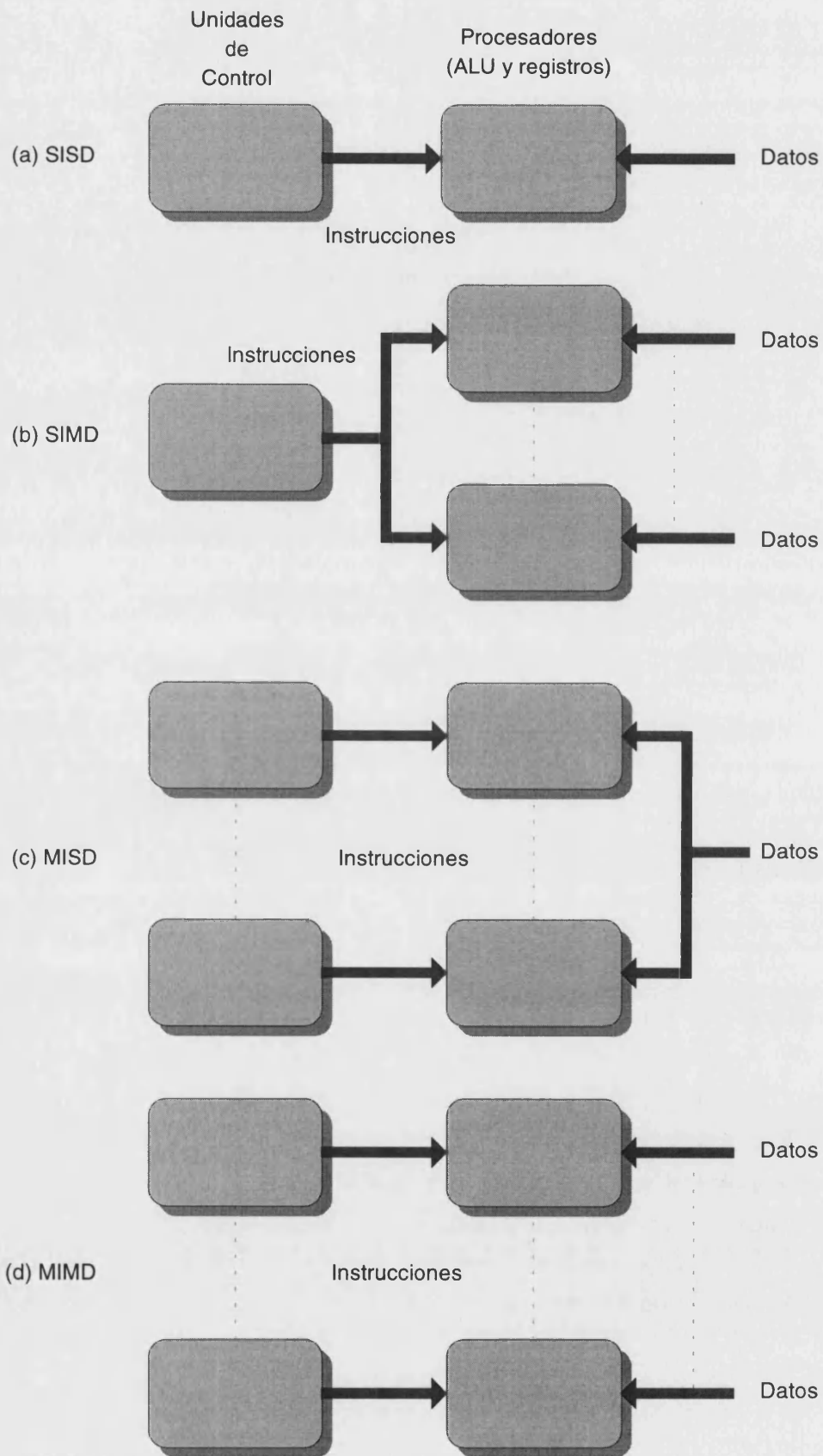


FIGURA 2.1 Clasificación de Flynn.

Organización SISD (figura 2.1a). Representa a la mayoría de los computadores serie disponibles actualmente. Las instrucciones se ejecutan secuencialmente pero pueden estar solapadas en las etapas de ejecución (segmentación encauzada). Un computador SISD puede tener más de una unidad funcional, pero todas ellas están bajo el control de una única unidad de control.

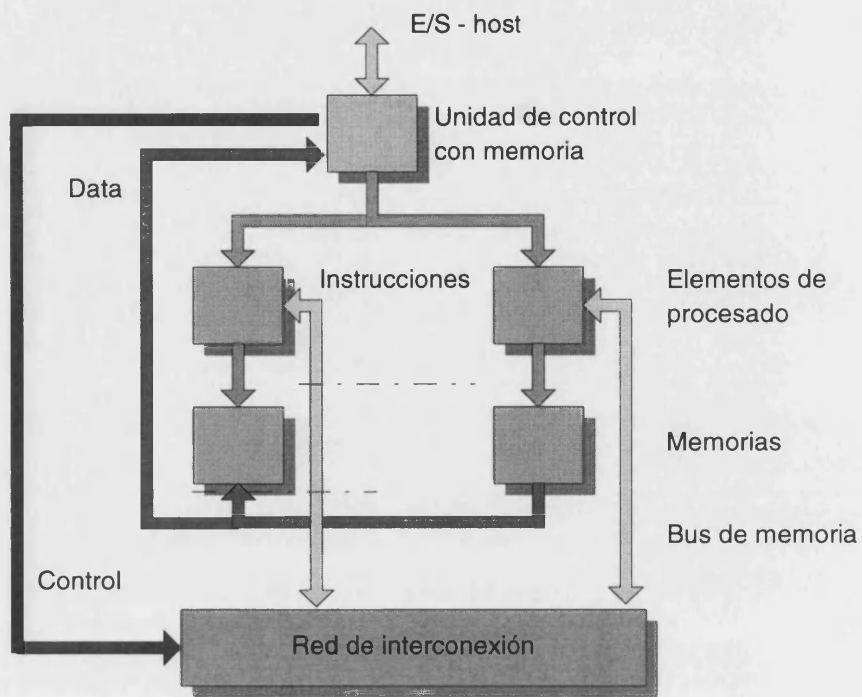
Organización SIMD (figura 2.1b) En esta clase se sitúan los procesadores matriciales en los que existen varias unidades de procesado trabajando sobre flujos de datos distintos pero ejecutando la misma instrucción proporcionada por una única unidad de control.

Atendiendo a la organización de la memoria, tenemos sistemas SIMD con memoria local o con memoria global en función de si los procesadores acceden a la memoria directamente o a través de una red de interconexión. La figura 2.2 aclara estos conceptos.

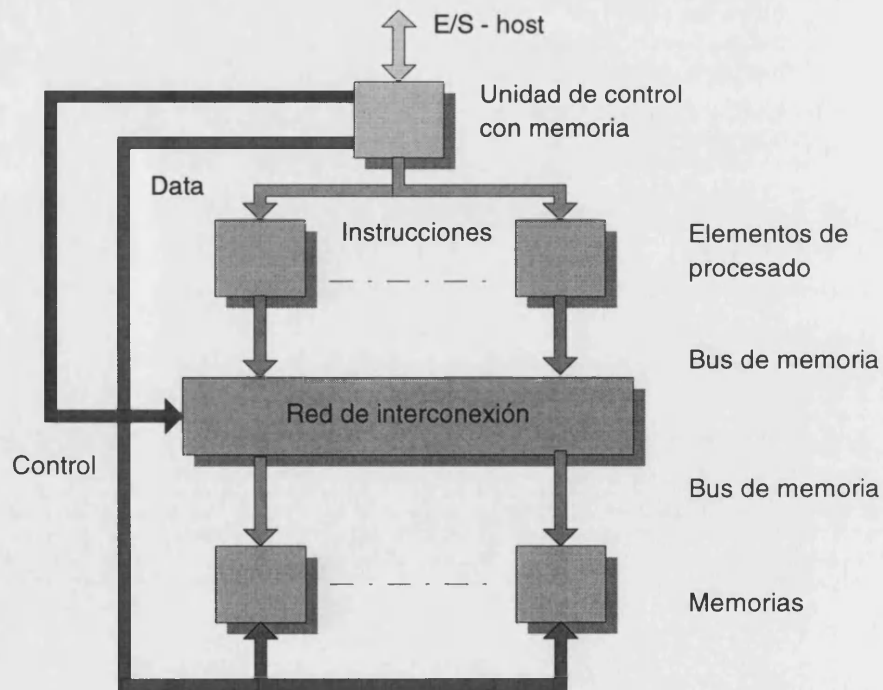
En función del elemento sobre el que se procesa, tenemos los sistemas de sección de palabra (*word-slice*) en los que el procesamiento se realiza sobre palabras binarias, y los sistemas de sección de bits (*bit-slice*) en los que el procesamiento se realiza bit a bit y con aplicaciones básicamente en procesado de imágenes.

Organización MISD (figura 2.1c). Este tipo de organización se caracteriza por la existencia de varias unidades procesadoras cada una ejecutando una instrucción diferente pero sobre el mismo flujo de datos. En este caso la salida de una unidad procesadora, es decir, los resultados del procesado, es la entrada de la siguiente, es decir, los operandos. No existe ninguna materialización real de esta categoría al menos que se clasifiquen específicamente en ella a los procesadores de segmentación encauzada.

Organización MIMD (figura 2.1d). En esta categoría se incluyen la mayoría de los sistemas multiprocesadores y multicomputadores. Un computador MIMD intrínsecamente hablando implica interacciones entre varios procesadores porque todos los flujos de memoria se derivan del mismo espacio de datos compartido por todos ellos. Si los n flujos de datos provinieran de subespacios disjuntos dentro de las memorias compartidas, entonces tendríamos un sistema SISD múltiple o MSISD, que no es más que un conjunto de sistemas monoprocesador SISD independientes.



(a) Sistema con memoria local



(b) Sistema con memoria global

FIGURA 2.2 Arquitecturas SIMD.

3. ARQUITECTURAS MIMD

De los cuatro tipos de arquitecturas descritas en la clasificación de Flynn, la que se ajusta más para la implementación de los sistemas de procesamiento de sensores en DSN es, sin duda, la arquitectura MIMD. En las DSN disponemos de una multitud de flujos de datos provenientes de los sensores que alimentan una serie de procesadores cada uno de los cuales puede estar ejecutando un procesamiento diferente sobre los datos. Así pues disponemos de múltiples flujos de datos y múltiples flujos de instrucciones.

Por esto, en este punto, vamos a estudiar un poco más en detalle las arquitecturas MIMD.

3.1 CLASIFICACIÓN [8]

Las arquitecturas MIMD pueden a su vez clasificarse en *multiprocesadores* o *multicomputadores* en función de la estructura de los procesadores y del mecanismo usado para la comunicación entre los mismos.

3.1.1 Sistemas multiprocesador

En los sistemas MIMD multiprocesador o con memoria compartida, la comunicación entre los distintos procesadores se realiza mediante la memoria común a la que se conectan los distintos procesadores a través de una red de interconexión. Este tipo de sistemas también es conocido como *fuertemente acoplado* pues, por lo general, existe un grado alto de interrelación entre los procesadores para llevar adelante la tarea propuesta.

Atendiendo a la organización de la memoria, los sistemas MIMD estrechamente acoplados se subdividen en tres modelos:

MODELO UMA (UNIFORM MEMORY ACCESS). Este modelo se caracteriza porque la memoria física se halla compartida uniformemente por todos los procesadores, aunque cada procesador pueda tener una caché privada. En este caso todos ellos tienen el mismo tiempo de acceso a todas las palabras de la memoria. El modelo UMA es adecuado para aplicaciones multiusuario de tiempo compartido de propósito general. La coordinación entre los procesadores se realiza a través de variables compartidas situadas en la memoria principal (figura 2.3). Ejemplo de este tipo de organización es el procesador S-81 de la firma Sequent Symmetry.

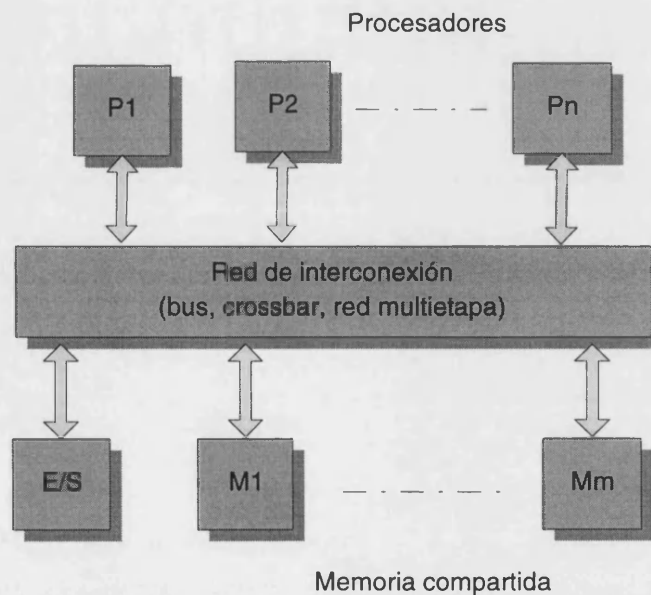
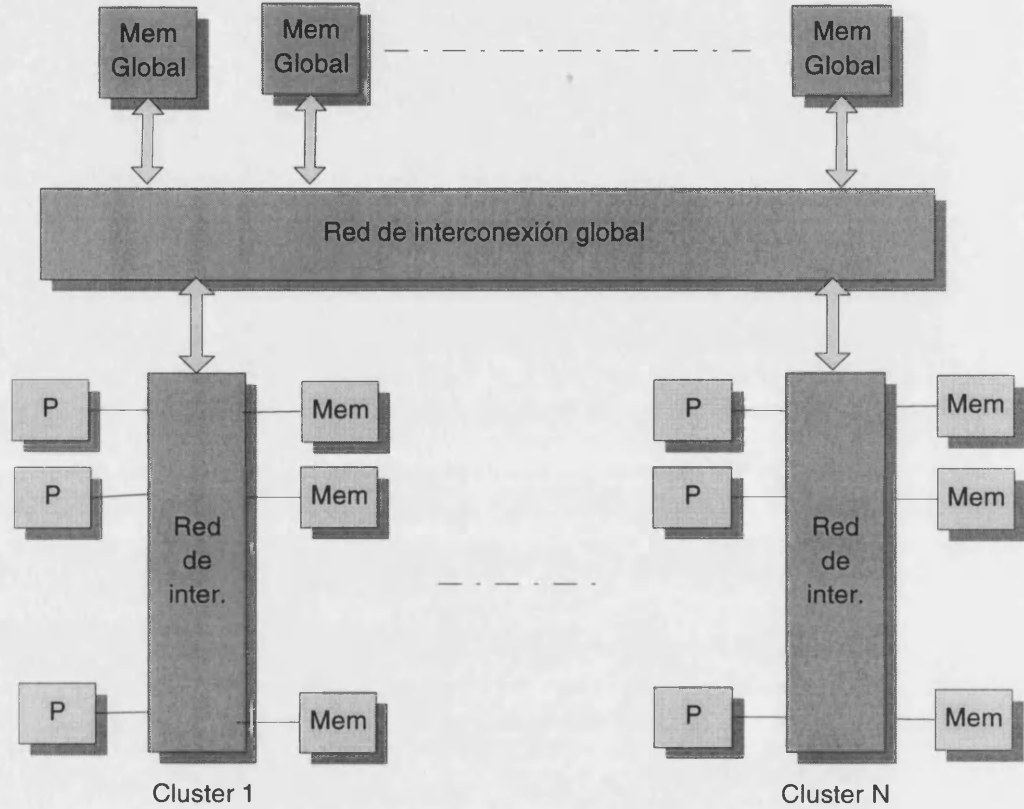
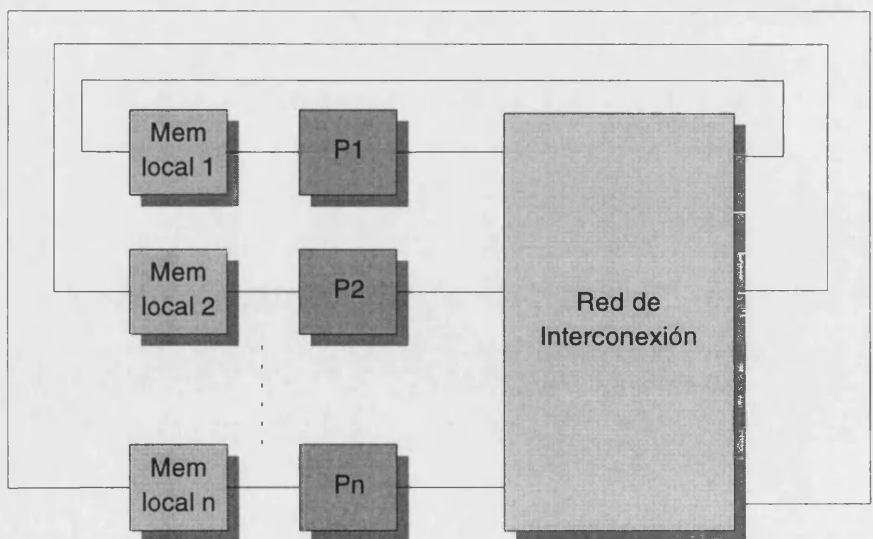


FIGURA 2.3 Modelo multiprocesador UMA.

MODELO NUMA (NON UNIFORM MEMORY ACCESS). En este modelo, el tiempo de acceso varía en función de la palabra a acceder. La figura 2.4 muestra los modelos de máquinas NUMA. La memoria compartida está físicamente distribuida entre todos los procesadores como memoria local. El conjunto de todas las memorias locales forma un espacio de direcciones global accesible por todos los procesadores.



a) Memorias locales y memoria global compartida.



b) Memorias locales compartidas.

FIGURA 2.4 Modelos NUMA para sistemas multiprocesador.

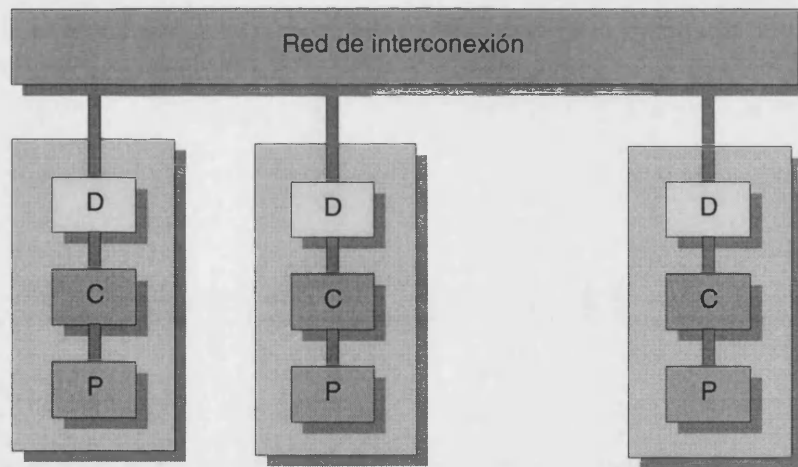
Como es evidente, resulta más rápido el acceso desde un procesador a la memoria local propia que a la de otro a través de la red de interconexión.

En este tipo de organización se puede incluir, además, memoria global accesible sólo a través de la red de interconexión (figura 2.4b), de manera que tenemos tres velocidades de acceso: a memoria local, global y remota en otro procesador, ordenadas de mayor a menor.

Como ejemplo de sistema NUMA con estructura como la mostrada en la figura 2.4a podemos citar el computador TC-2000 Butterfly de la firma BBN Advanced Computers Inc., mientras que el sistema Cedar de la Universidad de Illinois sigue la arquitectura de la figura 2.4b.

MODELO COMA (CACHE ONLY MEMORY ARCHITECTURE). Si las memorias locales compartidas del modelo NUMA se reemplazan por memorias caché en cada procesador obtenemos el modelo COMA. En este caso, el sistema de memoria solo tiene un nivel de jerarquía que consiste en las cachés distribuidas en los nodos procesadores. Los accesos a las cachés remotas se realizan a través de la red de interconexión como en el caso NUMA. A diferencia de éste, el modelo COMA permite que los datos almacenados cambien de caché en función del patrón de acceso a los mismos. Para lograr esto se emplean mecanismos de directorios distribuidos que sitúan los datos en la caché distribuida donde se solicitan (figura 2.5).

Ejemplos de máquinas COMA incluyen al Data Diffusion Machine (DDM) del Instituto Sueco de Ciencias de la Computación y el KSR-1 de la empresa Kendall Square Research Corporation.



D: Directorio; C: Cache; P: Procesador

FIGURA 2.5 Modelo COMA de un sistema multiprocesador.

3.1.2 Sistemas multicomputador

Los sistemas MIMD por intercambio de mensajes se conocen también con el nombre de multicomputadores, sistemas MIMD ligeramente acoplados o sistemas distribuidos (figura 2.6).

Cada nodo procesador dispone de un procesador, memoria y la interface correspondiente para el acceso a la red de interconexión. La entrada/salida de datos del sistema multicomputador puede estar disponible en algunos o en todos los nodos que lo componen.

A diferencia de los multiprocesadores, los procesadores del sistema multicomputador no tienen acceso directo a la memoria de cualquiera de los otros procesadores sino que toda la coordinación y la compartición de datos se realizan a través de intercambios de mensajes. La misión de la red de interconexión es facilitar este paso de mensajes entre los nodos procesador.

La red de interconexión en los sistemas multicomputador son redes estáticas punto a punto con topologías típicas de anillos, toros, hipercubos, etc.

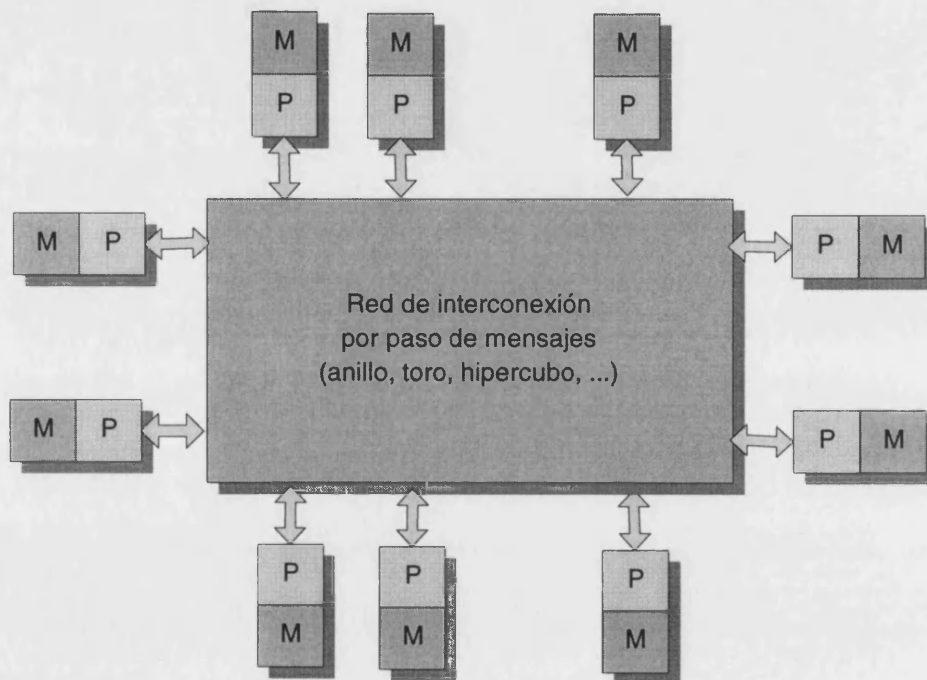


FIGURA 2.6 Modelo genérico de multicomputador.

3.2 PARÁMETROS DE EXCELENCIA DE LOS SISTEMAS MIMD

Para evaluar la bondad de los sistemas multiprocesador, se recurre a una serie de parámetros que miden la mejora producida con respecto al empleo de un único procesador. En este apartado vamos a analizar algunos de estos parámetros.

3.2.1 Escalabilidad [1] [9] [10]

La definición más simple de escalabilidad es que un sistema es escalable si el rendimiento del mismo se incrementa linealmente con relación al número de procesadores usados para una cierta aplicación. El rendimiento de un sistema procesador depende de un gran número de factores que influyen en la escalabilidad de la arquitectura del sistema y el programa de aplicación que se ejecute.

El análisis de la escalabilidad de un sistema debe realizarse para una cierta aplicación y bajo diferentes restricciones en el crecimiento del tamaño del problema (carga de trabajo) y el tamaño del sistema (número de procesadores).

Los estudios de escalabilidad determinan el grado de afinidad entre una arquitectura determinada y una aplicación. Para diferentes pares (arquitectura, algoritmo), el análisis puede obtener diferentes resultados. Una máquina puede ser muy eficiente para un algoritmo y muy poco para otro.

Los principales parámetros que afectan la escalabilidad de un sistema para una determinada aplicación son:

- **Tamaño del sistema:** el número de procesadores utilizados en el sistema. Un tamaño grande implica más recursos y más potencia de procesado.
- **Frecuencia de reloj:** la frecuencia de reloj determina el ciclo máquina básico. Se deseará un sistema cuyos componentes (procesadores, memorias, buses, etc.) estén controlados por un reloj que pueda incrementarse cuando la tecnología mejore.
- **Tamaño del problema:** la cantidad de trabajo computacional necesaria para resolver un determinado problema.
- **Tiempo de CPU:** el tiempo de CPU real usado en la ejecución de un determinado programa en un sistema con n procesadores.
- **Requerimientos de E/S:** los requerimientos de E/S necesarios para la transferencia de datos asociados con la aplicación.
- **Capacidad de memoria:** la cantidad de memoria principal usada para la ejecución del programa. Hay que tener en cuenta que la demanda de memoria viene afectada por el tamaño del problema, el tamaño del programa, los algoritmos y las estructuras de datos usadas. Puesto que la demanda de memoria puede variar en tiempo de ejecución, este parámetro se refiere a la máxima cantidad de memoria solicitada.
- **Pérdidas (overhead) de comunicación:** la cantidad de tiempo gastada en la comunicación entre procesadores, sincronización, accesos remotos, etc. Este overhead incluye todas las operaciones que no involucran a la CPU o a los dispositivos de E/S.
- **Coste del sistema:** el coste total de los recursos hardware y software necesarios para llevar a cabo la ejecución de un programa.

En función de los objetivos establecidos y las restricciones de recursos impuestas, se pueden fijar algunos de los parámetros anteriores y optimizar los restantes para conseguir el mayor rendimiento con el menor coste.

La noción de escalabilidad está ligada a las nociones de incremento de velocidad e eficiencia. Una buena expresión de la escalabilidad deber ser capaz de incluir los efectos de la red de interconexión de la arquitectura, de los patrones de comunicación inherentes a los algoritmos empleados, de las restricciones físicas impuestas por la tecnología y de la eficiencia del sistema. Veamos a continuación esos dos parámetros: incremento de velocidad y eficiencia.

3.2.2 Incremento de velocidad y eficiencia de un sistema multiprocesador

El incremento de velocidad, también conocido como ganancia de velocidad o *speedup*, S , de un sistema con un número $n > 1$ de procesadores es como máximo n veces superior a la de un procesador único. En la práctica, sin embargo, la ganancia es mucho menor puesto que algunos procesadores permanecen inactivos en algunos instantes debido a conflictos en los accesos a memoria o en las comunicaciones o al uso de algoritmos ineficaces.

Minsky y Papert [11] demostraron que un límite inferior típico para este incremento de velocidad venia dado por $S = \log_2 n$; sin embargo, otros estudios [12] han demostrado límites inferiores típicos mayores que el de Minsky.

Hwang y Briggs [10] demostraron que para un sistema multiprocesador en el que la carga se distribuye de manera equitativa y equiprobable, una cota superior del incremento de velocidad viene dada por la expresión:

$$S \leq \frac{n}{\ln n} \quad (2.1)$$

La figura 2.7 muestra la comparación entre el caso de Minsky, Hwang y el ideal. Como se observa, las curvas se aproximan al caso ideal para n pequeñas pero divergen para n grandes. La aproximación de Hwang se ajusta más al caso que la de Minsky, sobretodo para valores de n grandes.

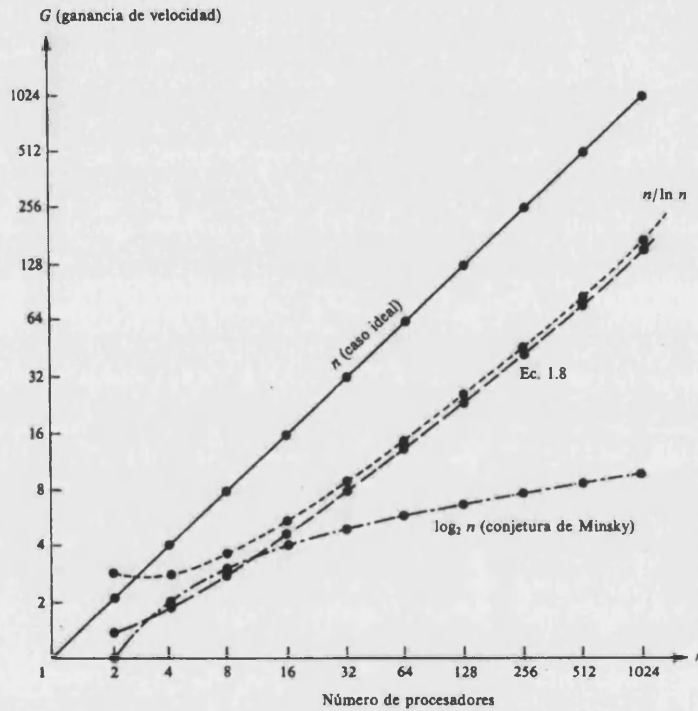


FIGURA 2.7 Ganancia de velocidad de un sistema de n procesadores frente a un procesador único.

Para el caso más general, la ecuación dada por Hwang en [9] proporciona el incremento de velocidad del sistema multiprocesador con respecto a un monoprocesador incluyendo la posibilidad de secciones secuenciales y la carga impuesta por las comunicaciones entre los procesadores. La ecuación mencionada es la siguiente:

$$S(n) = \frac{\sum_{i=1}^m W_i}{\sum_{i=1}^m \frac{W_i}{i} \left\lceil \frac{i}{n} \right\rceil + Q(n)} \quad (2.2)$$

donde W_i es el trabajo que se realiza cuando se usan i procesadores de los n existentes en el sistema, m es el número máximo de procesadores necesarios para ejecutar la tarea o, lo que es lo mismo, el grado máximo de paralelismo, $Q(n)$ es la carga introducida por las comunicaciones y $\lceil x \rceil$ representa la operación que devuelve el menor entero mayor o igual que el número real positivo x .

En el caso especial en que $Q(n) = 0$ y el sistema funciona en modo secuencial o en modo totalmente paralelo usando los n procesadores, donde n es ilimitado, es decir $n \gg m$, la expresión del incremento de velocidad se traduce en:

$$S(n) = \frac{n}{1 + (n-1)f} \quad (2.3)$$

donde f es la fracción de trabajo que no puede ser ejecutada concurrentemente. Esta expresión conocida con la *ley de Amdhal* [13] nos proporciona un límite en el incremento de velocidad al aumentar el número de procesadores,

$$\lim_{n \rightarrow \infty} S(n) = \frac{1}{f} \quad (2.4)$$

es decir, no podemos incrementar la velocidad más de $1/f$, lo que implica que el sistema viene limitado por cuellos de botella secuenciales.

Sin embargo, en otro tipo de aplicaciones, el aumento del número de procesadores se emplea, por ejemplo, en una determinación más exacta de la solución del problema manteniendo el tiempo de procesado constante y no en una resolución más rápida. Un ejemplo de este tipo de aplicación es el tratamiento de imágenes donde un aumento del número de procesadores puede conducirnos a obtener un grado de resolución mayor en el procesado de ésta.

En estos casos, la ecuación de Amdhal no es aplicable porque en su deducción se supone un sistema en el que la carga de trabajo es constante y lo que se consigue al aumentar el número de procesadores es disminuir el tiempo que se tarda en procesar esa carga. En los casos en los que el interés no es reducir el tiempo de ejecución sino aumentar la carga del sistema manteniendo el tiempo de ejecución constante, Guftavson [14] ha deducido una expresión para el incremento de velocidad:

$$S(n) = n - \alpha(n-1) \quad (2.5)$$

donde α es la fracción de tiempo empleado en el procesado serie.

La figura 2.8 muestra la comparación entre la expresión de Amdhal y la de Guftavson.

Como se observa, la expresión de Guftavson no tiene la limitación de la de Amdhal y refleja mejor casos reales como la predicción climática, el cálculo de estructuras o la adquisición de datos en experimentos de Altas Energías como veremos más tarde, donde un aumento del número de procesadores permite una mejor exactitud en los cálculos o el procesado de un número mayor de datos.

Para cualquiera de las definiciones de *speedup* anteriores, la eficiencia de un sistema se define mediante la siguiente ecuación [9]:

$$E(n) = \frac{S(n)}{n} \quad (2.6)$$

Evidentemente, la mejor eficiencia posible es uno, lo que implica que el *speedup* se incrementa de igual manera que el número de procesadores del sistema, es decir, que $S(n) = n$.

Podemos ahora redefinir la escalabilidad diciendo que un sistema es escalable si su eficiencia $E(n) = 1$ para todos los algoritmos con cualquier número n de procesadores y para cualquier tamaño de problema.

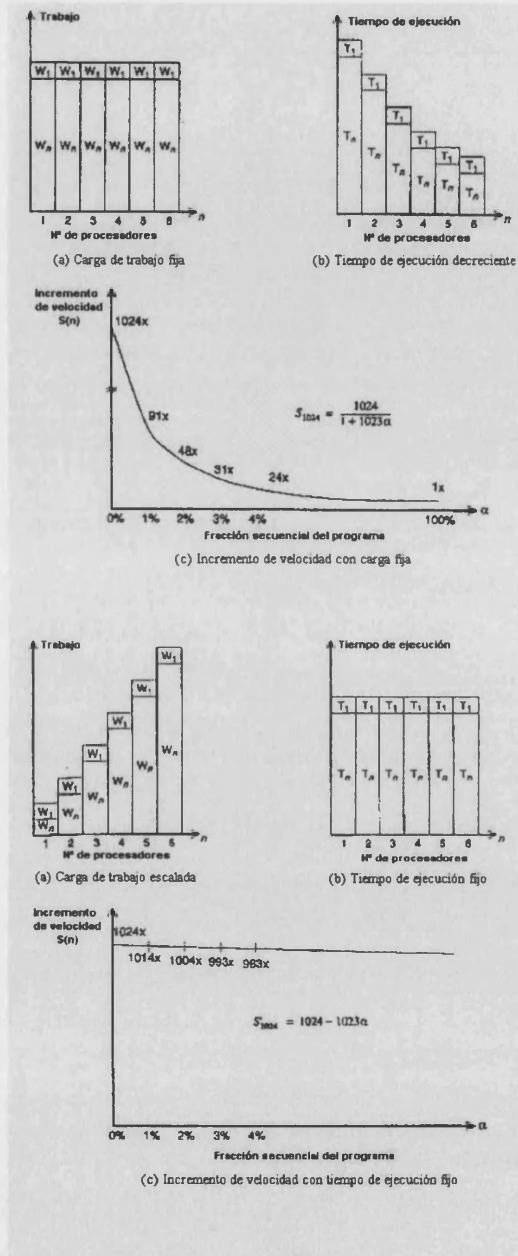


FIGURA 2.8 Comparación entre las expresiones de Amdahl (parte superior) y Gustafson (parte inferior) para el incremento de velocidad.

4. REDES DE INTERCONEXIÓN EN SISTEMAS MIMD

Las redes de interconexión son vitales en cualquier sistema multiprocesador puesto que para el buen funcionamiento del mismo, es necesario que exista una coordinación entre los procesadores. Esta comunicación, establecida a través de la red de interconexión, sin embargo, debe suponer el mínimo tiempo posible en el conjunto general del procesado. Por lo tanto, el diseño de la red de interconexión, tanto en su

topología como en su implementación, es uno de los aspectos más importantes en los sistemas multiprocesador.

En general, todas estas redes se pueden dividir en dos grandes grupos: redes estáticas y redes dinámicas. Las redes estáticas o punto a punto son aquellas que interconectan dos procesadores de manera que se establece entre ellos un canal de comunicación permanente, de forma que la topología de la red queda fijada. En las redes dinámicas, el patrón de interconexión entre los nodos de la red puede variar dinámicamente mediante el uso de elementos de conmutación.

Las redes de interconexión empleadas en los sistemas MIMD difieren según se trate de un sistema multiprocesador o un sistema multicomputador. En el primer caso, las redes suelen ser estáticas mientras que en el segundo se prefieren redes de interconexión dinámicas. A continuación veremos los distintos tipos de redes de interconexión para cada uno de los tipos de sistema MIMD.

4.1 SISTEMAS MULTIPROCESADOR

Los mecanismos de interconexión entre procesadores en un sistema MIMD multiprocesador pueden clasificarse en cuatro grupos (Enslow, 1977, [15]):

BUSES COMUNES O DE TIEMPO COMPARTIDO. Este es el esquema más simple (figura 2.9) en el que únicamente existe un camino de comunicación común que conecta todas las unidades funcionales. Puesto que el bus es compartido, se debe disponer de mecanismos para la resolución de conflictos de acceso al mismo. Ejemplos de este tipo de red de interconexión son el bus VME [16] utilizado ampliamente en aplicaciones industriales o el bus Futurebus+ [17].

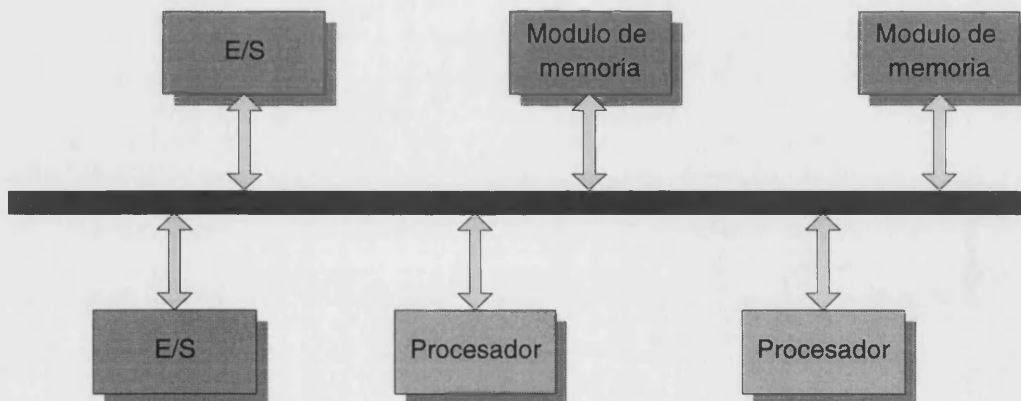


FIGURA 2.9 Organización multiprocesador de bus común.

REDES CROSSBAR. Si multiplicamos el número de buses, se llega a un punto en el que existe un camino disponible de cada procesador a cada memoria (figura 2.10). Este tipo de organización se denomina de *barras cruzadas sin bloqueo* pues existe un bus separado asociado con cada módulo de memoria. El número máximo de transferencias que se pueden realizar está limitado más por el número de módulos de memoria y el producto ancho de banda · velocidad de los buses, que por el número de caminos disponibles. Ejemplos de computadores que emplean este tipo de red de interconexión son el VPP500 de Fujitsu [18] que dispone de una red crossbar de 224 x 224 o el Cray Y-MP/816 [19] que emplea una serie de redes crossbar de 4 x 4 y 8 x 8 en sucesivas etapas de conmutación.

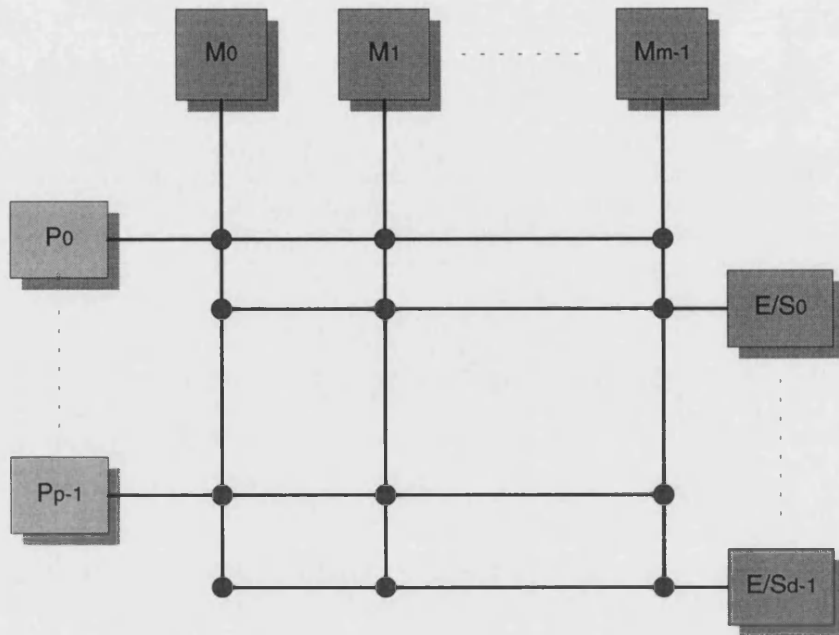


FIGURA 2.10 Organización de multiprocesador con crossbar.

MEMORIAS MULTIPUERTO. En este caso la lógica de control, conmutación y arbitraje distribuido en la red crossbar, se distribuye en las interfaces de los módulos de memoria, dando como resultado una memoria con varios puertos de lectura/escritura. Si, por ejemplo, tenemos 16 procesadores, la memoria tendría 16 puertos, uno para cada procesador (figura 2.11).

Aunque es factible el diseño de memorias con un número elevado de puertos, éste es muy complejo y caro, por lo que a menudo se usan memorias pseudomultipuerto en las que los accesos son secuenciales pero a alta velocidad. Como ejemplo de sistema que emplea este tipo de interconexión está el Univac 1100/94 [20], constituido internamente por cuatro CPUs, cuatro procesadores de E/S y dos procesadores vectoriales, utilizando cuatro memorias de 10 puertos para dar servicio a los procesadores.

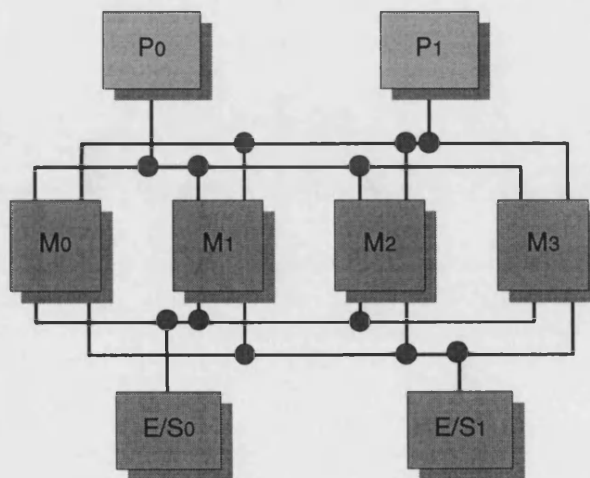


FIGURA 2.11 Sistema multiprocesador mediante memoria multipuerto.

REDES MULTIETAPA. El coste y la complejidad de la red crossbar crecen como el producto de las fuentes por destinos en la red. Si tenemos N procesadores y M módulos de memoria, la red crossbar tendrá $N \times M$ puntos de cruce. En cada uno de ellos existirán $N \times M \times W$ entradas/salidas, donde W es el ancho de la palabra.

Si tenemos un número grande de procesadores y memorias o de ambas, la red crossbar puede ser muy compleja de construir. Para este tipo de casos existe un método para realizar la conexión usando sucesivas etapas de conmutadores como se muestra en la figura 2.12, logrando abaratar el coste y disminuir la complejidad de la red de interconexión a cambio de reducir también las prestaciones, puesto que en este caso se pueden producir colisiones en los puntos de interconexión.

La barajadura- a (a -shuffle) de $n = a \cdot c$ objetos es una permutación de $a \cdot c$ índices (0, 1, ..., $a \cdot c - 1$) definida como:

$$S_{a \cdot c}(i) = \left(a \cdot i + \left\lfloor \frac{i}{c} \right\rfloor \right) \bmod (a \cdot c) \quad (2.7)$$

Ejemplos de sistemas con red de interconexión multietapa son el IBM RP3 [21] o el NYU Ultracomputer [22].

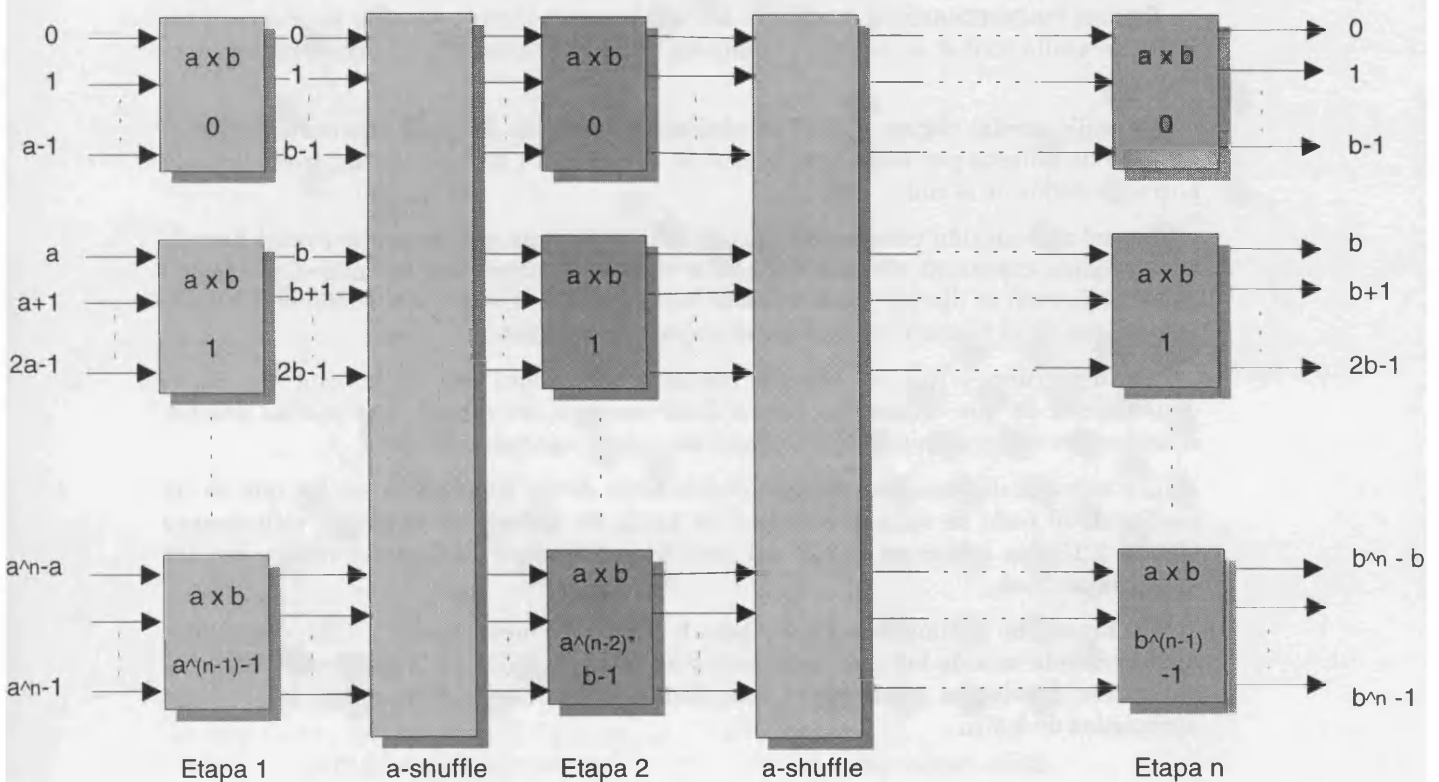


FIGURA 2.12 Red multietapa con a^n entradas y b^n salidas.

4.2 SISTEMAS MULTICOMP UTADOR

Para los sistemas MIMD multicomputador, las redes de interconexión se clasifican según las dimensiones que requiera su disposición estructural. Así tendremos redes unidimensionales, bidimensionales, tridimensionales e hipercubos. La figura 2.13 muestra ejemplos de redes de interconexión para cada uno de los tipos descritos.

REDES UNIDIMENSIONALES. Como ejemplo de red unidimensional tenemos la red lineal (figura 2.13a) empleada en muy pocas ocasiones. En esta red, N nodos procesadores se conectan unos con otros mediante $N-1$ enlaces como si se dispusiesen en una línea.

Esta topología es la más sencilla y presenta problemas de comunicación cuando N se hace grande. Para valores de N pequeños es una red bastante sencilla de implementar. Se diferencia de la interconexión en bus común en el hecho de que en un momento dato puede realizarse más de una transferencia simultáneamente siempre que sea a través de enlaces diferentes.

REDES BIDIMENSIONALES. Entre las redes bidimensionales podemos citar el anillo, la estrella, el árbol, la malla y la red sistólica. Este tipo de organización es usado con mucha frecuencia en la interconexión de multicomputadores. De todas ellas, quizá sean el anillo, el árbol y la malla las más usadas (figura 2.13b, c, d, e, f).

Un anillo se consigue sin más que conectar los dos extremos de una red lineal. Los anillos pueden ser uni o bidireccionales en función de que la información dentro del anillo pueda fluir en uno o ambos sentidos.

REDES TRIDIMENSIONALES. Entre las topologías tridimensionales se incluyen las redes de anillo cordal, de conexión completa, cubos- n k -arios, hipercubos y cubos ciclo-conexos.

El anillo cordal (figura 2.13g) se obtiene a partir de un anillo incrementando el número de enlaces por nodo, con lo cual se disminuyen los tiempos de transferencia entre los nodos de la red.

La red de conexión completa (figura 2.13h) no es más que un anillo cordal llevado a su máxima expresión. En esta red existe un camino entre cada dos nodos, por lo que la comunicación es directa, reduciéndose los tiempos de transferencia. Sin embargo al incrementarse el número de enlaces, se incrementa el coste de la red.

Un hipercubo o cubo- n binario contiene dos nodos por cada una de las n dimensiones de que consta. La figura 2.13i muestra un cubo-3. Las mallas pueden considerarse como un hipercubos bidimensionales, respectivamente.

Los cubos ciclo-conexos son una modificación de los hipercubos en los que se ha sustituido el nodo de cada vértice por un anillo de nodos. Así el cubo-3 ciclo-conexo (figura 2.13j) se obtiene a partir del cubo-3 sustituyendo cada nodo vértice por un anillo de 3 nodos.

Un hipercubo n -dimensional de ancho k o cubo- n k -ario (figura 2.13k), contiene k nodos en cada una de las n dimensiones y en total k^n nodos en toda la red. Todas las anteriores topologías pueden ser asimiladas a un cubo- n k -ario con los valores apropiados de k y n .

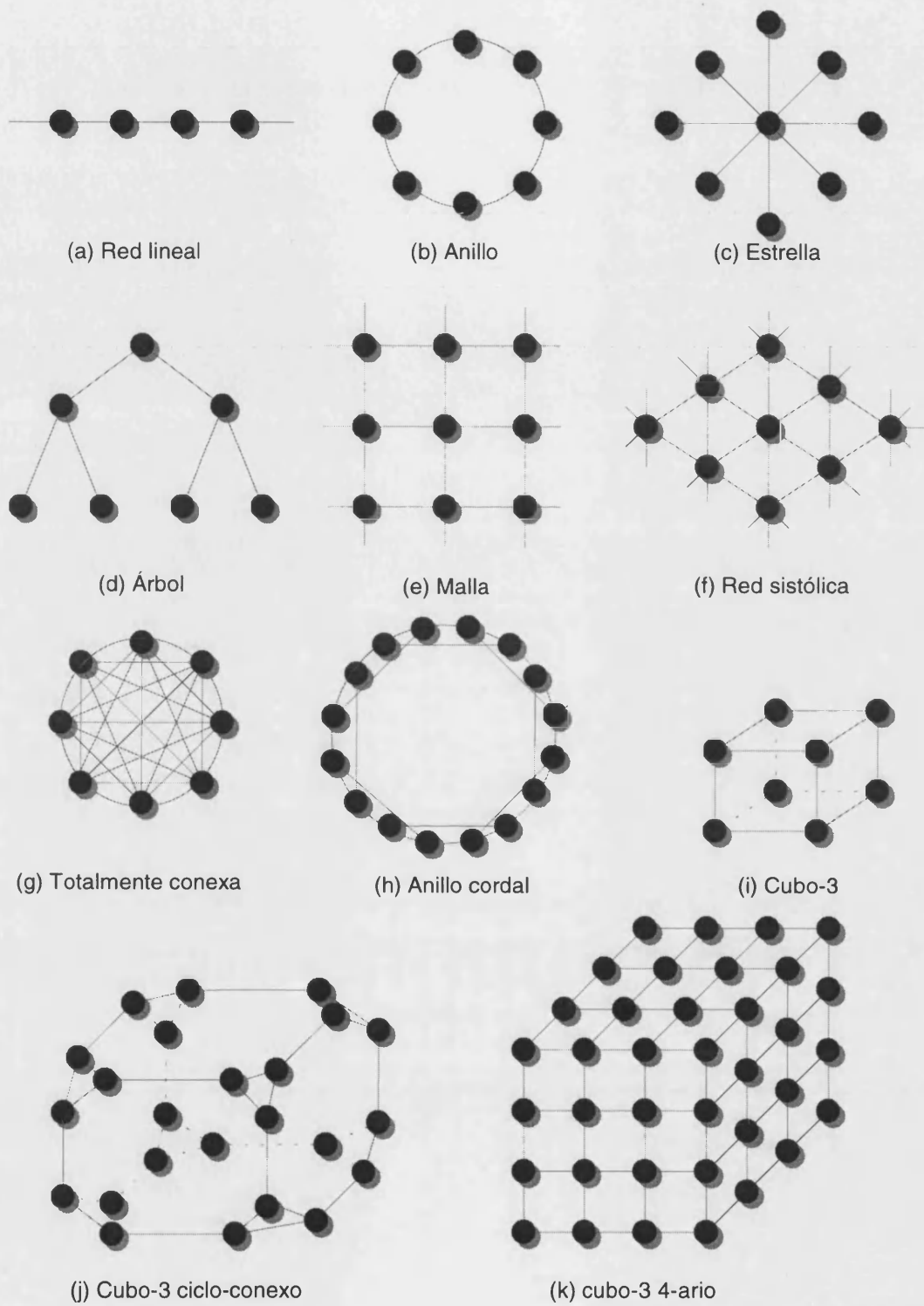


FIGURA 2.13 Topologías de redes de interconexión estáticas.

4.3 TECNOLOGÍAS DE REDES DE INTERCONEXIÓN

Vamos a ver en este apartado, tres ejemplos de tecnologías de redes de interconexión actuales de altas prestaciones. Cada una de ellas se adapta mejor a un tipo de aplicación concreto dentro del campo de los sistemas MIMD.

4.3.1 Scalable Coherent Interface (SCI)

El estándar Scalable Coherent Interface o SCI, [23] fue desarrollado por un grupo de ingenieros que trabajaban con sistemas de bus de altas prestaciones que habían llegado a entender y sufrir las limitaciones de esta tecnología.

La razón ya la había apuntado en 1987 Paul Sweazey, líder del grupo de Coherencia de caché de Futurebus+ [17], al darse cuenta que por aquel entonces la tendencia era que la velocidad de los procesadores pronto superaría la capacidad de cualquier bus para poder realizar multiproceso de manera eficaz. Con esta limitación en mente, en 1988, el IEEE organizó un grupo de trabajo liderado por David Gustavson para intentar evitar lo que parecía una catástrofe.

Inicialmente el grupo de trabajo se centró en intentar superar las limitaciones del bus, lo que, poco a poco, condujo a la necesidad de olvidar esta arquitectura y pasar a considerar otras diferentes.

Para la elección de la arquitectura se tuvo presente que se deseaba obtener una solución que mejorara las prestaciones del sistema de bus común sin complicar excesivamente el diseño, tanto al nivel de la conexión hardware como de protocolos [24].

En primer lugar se olvidó la idea de un bus común y se pasó a una arquitectura basada en enlaces punto a punto, que facilitaban el incremento de velocidad en la transmisión de manera significativa. Además la información fluiría siempre en el mismo sentido, es decir, los enlaces serían unidireccionales.

Se pensó en reducir el margen de las señales para aumentar la velocidad, así como emplear señalización diferencial para minimizar los problemas de ruido eléctrico.

Ya que el medio de transmisión (cable coaxial, fibra óptica, etc.) condiciona la velocidad del canal de comunicación, el nuevo estándar debería optimizar el ancho de banda para cualquiera de las tecnologías empleadas en la comunicación.

Puesto que se deseaban altas prestaciones se deseaba minimizar la latencia en la comunicación, lo que implicaba emplear paquetes de información pequeños.

Además el sistema debería ser fácilmente escalable para permitir su crecimiento sin por ello disminuir en sus prestaciones sino, al contrario, incrementarlas de forma lineal.

Cada uno de los nodos debería usar su propio reloj para la transmisión de los datos porque la distribución de un reloj central a todos los nodos es un mecanismo caro y que no se escala bien. Los datos que le llegan a cada nodo contienen el reloj al cual se sincroniza el nodo receptor para poder descodificar adecuadamente el paquete.

Con estas consideraciones, en enero de 1991, las especificaciones finales del nuevo estándar SCI estaban ya preparadas para su aprobación por el IEEE, en marzo de 1992 y por ANSI en octubre de ese mismo año.

4.3.1.1 Características

El estándar SCI se define a grandes rasgos por sus tres siglas: S de escalabilidad, C de coherencia e I de interface.

La escalabilidad, como ya se ha apuntado, se refiere a que el sistema puede emplear los mismos mecanismos (hardware y software) tanto para sistemas grandes como pequeños, sin por ello sufrir degradaciones en sus prestaciones.

En los sistemas multiprocesadores con memoria compartida, los procesadores pueden disponer de memoria caché para almacenar los valores más accedidos minimizando los tiempos de accesos. SCI soporta mecanismo de coherencia de caché para asegurar que los datos de las caché se actualizan e invalidan de forma adecuada.

Por último, la interface hace mención a la voluntad de que la arquitectura sea abierta de forma que puedan emplearse en un mismo sistema productos de fabricantes diversos.

De forma resumida, las características de este estándar son las siguientes [24][25]:

- Enlaces punto a punto unidireccionales.
- Protocolo de transferencia síncrono y transferencia en ambos flancos del reloj.
- Velocidad de 1Gbyte/s para distancias cortas (metros) y cable triaxial.
- Velocidad de 1250 Mbits/s para distancias largas (kilómetros) y fibra óptica.
- Configuración de memoria NUMA. Memoria distribuida y compartida.
- Coherencia de caché basada en directorios distribuidos.
- Posibilidad de empleo de paso de mensajes.
- Escalable de 1 a 64000 nodos.
- Arquitectura CSR (Control State Registers [26]).
- Posibilidad de diversas configuraciones de la red de interconexión: anillos, malla, hipercubo.

4.3.1.2 Protocolo de transmisión

El protocolo en que se basa SCI es del tipo SEND-ECHO, en el cual cada paquete que envía un emisor hacia el receptor produce en éste una respuesta hacia el primero. De esta forma se realiza el control de errores en la recepción o transmisión de los paquetes. La conexión a 1 Gbyte/s emplea 18 líneas de señal divididas en 16 líneas de datos, 1 señal de reloj de 250 MHz y una línea de flag que indica el comienzo del paquete.

Toda transferencia consta de dos subacciones: petición y respuesta, que a su vez constan de sendos pares SEND-ECHO. Así pues, en cada transacción se realizan cuatro transferencias de datos (figura 2.14).

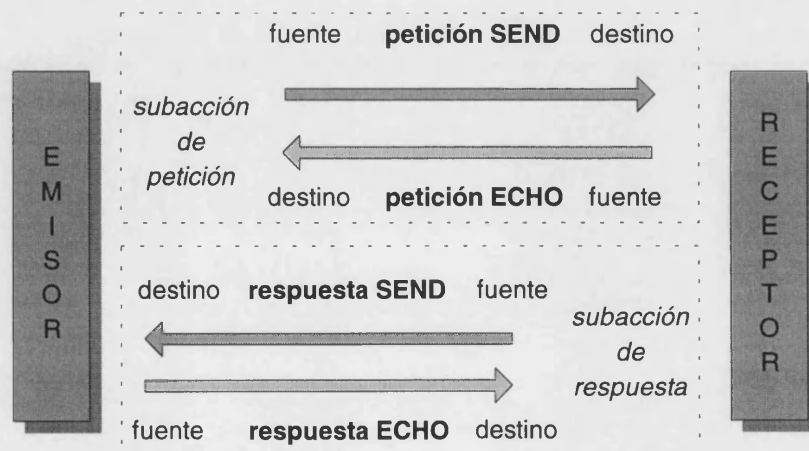


FIGURA 2.14 Fases de una transacción en SCI.

Los paquetes (figura 2.15) constan de una cabecera de 16 bytes que contiene la información relativa al destino y al comando, un campo de datos variable de 0 a 256 bytes (no presente en algunos comandos) y un código CRC para el control de errores. En la figura, **xx** indica uno de los tamaños de bloque de datos permitidos.

Operación	Petición	Respuesta
readxx*	cabecera	cabecera 0, 16, 64, 256
writexx*	cabecera 16, 64, 256	cabecera
movexx*	cabecera 0, 16, 64, 256	
eventxx*	cabecera 0, 16, 64, 256	
locks b	cabecera 16	cabecera 16

FIGURA 2.15 Formato de los paquetes en SCI.

4.3.2 Topologías interconexión

Para asegurar el éxito de SCI, se planteó la necesidad de que el sistema fuera interconectable con sistemas actualmente en uso de manera sencilla.

La topología más sencilla implementable mediante SCI es la de anillo (figura 2.16) donde se pueden conectar los puentes necesarios para la interconexión con sistemas de bus común actualmente existentes VME o Futurebus+.

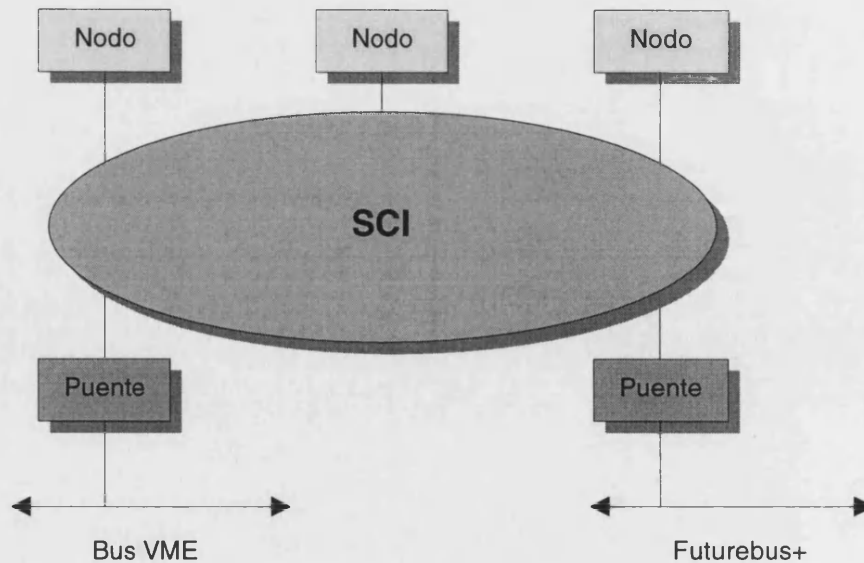


FIGURA 2.16 Configuración en anillo de SCI con interconexión a otros sistemas de bus común.

No obstante, mediante SCI es posible también configurar redes más complejas como mallas o hipercubos mediante el empleo de puentes entre anillos SCI (figura 2.17).

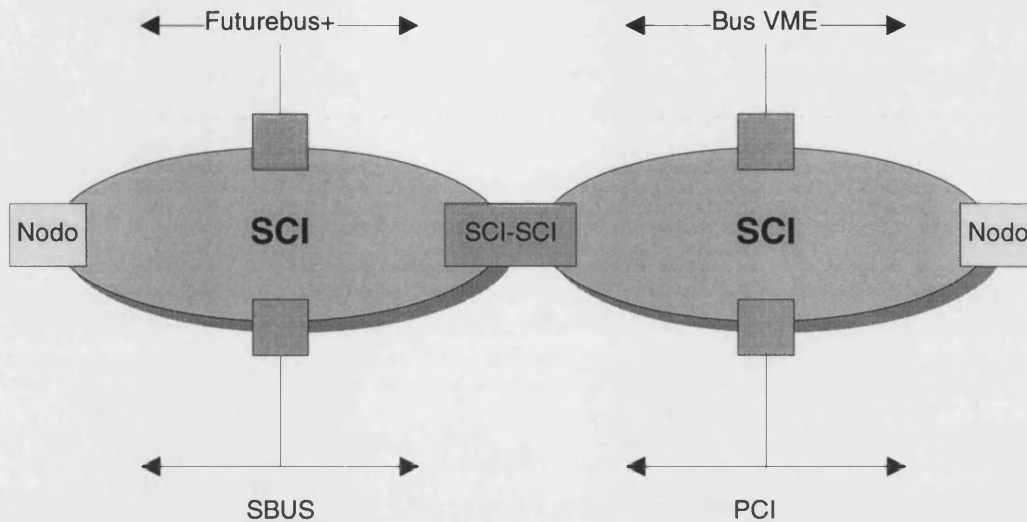


FIGURA 2.17 Interconexión de dos anillos SCI mediante un puente.

4.3.2.1 Integrados para el estándar SCI

Actualmente están disponibles comercialmente integrados VLSI, fabricados por la empresa noruega Dolphin, denominados *NodeChips*, que permiten la construcción de sistemas basados en SCI. Estos integrados disponen de un bus local o de aplicaciones, al cual se conectan los dispositivos externos específicos, y, por otra parte, de las líneas necesarias para la conexión directa del cable de interconexión.

Cada nodo SCI consta, a parte de los dos buses mencionados, de dos colas de datos, una de entrada y otra de salida, y de una FIFO de bypass, como se muestra en la figura 2.18. Las colas de entrada y salida almacenan los datos que llegan o salen respectivamente del nodo, mientras que la FIFO de bypass se encarga de los paquetes que están de tránsito hacia otro nodo.

La primera generación de *NodeChips* apareció en abril de 1993 y estaban realizados en tecnología de AsGa. Operaban inicialmente a una velocidad de 500 Mbytes/s en el anillo y 250 Mbytes/s en el bus de aplicaciones de 64 bits, denominado CBUS.

La segunda generación, basada en tecnología CMOS para facilitar su comercialización y abaratar costes, se presentó en marzo del 1994. Al ser CMOS, era más accesible para aplicaciones comerciales aún a costa de bajar las prestaciones del integrado. Mediante esta versión, se podía transmitir información a través del anillo a 125 Mbytes/s.

En 1995 apareció una nueva generación de interfaces SCI desarrollados también por Dolphin y bautizados como *LinkControllers*. Las diferencias con los *NodeChips* de la segunda generación están en el incremento de la velocidad de transferencia en el anillo, ahora de 200 Mbytes/s, en la sustitución del bus CBUS por otro nuevo, denominado B-LINK [27], y diseñado a tal efecto para mejorar las limitaciones del primero y en la mejora de la gestión y control del flujo de datos tanto en el anillo como en la aplicación.

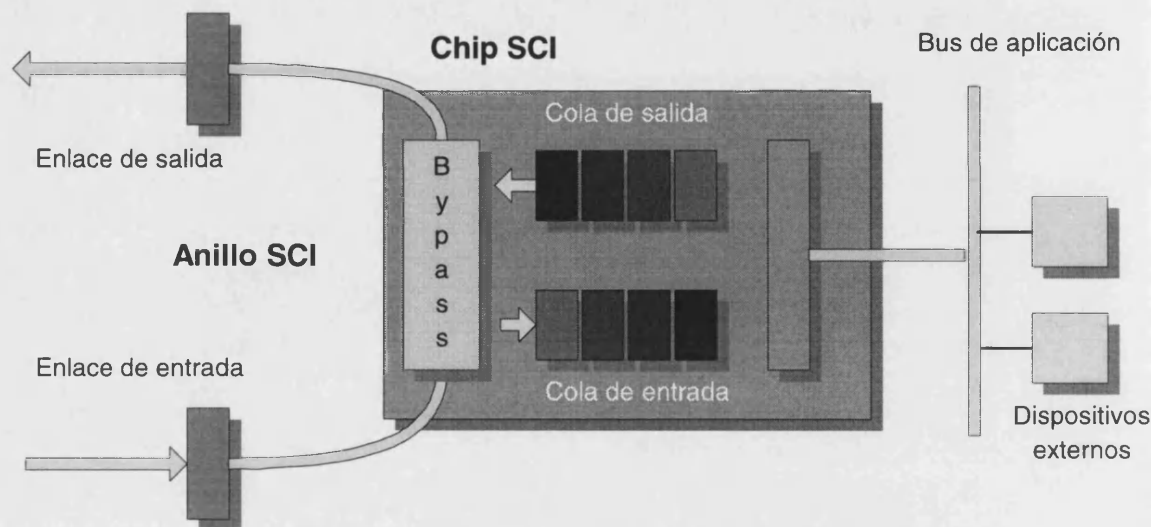


FIGURA 2.18 Estructura de un integrado interface SCI.

4.3.3 Asynchronous Transfer Mode (ATM)

En el último decenio, las necesidades de anchos de banda cada vez mayores y de diferentes tipos de datos (voz, vídeo, datos) para la transmisión de la información ha hecho que se creen redes especializadas para cada uno de esos tipos de información.

Por ello, en 1984 el CCITT publica las primeras recomendaciones para la realización de una Red Digital de Servicios Integrados (RDSI), que define como "una red que permite la conexión digital punto a punto y soporta una gran variedad de servicios, incluyendo voz y datos".

La RDSI original está basada sobre la red digital de teléfono, caracterizada por canales de 64 Kbits/s, con el inconveniente de no poder ofrecer más que este tipo de canal de transmisión. Sin embargo, basándose en este canal, RDSI ofrece dos tipos diferentes: *Acceso básico*, para anchos de banda pequeños, y *Acceso primario*, para anchos de banda mayores (1.5-2 Mbits/s).

Sin embargo, incluso los mayores anchos de banda ofrecidos por RDSI se quedan cortos ante la necesidad de enviar, por ejemplo, imágenes en movimiento con buena resolución. Por ello, se definió una Red Digital de Servicios Integrados de Banda Ancha o B-ISDN.

La RDSI de Banda Ancha incluye los canales de la RDSI básica y además permite la utilización de canales con anchos de banda desde 50 Mbits/s a cientos de Mbits/s. La información sobre esta red se envía en paquetes de tamaño pequeño y fijo, llamados células. Para la implementación de esta red se tuvo que desarrollar un protocolo de transferencia al que se denominó ATM (Modo de transferencia asíncrono) [28].

ATM se desarrolló como un sistema que pretendía dar la máxima flexibilidad al sistema de comunicaciones que implementase. Aunque en principio fue diseñado para la implementación de la RDSI de Banda Ancha, es posible transportarlo y aplicarlo en redes de interconexión o de comunicación entre sistemas.

Por el momento, ATM puede clasificarse como un sistema de comunicación para redes de computadores orientadas a aplicaciones multimedia, aunque existen trabajos en los que ya se estudia la forma de emplearlo para la realización de sistemas

multiprocesador o multicomputador [29], con las modificaciones necesarias para su adaptación.

4.3.3.1 Aspectos y conceptos básicos de ATM

ATM es una técnica de conmutación, de multiplexaje, incluso de transmisión, derivada de la conmutación de paquetes en cuanto que recurre a paquetes cortos y de tamaño fijo. El tratamiento de estos paquetes o células en los conmutadores está limitado al análisis de su cabecera para permitir el encaminamiento.

Las funciones de control del flujo o de tratamiento de errores no se efectúan en la red ATM, sino que se dejan a cargo de las aplicaciones usuarias o de los equipos de acceso. Estas características permiten a ATM responder razonablemente a las exigencias de tráficos tan diferentes como la voz, las imágenes animadas o los datos.

La tecnología ATM está basada en dos potentes pero flexibles conceptos:

- Cuando se necesita realizar una comunicación, el remitente negocia unas características de la conexión con la red, punto a punto. Cuando se configura la conexión, el remitente especifica el tipo, velocidad y otros atributos de la llamada, los cuales determinan la calidad del servicio desde el inicio hasta el final de la conexión. Es como si eligiésemos "en qué clase" queremos que viaje nuestra información.
- ATM es una tecnología basada en conmutación. La comunicación entre dos puntos se realiza a través de la conexión entre sucesivos conmutadores que van encaminando los datos de manera adecuada.

ATM segmenta la información a enviar en células que son transportadas y reensambladas en el destino. Las células ATM tienen una longitud fija de 53 bytes, lo cual hace que la información sea transportada de manera predecible. Cada célula se divide en dos partes: la cabecera y la carga útil (figura 2.17). La carga útil es de 48 bytes y es la porción donde se transporta la información, bien sea voz, datos o vídeo. La cabecera, de 5 bytes, permite el direccionamiento de la información.

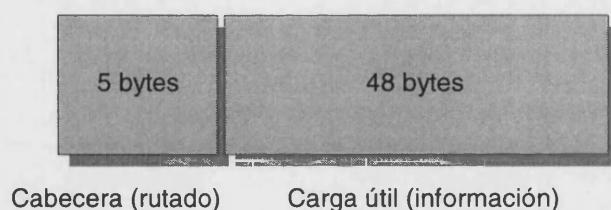


FIGURA 2.17 Formato de la célula ATM.

Al igual que en SCI, el medio físico no queda determinado en la definición del protocolo, lo cual hace que la velocidad de transmisión pueda variar. Las recomendaciones del CCITT definidas para ATM proponen velocidades de transmisión de 155 Mb/s y 622 Mb/s, aunque ya se está trabajando para aumentar la velocidad hasta 2.4 Gb/s.

ATM organiza el acceso al medio físico mediante la definición de dos conceptos: Virtual channel (VC) o canal virtual, y Virtual path (VP) o camino virtual (figura 2.18). Cada camino virtual puede alojar uno o varios canales virtuales. La idea es

realizar el direccionamiento de las células a través de los valores VC y VP contenidos en la cabecera. De algún modo, haciendo un símil con el direccionamiento paginado de memoria, el valor de VP indica el número de página y VC es el offset dentro de esa página. Puesto que el rutado en ATM es dinámico, los valores de VC y VP pueden ir variando a medida que las células atraviesan los conmutadores de la red.

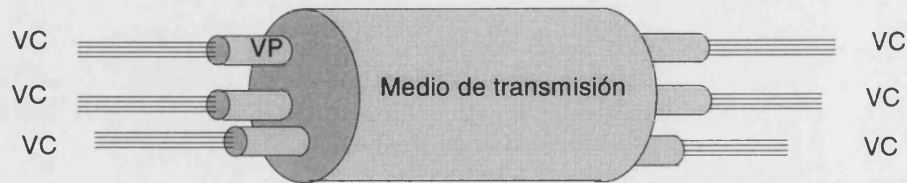


FIGURA 2.18 Organización del medio físico en ATM.

4.3.3.2 El protocolo ATM

El protocolo ATM o PRM (Protocol Reference Model) está compuesto de tres planos:

- Plano de usuario: permite al usuario transferir información. Incluye los mecanismos relacionados con el control de flujo y errores.
- Plano de control: controla la realización de la llamada y el establecimiento de la conexión.
- Plano de mantenimiento: incluye las funciones para el mantenimiento de todo el sistema que coordina los distintos planos, mantenimiento de recursos y parámetros de protocolo.

La figura 2.19 muestra esquemáticamente el protocolo ATM. Cada uno de los planos está subdividido en capas cada una de las cuales tiene una función específica.

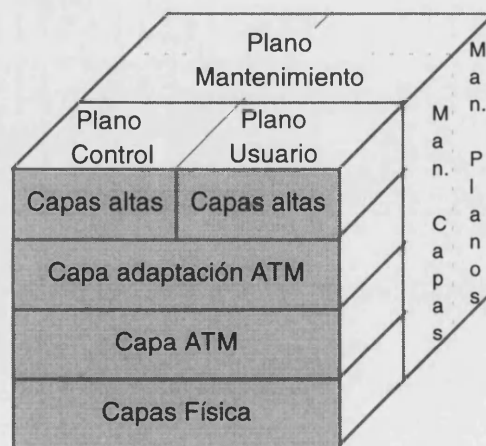


FIGURA 2.19 Modelo de capas del protocolo ATM.

La capa de adaptación ATM (AAL) asegura que la información, independientemente del tipo que sea, es segmentada e insertada de manera adecuada en la carga útil de las células. También asegura que el proceso inverso, extracción de la carga útil de las células, se realiza de forma correcta.

La capa ATM, añade a las células procedentes de la capa AAL, la cabecera para enviarlas a la dirección correcta y es independiente del medio físico que se esté empleando para la transmisión.

Por último, la capa física define las características eléctricas e interfaces de red. Es la encargada de generar las señales adecuadas para enviar la información. Puesto que ATM no define el medio físico, esta capa debe adaptarse a las características del mismo.

El funcionamiento del protocolo ATM es como sigue. En un momento dado se quiere establecer la comunicación entre un emisor y un receptor. El primer paso es reservar el canal para realizar esta transmisión. Esto lo hace el emisor, y en su petición incluye las características del canal virtual que desea emplear (básicamente la velocidad de transmisión).

Esta información es enviada a la red y fluye a través de los diversos conmutadores de ésta. En cada paso por cada conmutador, se reserva un canal virtual para la comunicación demanda (si es posible), y se reencamina la célula hacia el siguiente conmutador, de forma que tras un periodo de tiempo, la célula llega al destino, habiendo reservado en su transmisión el camino para enviar las células con la información. Cuando el canal ha sido reservado, se le notifica al emisor, el cual empieza a enviar la información en las células necesarias.

Aunque la transmisión puede ser a gran velocidad, en cada establecimiento de conexión entre dos nodos hay que contar con un tiempo adicional necesario para la reserva del canal a emplear. Por eso, la latencia (tiempo entre que se pide la conexión y se empieza a enviar datos) de ATM es superior, por ejemplo, a la de SCI, en la que no existe ese tiempo.

4.3.3.3 Arquitectura genérica de un conmutador

Sin duda, el elemento fundamental en la construcción de redes ATM es el conmutador, aunque, paradójicamente, la conmutación no está definida en la norma. No obstante, el desarrollo de conmutadores ATM es una de las actividades de investigación que más se ha desarrollado desde la aparición del estándar [30].

Puesto que la conmutación no está definida en la norma, los fabricantes emplean una gran variedad de técnicas para la construcción de los conmutadores, teniendo en mente el objetivo de aumentar su velocidad y capacidad. La conmutación ATM se diferencia de la convencional en las interfaces de entrada/salida, que han de trabajar a alta velocidad (desde 50 Mbits/s hasta 2.4 Gbits/s), y en la velocidad de los datos dentro del conmutador, superior a 80 Gbits/s.

Sin perjuicio de la implementación concreta, todo conmutador ATM debe responder al diagrama de bloques mostrado en la figura 2.20.

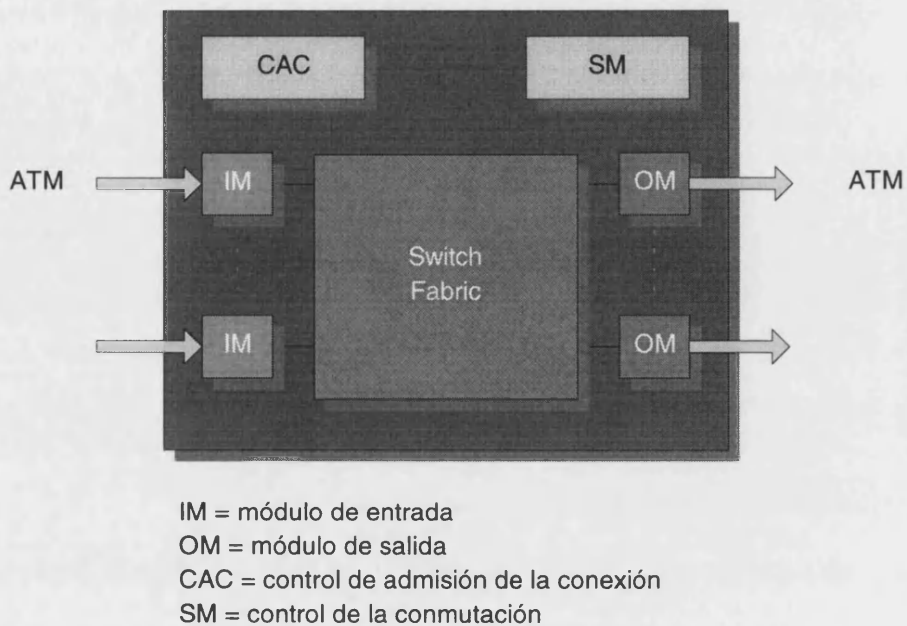


FIGURA 2.20 **Arquitectura genérica de un conmutador ATM.**

Los módulos de entrada IM hacen de interface entre la capa física y la capa ATM. El módulo extrae el flujo de células, realiza la delimitación de las mismas y la adaptación a la velocidad de transmisión. Además de esto, para cada célula debe realizar la comprobación de errores de transmisión, identificar las células de establecimiento de la conexión y pasarlas al CAC y las de operación y mantenimiento de la red que pasará al SM.

Los módulos de salida OM preparan el flujo de células ATM para su transmisión física realizando la adaptación a la velocidad del canal y la generación del código de comprobación de errores, entre otras tareas.

El módulo de control de admisión de la conexión (CAC) recibe las células de petición de establecimiento de conexión y decide hacia donde debe encaminarlas para lograr la comunicación entre el emisor y el receptor.

El módulo de control de la conmutación (SM) está conceptualmente todavía en desarrollo. Su misión es la de realizar operaciones de mantenimiento de la red, como asegurar prestaciones, mantener la seguridad, gestionar el tráfico, etc. La información para realizar estas tareas le llega a través de células OAM (operación y mantenimiento).

El módulo *switch fabric* es el principal responsable de la transmisión de células entre los otros bloques funcionales. Puede incluir también funciones de almacenamiento de células, multiplexaje y concentración de tráfico, redundancia para tolerancia a fallos, control de congestión, etc. Su implementación es una de las partes más delicadas del conmutador pues determina en gran manera las prestaciones del mismo.

Actualmente, diversos fabricantes producen conmutadores ATM. Entre ellos podemos destacar el GCNS-2000 de AT&T, el PARIS y plaNET de IBM, el ATOM y el ASX-100 de NEC y el Data Networks 1100 de Alcatel. Todos ellos cumplen el estándar ATM, diferenciándose en la implementación práctica de módulo *switch fabric*.

4.3.4 Fibre Channel (FC)

El objetivo del estándar Fibre Channel fue el desarrollo de un medio práctico, barato pero útil para transferir datos entre estaciones de trabajo, grandes computadores, ordenadores sobremesa, dispositivos de almacenamiento, pantallas y otros periféricos. Fibre Channel (FC) es el nombre genérico de un conjunto integrado de estándares desarrollados por la American National Standards Institute (ANSI) [31].

Existen dos tipos básicos de comunicación de datos entre procesadores y entre procesadores y periféricos: canales y redes. Un canal proporciona una conexión directa o punto a punto entre los dispositivos que se comunican. El canal basa la mayor parte de sus prestaciones en el hardware sobre el que se construye y transporta los datos a alta velocidad con bajas latencias. Por el contrario, una red es una agrupación de nodos distribuidos (como estaciones de trabajo, servidores de ficheros o periféricos), con sus propio protocolos, que soportan la interacción entre los nodos. Las redes tienen latencias relativamente altas puesto que basan su funcionamiento en estructuras de software, y consecuentemente son más lentas que los canales. Las redes pueden realizar una variedad de tareas mayor que los canales ya que operan en un medio de conexiones no previstas, mientras que los canales operan solo entre unos pocos dispositivos con direcciones predefinidas. FC intenta combinar lo mejor de estos dos métodos de comunicación en un nuevo interface de entrada/salida que responde a las necesidades de los usuarios de canales y de redes.

Aunque se llama Fibre Channel, su arquitectura no representa ni un canal ni cualquier tipo de topología de red, sino que permite que la conexión entre los dispositivos se realice a través de un esquema de interconexión activa e inteligente llamada *Fabric*. Todo lo que un puerto FC tiene que hacer es manejar un enlace punto a punto entre el mismo y el elemento *Fabric*.

4.3.4.1 Topología Fibre Channel

La topología de Fibre Channel gira en torno a la conexión de los diferentes elementos al conmutador *Fabric*. Esta conexión se realiza a través de un enlace consistente en dos fibras unidireccionales que transmiten la información hacia y desde el *Fabric*. Cada fibra está conectada al transmisor de un puerto en un extremo y al receptor de otro en el otro. Cuando existe un *Fabric* en la configuración, la fibra puede conectar el puerto (N_Port) de un nodo con uno de los del *Fabric* (F_Port).

Puesto que el sistema de FC descansa sobre la conexión entre sí de puertos a través del *Fabric*, es irrelevante si éste es un circuito de conmutación, un concentrador activo o un anillo. La topología puede seleccionarse en función de los requerimientos de prestaciones necesarias. Topologías posibles en FC incluyen las de punto a punto, punto a punto conmutadas o anillo (figura 2.21)

FC opera a varias velocidades de transmisión (133 Mbits/s, 266 Mbits/s, 530 Mbits/s y 1 Gbit/s) y sobre tres tipos de medios eléctricos y ópticos. Las distancias de transmisión varían en función de la combinación de velocidad y medio. La transmisión mediante fibra óptica monomodo usando una fuente de luz láser ofrece el mejor comportamiento (10 km de distancia máxima a 1 Gbit/s) [32].

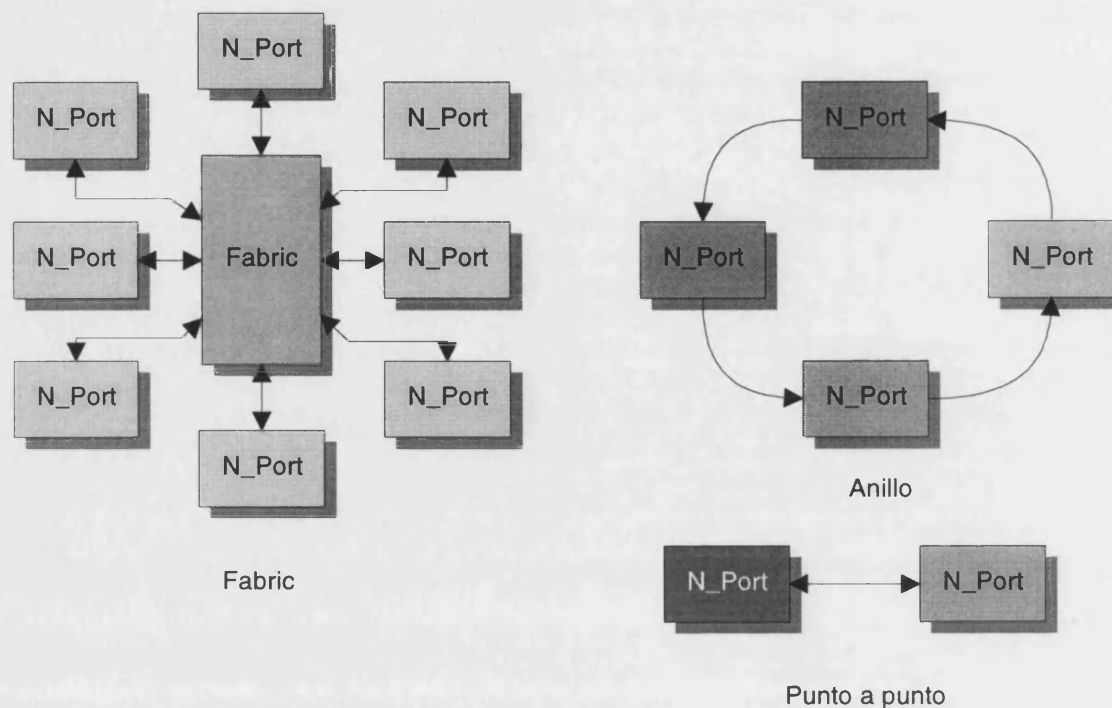


FIGURA 2.21 Topologías Fibre Channel.

4.3.4.2 Estructura del protocolo FC

Fibre Channel está estructurado como un conjunto de funciones jerárquicas organizadas en cinco niveles FC-0 a FC-4 (figura 2.22).

Nivel FC-0

El nivel más bajo (FC-0) define el enlace físico, incluyendo la fibra, los conectores y los parámetros eléctricos y ópticos para las diferentes tasas de datos. La tasa de error de bit para cualquiera de los medios y velocidades admitidos es inferior a $12 \cdot 10^{-12}$. El nivel físico está diseñado para la utilización de un gran número de tecnologías para poder acomodar el margen más amplio de requerimientos del sistema.

Nivel FC-1

El nivel FC-1 define el protocolo de transmisión incluyendo las reglas de codificación y decodificación, los caracteres especiales y el control de errores. La información enviada a través de la fibra consiste en un Carácter de Transmisión (*Transmission Character*) de 10 bits codificados a partir de grupos de 8 bits, que mejora las características de transmisión a través de la fibra y asegura que existen suficientes transiciones para poder recuperar el reloj en la recepción.

La información sin codificar consiste en ocho bits de información A, B, C, D, E, F, G, H (donde H es el bit de mayor peso) y la variable de control Z. Esta información es codificada por el nivel FC-1 en los bits a, b, c, d, e, f, g, h, i, j de un Carácter de Transmisión de 10 bits. La variable de control puede tener dos posibles valores: D para caracteres de datos o K para caracteres especiales. A cada Carácter de Transmisión se le asigna un nombre usando la convención $Z_{xx.y}$ donde Z es la variable de control del carácter sin codificar, xx es el valor decimal del número binario compuesto por los bits E, D, C, B, A, e y es el valor decimal del número binario compuesto por los bits H, G, F. Por ejemplo, el nombre del Carácter de Transmisión para el valor binario sin codificar 'BC' de tipo K es K28.5.

Los caracteres tipo D son descodificados como datos mientras que los del tipo K son usados para funciones de manejo del protocolo.

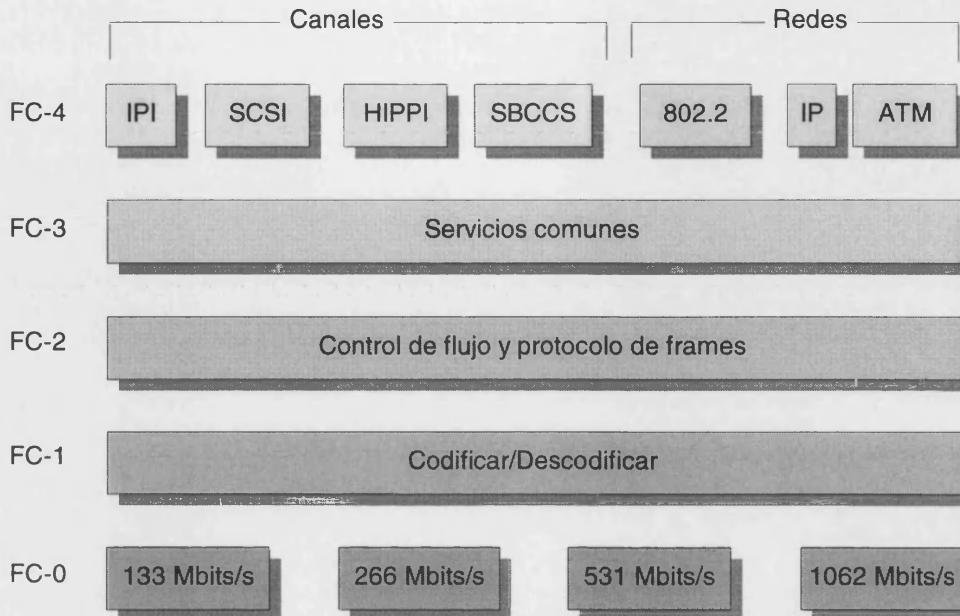


FIGURA 2.22 Estructura del protocolo Fibre Channel.

Nivel FC-2

El nivel FC-2 realiza las funciones de transporte de Fibre Channel. En él se definen las reglas para el tramado de los datos, los diferentes mecanismos para el control de la clase de servicio que se presta, y los medios para manejar las secuencias de datos definidas por el nivel FC-2. El estándar define las siguientes agrupaciones de datos:

- Conjunto ordenado (*Ordered Set*)
- Marco (*Frame*)
- Secuencia (*Sequence*)
- Intercambio (*Exchange*)
- Protocolo (*Protocol*)

Un *conjunto ordenado* consiste en una palabra de transmisión de 4 bytes que contiene datos o caracteres especiales y que tiene un significado especial. Existen tres grandes tipos de conjuntos ordenados:

- Los delimitadores de marco (Start-of-Frame y End-of-Frame). Preceden o siguen inmediatamente el contenido del marco.
- Las señales Idle y Receiver Ready (R_RDY). Se emplean para indicar que el puerto está listo para recibir o transmitir (Idle) o que el buffer de la interface con el medio está listo para recibir nuevos marcos (R_RDY).
- Las primitivas de secuencia. Se transmiten y se repiten continuamente para indicar condiciones específicas de un puerto o condiciones encontradas por la lógica del receptor de un puerto. Cuando se recibe una de estas primitivas, se genera una respuesta consistente en otra primitiva de secuencia o una señal Idle.

El *marco* es el bloque básico de una conexión FC. Contiene la información a transmitir (*PayLoad*), la dirección de los puestos de origen y destino e información de control del enlace. Los marcos se clasifican en dos grandes grupos: marcos de datos y marcos de control del enlace.

Cada marco comienza y termina con un delimitador de marco (*Frame Delimiter*) (figura 2.23). Después del delimitador de comienzo de marco se sitúa la cabecera del marco (*Frame Header*), tras el cual se sitúan como máximo 2112 bytes conteniendo la información a transferir. Tras la información se envían cuatro bytes de CRC y el delimitador de fin de marco [31][32].

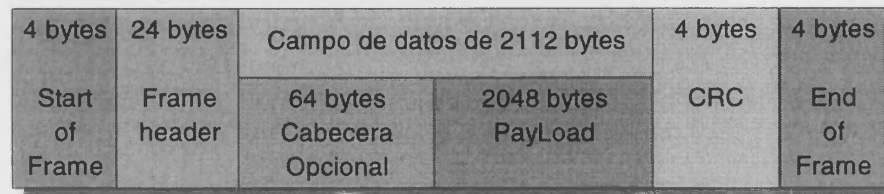


FIGURA 2.23 Estructura del Marco en Fibre Channel [32].

Una *secuencia* está formada por uno o más marcos transmitidos unidireccionalmente desde un puerto a otro. Cada marco en la secuencia está numerado de forma única.

Un *intercambio* se compone de una o más secuencias no concurrentes para una única operación. Los intercambios pueden ser unidireccionales o bidireccionales entre dos puertos. En un intercambio, solo una secuencia puede estar activa en un momento dado, aunque las secuencias de diferentes intercambios pueden estar activas concurrentemente.

Los *protocolos* están relacionados con los servicios ofrecidos por Fibre Channel. El estándar define los siguientes protocolos [31]:

- Protocolos de primitivas de secuencias. Se basan en primitivas de secuencias y están especificados para fallos en el enlace.
- Protocolo de conexión al *Fabric*. Se refiere al intercambio de parámetros entre un puerto y el *Fabric*.
- Protocolo de conexión del puerto. Antes de realizar la transferencia de datos, los puertos intercambian parámetros de transmisión a través de este protocolo.
- Protocolo de transferencia de datos. Describe los métodos para transferir datos usando el control de flujo de FC.
- Protocolo de desconexión del puerto. Se emplea cuando un puerto quiere eliminar sus parámetros de otro puerto para liberar recursos en el puerto donde se conectaba.

Los tipos de servicios que define FC son tres:

- La clase 1 proporciona conexiones específicas como si de una conexión física directa se tratase. Este servicio asegura el máximo ancho de banda.
- La clase 2 se basa en la conmutación de marcos que permite compartir el ancho de banda disponible entre los diferentes orígenes y destinos. Tanto esta clase como la clase 1 envían marcos de reconocimiento que confirman que los marcos han llegado de forma satisfactoria.

La clase 3 es idéntica a la clase 2 excepto que en este caso no se envían marcos de reconocimiento, con lo que se gana velocidad en la transmisión.

Nivel FC-3

El nivel FC-3 proporciona los servicios comunes requeridos por mecanismos de comunicación avanzados como:

- *Striping*. Multiplica el ancho de banda usando varios puertos en paralelo para enviar una única unidad de información a través de múltiples enlaces.
- *Hunt groups*. Posibilidad de que más de un puerto responda a la misma dirección. Esto mejora la eficiencia al decrementar la posibilidad de encontrar un puerto ocupado.
- *Multicast*. Envía una información a varios puertos de destino. Esta opción incluye la de enviar a todos los puertos de conectados a un mismo *Fabric (broadcast)*.

Nivel FC-4

El nivel FC-4 es el más alto en la estructura de FC y define las interfaces de aplicación que pueden ejecutarse sobre Fibre Channel. Especifica las reglas para la utilización de los niveles FC inferiores por parte de protocolos superiores. Fibre Channel permite tanto el empleo de protocolos basados en canales o en redes. Los siguientes protocolos están actualmente especificados o propuestos para el nivel FC-4 [32]:

- Small Computer System Interface (SCSI).
- Intelligent Peripheral Interface (IPI).
- High Performance Parallel Interface (HIPPI).
- Internet Protocol (IP).
- ATM Adaptation Layer para datos de ordenador (AAL5).
- Link Encapsulation (FC-LE).
- Single Byte Command Code Set Mapping (SBCCS).
- IEEE 802.2 Logical Link Control.

5. ARQUITECTURAS PARA LA INTEGRACIÓN DE SENSORES

En el capítulo 1 ya se apuntó que en aquellos casos en los que el volumen de información a tratar era elevado, el sistema de procesado de sensores se organizaba de manera jerárquica y generalmente en tres niveles. Sin embargo, no se ha encontrado en la literatura una comparación de la adecuación de otras soluciones a este problema.

En este punto, y después de haber estudiado las implementaciones posibles para un sistema con múltiples procesadores, se va a profundizar en el estudio de la arquitectura jerárquica para la integración de sensores, comparándola con una solución totalmente paralela y obteniendo las expresiones que permiten escoger una u otra a la hora de solucionar un determinado problema.

5.1 PROBLEMAS CON ELEVADA TASA DE DATOS

En la actualidad, debido a la mejora de la tecnología, es posible tratar grandes cantidades de datos de forma casi instantánea. Sin embargo, esta mejora no da solución a una serie de problemas que antes del desarrollo de los integrados VLSI se consideraban inabordables y que ahora sólo lo son mediante el empleo de sistemas multiprocesador.

En el procesamiento de imágenes hasta no hace mucho se empleaban imágenes de baja resolución debido a que no se tenía la potencia de procesamiento suficiente. Con la mejora de la tecnología ha sido posible tratar imágenes cada vez de mayor resolución, pero ahora el reto es hacerlo con imágenes de vídeo, lo que supone aún mayores potencias de procesamiento.

En el campo de la experimentación en física de altas energías, se empleaban detectores de traza para poder observar las trayectorias de las partículas. A menor potencia de procesamiento, menor resolución en la determinación de la traza, lo cual se convertía en posibles imprecisiones a la hora de identificar la partícula. Con la aparición de procesadores más potentes es posible plantearse la construcción de sistemas que obtengan mejores resoluciones.

Estos dos, son ejemplos representativos de áreas de investigación donde el volumen de información proporcionado por los sensores es elevado, aunque no son los únicos puesto que también podemos encontrarnos estos problemas en áreas como la robótica, los sistemas de control aeroespaciales o los sistemas de predicción meteorológica.

Tradicionalmente, en estos casos donde tenemos un gran volumen de información procedente de los sensores, se han empleado arquitecturas jerárquicas para el procesamiento de la misma. Estas arquitecturas, compuestas de tres o más niveles conseguían la resolución del problema. Muchas veces, cada uno de los niveles se constituía como un sistema MIMD, de forma que el sistema global era un conjunto de sistemas MIMD interconectados entre sí (figura 2.24).

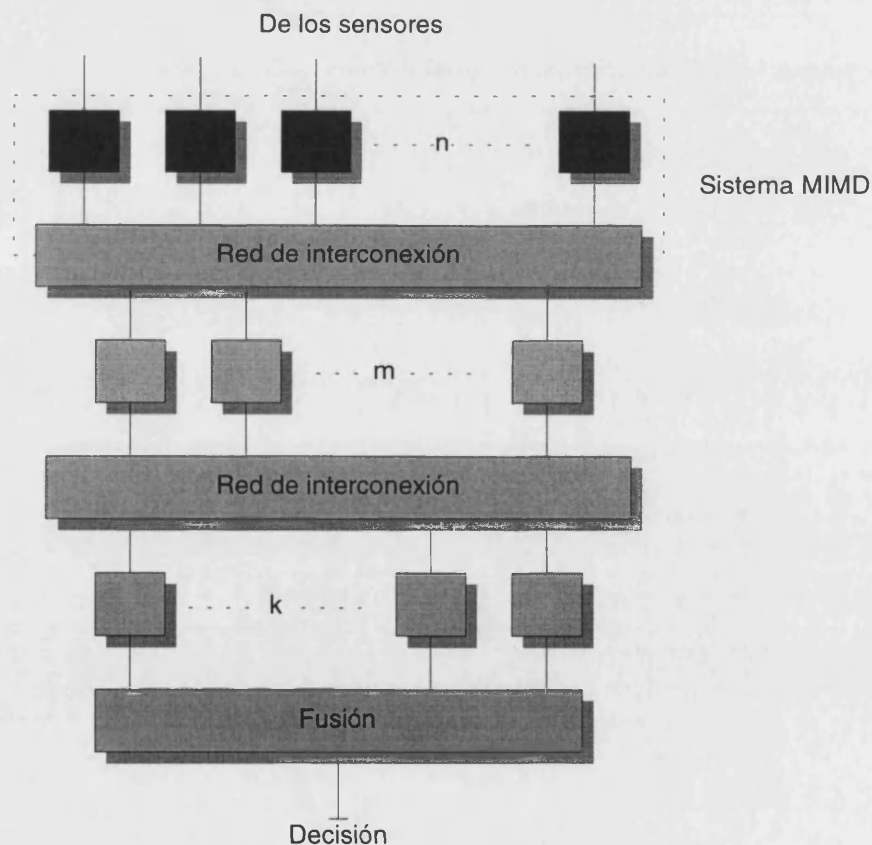


FIGURA 2.24 Sistema jerárquico compuesto de sistemas MIMD para la integración de sensores.

Sin embargo, la opción de una arquitectura jerárquica no es siempre la mejor, como a continuación se demostrará. En efecto, ante un problema de procesamiento de grandes volúmenes de información se pueden plantear diversas soluciones entre las que se encuentra el sistema jerárquico, pero también la posibilidad de implementar un sistema MIMD paralelo para tratar de información.

¿Cuándo convendrá emplear uno u otro sistema? Esta pregunta no tiene fácil solución puesto que depende de factores tales como el mismo problema a tratar, la tecnología disponible la fiabilidad, complejidad y prestaciones requeridas a la solución e, inevitablemente, el presupuesto del que se disponga. Nosotros vamos a centrar nuestro análisis en las prestaciones del sistema, y vamos a definir un parámetro que nos determina la calidad requerida a un sistema para cumplir con las especificaciones del problema que nos permita la comparación de los sistemas.

5.1.1 Calidad requerida de un sistema

Puesto que, en general, los sistemas a comparar pueden estar implementados de maneras totalmente diferentes, necesitamos un parámetro que sea independiente de la implementación concreta y que, por otra parte, nos indique, de alguna manera, cuál de los dos sistemas será más complejo, difícil y/o costoso de implementar.

Para ello, definimos un parámetro de calidad requerida de un sistema Q como el producto del ancho de banda, AB_R , por la capacidad de procesamiento, CP_R , necesarias para poder solucionar un determinado problema. Es decir,

$$Q = AB_R \cdot CP_R \quad (2.8)$$

Según esto un problema con alto Q conlleva unas grandes prestaciones pues implicará un gran número de operaciones (CP alto) sobre una gran tasa de datos (AB alto).

Para después poder obtener las expresiones necesarias para los sistemas paralelo y jerárquico nos hace falta conocer cuál sería el valor del parámetro Q cuando se asocian en serie y en paralelo un número determinado de procesadores cada uno con su parámetro Q específico.

Calidad requerida en un sistema paralelo

En este caso tenemos un número N de procesadores conectados en paralelo cada uno con un valor de Q_i determinado. La pregunta que nos hacemos es, ¿qué valor de Q tendría un sistema formado por un solo procesador que equivaliese a los N del sistema paralelo?

Para responder a esta pregunta, evaluamos por separado el ancho de banda total del sistema paralelo y su capacidad de procesamiento.

Evidentemente, el ancho de banda total del sistema será la suma de los anchos de banda, AB_i , de cada uno de los elementos de procesamiento, es decir:

$$AB_P = \sum_{i=1}^N AB_i \quad (2.9)$$

La capacidad de procesamiento también es la suma de las capacidades CP_i de cada uno de los elementos de procesamiento:

$$CP_P = \sum_{i=1}^N CP_i \quad (2.10)$$

Por tanto la calidad requerida de un sistema paralelo en función de las calidades requeridas de cada elemento que lo componen es:

$$Q_P = AB_P \cdot CP_P = \sum_{i=1}^N AB_i \cdot \sum_{i=1}^N CP_i = AB_P \cdot \sum_{i=1}^N \frac{Q_i}{AB_i} = CP_P \cdot \sum_{i=1}^N \frac{Q_i}{CP_i} \quad (2.11)$$

Calidad requerida en un sistema serie

Supongamos ahora el caso en el que los procesadores se conectan en serie, es decir, la salida de uno es la entrada del siguiente. En este caso, si tenemos N procesadores, cada uno con su ancho de banda AB_i y su capacidad de procesamiento CP_i , el sistema equivalente con un solo procesador deberá tener el ancho de banda del primero de la serie, es decir:

$$AB_S = AB_1 \quad (2.12)$$

La capacidad de procesamiento equivalente, por su parte, vendrá determinada por el tiempo de procesamiento total, t_N , suma del de cada uno de los procesadores. En este tiempo, el número de operaciones a realizar será la suma de las que hacen todos los procesadores. Por lo tanto,

$$CP_S = \frac{D \cdot \sum_{i=1}^N op_i}{t_N} = \sum_{i=1}^N CP_i \cdot \frac{t_i}{t_N} \quad (2.13)$$

donde D es el número de datos y op_i , el número de operaciones que realiza el procesador i .

De las dos expresiones, la calidad requerida al sistema equivalente es:

$$Q_S = AB_S \cdot CP_S = AB_1 \cdot \sum_{i=1}^N CP_i \cdot \frac{t_i}{t_N} = \frac{AB_1}{t_N} \cdot \sum_{i=1}^N CP_i \cdot t_i \cdot \frac{AB_i}{AB_i} = \frac{AB_1}{t_N} \cdot \sum_{i=1}^N Q_i \cdot \frac{t_i}{AB_i} \quad (2.14)$$

5.1.2 Parametrización de las arquitecturas paralela y jerárquica

Un determinado problema de procesamiento podemos parametrizarlo indicando el ancho de banda total AB requerido para leer los datos y la capacidad de procesamiento total CP necesaria para el tratamiento de los mismos.

Si resolvemos el problema mediante una arquitectura paralela como la de la figura 2.25 tendremos las siguientes expresiones:

$$\begin{aligned} AB_i &= \frac{AB}{N} & 1 \leq i \leq N \\ CP_i &= \frac{CP}{N} & 1 \leq i \leq N \end{aligned} \quad (2.15)$$

donde N es el número de procesadores del sistema.

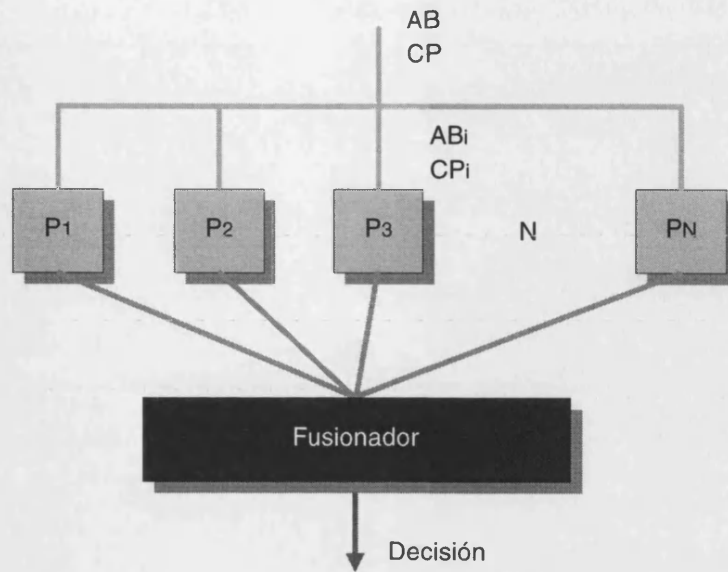


FIGURA 2.25 Sistema de procesamiento paralelo.

Sin embargo, es habitual disponer de una serie de unidades de procesamiento (up) con unas características determinadas en cuanto a capacidad de procesamiento $CP_i = CP_{up}$ y ancho de banda $AB_i = AB_{up}$. Por lo tanto, el parámetro a determinar será el número de unidades con esas características que nos hacen falta para implementar nuestro sistema. Este número, N , es igual a :

$$N = \text{máx} \left(\frac{AB}{AB_i}, \frac{CP}{CP_i} \right) \quad (2.16)$$

El fusionador recoge las decisiones de los N elementos de procesamiento y elabora una decisión final. Las formas de elaborar esta decisión son muy variadas. Para nuestro análisis vamos a suponer que el sistema ofrece una decisión binaria del tipo (*sí/no*) y que se elabora a partir de la *and* lógica de todas las decisiones. De esta manera el sistema ofrecerá un *sí* sólo cuando todas las decisiones parciales sean *sí*.

El ancho de banda que requiere el fusionador depende del tamaño de las decisiones parciales, del número N de elementos de procesamiento y del tiempo T que se tiene para realizar la decisión. La capacidad de procesamiento depende del número de operaciones *and* que haya que realizar y del tiempo T que hay para realizarlas. Suponiendo que estamos empleando operaciones *and* de dos entradas, el número de operaciones necesarias para obtener el resultado si tenemos N entradas es $N-1$.

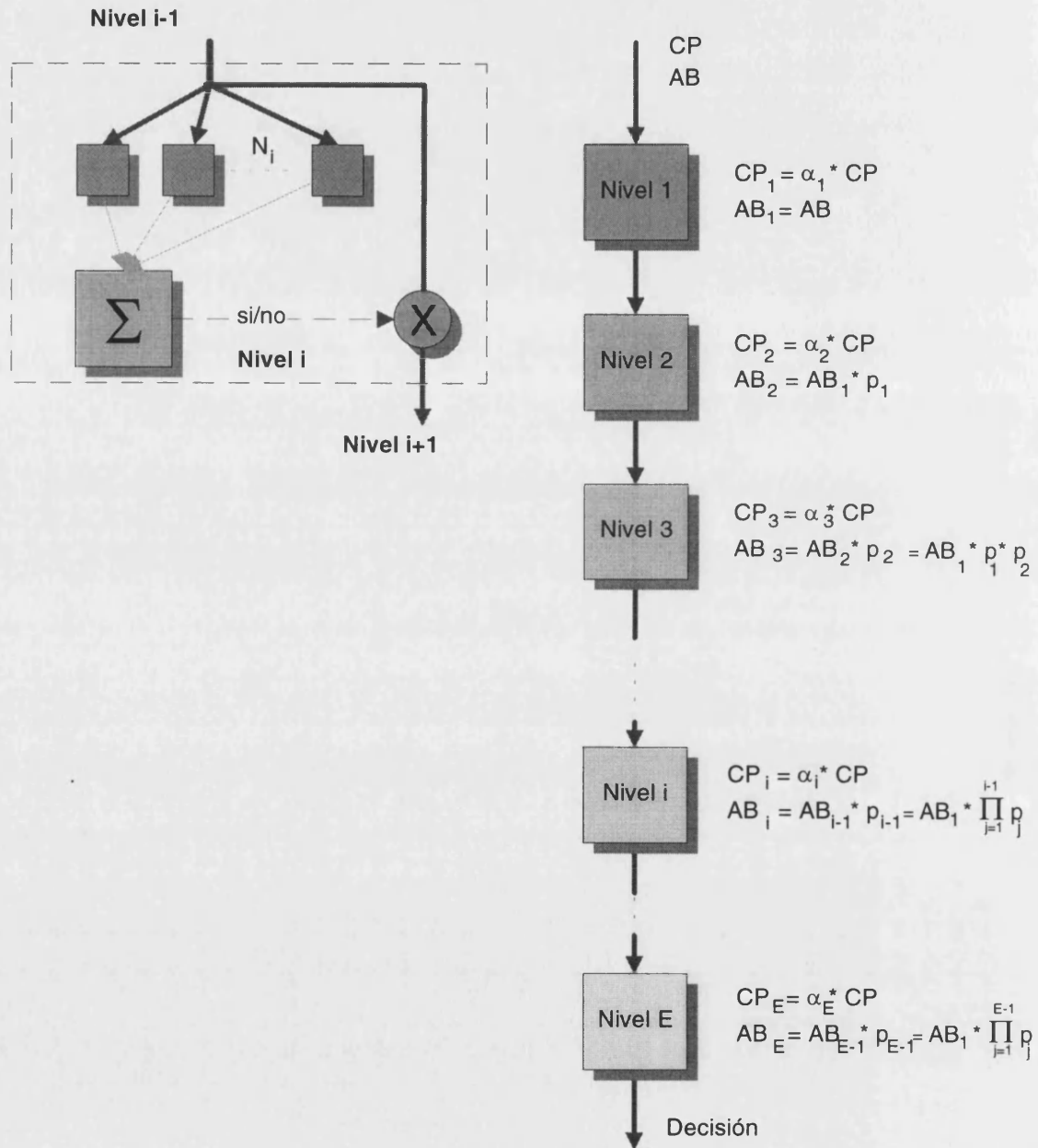
En nuestro caso estamos suponiendo que el sistema funciona en modo pipeline: en primer lugar se obtienen las decisiones parciales en un tiempo T , tras esta fase se obtiene la decisión final también en un tiempo T , a tiempo de recibir las siguientes. De esta manera se solapa la obtención de las decisiones parciales y la decisión final.

Si S_{dec} es el tamaño en bytes de la decisión parcial, S_{dato} el número de bytes por dato procedente de los sensores, ξ la relación entre el tamaño de la decisión parcial y el tamaño del dato procedente de cada sensor, D el número de datos, op el número de

operaciones por dato y K_1, K_2 son valores constantes para un determinado problema, podemos expresar ambos parámetros como:

$$AB_f = N \cdot \frac{S_{dec}}{T} = \left(\frac{S_{dec} = \xi \cdot S_{dato}}{AB = \frac{D \cdot S_{dato}}{T}} \right) = N \cdot \frac{\xi}{D} \cdot AB = N \cdot K_1 \cdot AB$$

$$CP_f = \frac{N-1}{T} = \left(CP = \frac{D \cdot op}{T} \right) = (N-1) \cdot \frac{1}{D \cdot op} \cdot CP = (N-1) \cdot K_2 \cdot CP$$



(2.17)

FIGURA 2.26 Sistema de procesamiento jerárquico generalizado.

Si ahora planteamos la resolución del problema mediante el empleo de una arquitectura jerárquica como la mostrada en la figura 2.26, tendremos que introducir un parámetro nuevo para poder obtener las siguientes expresiones para AB_i y CP_i que en este caso serán función del nivel en la arquitectura. Este parámetro es el factor de reducción en el volumen de datos debido a la extracción de la información de los datos en bruto recibidos de los sensores, en el caso de un sistema de medida, o la proporción de datos descartados por no cumplir ciertos requisitos si se trata de un sistema de detección.

Hay que hacer notar como en cada nivel, el ancho de banda AB_i disminuye en un factor igual a la reducción debida a la extracción de información o a la eliminación de datos por no ser de interés (un factor p). En nuestro caso supondremos que de nivel a nivel o pasan todos los datos (porque se ha decidido que *sí*) o ninguno (decisión *no*). Como esta decisión tiene una probabilidad p , el periodo entre paso de la información viene aumentado por la inversa de esa probabilidad. Puesto que se envían todos los datos, el efecto es un tiempo mayor para enviar la misma cantidad de datos, lo que conlleva una reducción del ancho de banda.

La capacidad de procesado en cada nivel i es proporcional, según el factor α_i , a la capacidad de procesado total CP .

Cada uno de los niveles tiene una estructura formada por un sistema paralelo con un fusionador cuya decisión hace que la totalidad de los datos pase o no hacia el siguiente nivel.

5.1.3 Evaluación de la calidad requerida

Según se definió en el epígrafe 5.1.1, podemos ahora evaluar el parámetro de calidad requerida para el sistema paralelo y el jerárquico.

Sistema paralelo

El factor Q para el sistema paralelo tiene dos términos. El primero, Q_{proc} , depende de los procesadores empleados y el segundo, Q_{fus} , del fusionador. De esta manera, el equivalente del sistema paralelo es una asociación en paralelo de N procesadores en serie con el fusionador.

Si suponemos que todos los procesadores empleados tienen las mismas características, aplicando la ecuación (2.23) tenemos:

$$Q_{proc} = AB_{proc} \cdot CP_{proc} = \sum_{i=1}^N AB_i \cdot \sum_{i=1}^N CP_i = N \cdot AB_{up} \cdot N \cdot CP_{up} = N^2 \cdot Q_{up} \quad (2.18)$$

donde N es el número de unidades y Q_{up} es el factor de calidad requerida de las unidades de procesado empleadas.

El segundo término, relacionado con el fusionador es:

$$Q_{fus} = AB_{fus} \cdot CP_{fus} = N \cdot (N-1) \cdot K_1 \cdot K_2 \cdot AB \cdot CP = N \cdot (N-1) \cdot K \cdot AB \cdot CP \quad (2.19)$$

El factor de calidad del sistema paralelo será el serie de los dos calculados, es decir:

$$\begin{aligned}
Q_P &= \frac{AB_{proc}}{t + t} \cdot \sum_{i=1}^2 Q_i \cdot \frac{t}{AB_i} = \frac{AB_{proc}}{2} \cdot \left[\frac{N^2 \cdot AB_{up} \cdot CP_{up}}{AB_{proc}} + \frac{N \cdot (N-1) \cdot K \cdot AB \cdot CP}{N \cdot K_1 \cdot AB} \right] = \\
&= \frac{N \cdot b \cdot AB}{2} \cdot \left[\frac{N^2 \cdot b \cdot AB \cdot a \cdot CP}{N \cdot b \cdot AB} + \frac{N \cdot (N-1) \cdot K_1 \cdot K_2 \cdot AB \cdot CP}{N \cdot K_1 \cdot AB} \right] = \quad (2.20) \\
&= AB \cdot CP \cdot \frac{N \cdot b}{2} \cdot [N \cdot a + (N-1) \cdot K_2]
\end{aligned}$$

donde a y b son las relaciones entre la capacidad de procesado, CP_{up} , y el ancho de banda, AB_{up} , respectivamente, de cada procesador y el total del problema.

Sistema jerárquico

El sistema jerárquico no es más que una serie de E sistemas paralelos conectados en serie. En cada uno de los niveles el sistema está formado por N_i procesadores y un fusionador. Por tanto el factor de calidad requerida, Q_H , será:

$$Q_H = \frac{AB_1}{\sum_{i=1}^E t_i} \sum_{i=1}^E Q_{P_i} \quad (2.21)$$

donde Q_{P_i} es el factor de calidad requerido en cada nivel, que se expresa como:

$$Q_{P_i} = \frac{AB_i}{t_i + t_i} \cdot \sum_{j=1}^2 Q_{ij} \cdot \frac{t_j}{AB_j} = \frac{AB_{proc_i}}{2} \cdot \left(\frac{Q_{proc_i}}{AB_{proc_i}} + \frac{Q_{fus_i}}{AB_{fus_i}} \right) \quad (2.22)$$

suponiendo, como se hizo en el caso paralelo, que $t_j = t_i \forall j$.

El factor de calidad requerido para los procesadores es:

$$Q_{proc_i} = N_i^2 \cdot a_i \cdot b_i \cdot AB \cdot CP \quad (2.23)$$

donde a_i y b_i tienen el mismo sentido definido en el sistema paralelo, y N_i está definido como:

$$N_i = \max \left(\frac{AB_i}{AB_{up}}, \frac{CP_i}{CP_{up}} \right) = \max \left(\frac{AB \cdot \prod_{j=0}^{i-1} p_j}{AB_{up}}, \frac{\alpha_i \cdot CP}{CP_{up}} \right) \quad (2.24)$$

Fijándonos en la figura 2.26, y haciendo

$$\begin{aligned}
P_i &= \prod_{j=1}^{i-1} p_j \quad i \geq 2 \\
P_1 &= 1
\end{aligned} \quad (2.25)$$

podemos calcular el factor de calidad requerida del fusionador. Para ello primero calculamos el ancho de banda del fusionador del nivel i que resulta ser:

$$\begin{aligned}
AB_{fus_i} &= N \cdot \frac{S_{dec}}{t_i} = (t_i = T/P_i) = N \cdot \frac{S_{dec}}{T} \cdot P_i = \left[\begin{array}{l} S_{dec} = \xi \cdot S_{dato} \\ AB = \frac{D \cdot S_{dato}}{T} \end{array} \right] = \\
&= N \cdot \frac{\xi}{D} \cdot P_i \cdot AB = N \cdot K_1 \cdot P_i \cdot AB
\end{aligned} \tag{2.26}$$

Por otra parte, la capacidad de procesamiento necesaria en el fusionador del nivel i se puede expresar como:

$$\begin{aligned}
CP_{fus_i} &= \frac{N-1}{t_i} = (t_i = T/P_i) = \frac{N-1}{T} \cdot P_i = \left[CP = \frac{D \cdot op}{T} \right] = (N-1) \cdot \frac{1}{D \cdot op} \cdot P_i \cdot CP = \\
&= (N-1) \cdot K_2 \cdot P_i \cdot CP
\end{aligned} \tag{2.27}$$

Por lo tanto, el factor de calidad requerida del fusionador del nivel i será:

$$Q_{fus_i} = AB_{fus_i} \cdot CP_{fus_i} = N \cdot (N-1) \cdot K_1 \cdot K_2 \cdot P_i^2 \cdot AB \cdot CP \tag{2.28}$$

Sustituyendo en (2.19) nos queda:

$$\begin{aligned}
Q_{P_i} &= \frac{AB_{proc_i}}{2} \cdot \left(\frac{N_i^2 \cdot a_i \cdot b_i \cdot AB \cdot CP}{N_i \cdot b_i \cdot AB} + \frac{N_i \cdot (N_i-1) \cdot K \cdot P_i^2 \cdot AB \cdot CP}{N_i \cdot K_1 \cdot P_i \cdot AB} \right) = \\
&= AB \cdot CP \cdot \frac{N_i \cdot b_i}{2} \cdot (N_i \cdot a_i + (N_i-1) \cdot K_2 \cdot P_i)
\end{aligned} \tag{2.29}$$

Por tanto, el factor de calidad requerida del sistema jerárquico, Q_H , resulta:

$$\begin{aligned}
Q_H &= \frac{AB_1 \cdot CP}{\sum_{j=1}^E t_j} \cdot \sum_{i=1}^E \frac{N_i \cdot b_i}{2} \cdot (N_i \cdot a_i + (N_i-1) \cdot K_2 \cdot P_i) \cdot \frac{t_i}{AB_i} = \\
&= \frac{AB^2 \cdot CP}{\sum_{j=1}^E t_j} \cdot N_1 \cdot b_1 \cdot \sum_{i=1}^E \frac{N_i \cdot b_i}{2} \cdot (N_i \cdot a_i + (N_i-1) \cdot K_2 \cdot P_i) \cdot \frac{t_i}{N_i \cdot b_i \cdot AB} = \\
&= \frac{AB \cdot CP}{\sum_{j=1}^E t_j} \cdot N_1 \cdot b_1 \cdot \sum_{i=1}^E \frac{1}{2} \cdot (N_i \cdot a_i + (N_i-1) \cdot K_2 \cdot P_i) \cdot t_i = \\
&= \frac{AB \cdot CP}{\sum_{j=1}^E t_1/P_j} \cdot N_1 \cdot b_1 \cdot \sum_{i=1}^E \frac{1}{2} \cdot (N_i \cdot a_i + (N_i-1) \cdot K_2 \cdot P_i) \cdot \frac{t_1}{P_i} \Rightarrow \\
\Rightarrow Q_H &= AB \cdot CP \cdot \frac{N_1 \cdot b_1}{2} \cdot \sum_{i=1}^E \frac{(N_i \cdot a_i + (N_i-1) \cdot K_2 \cdot P_i)}{P_i \cdot \sum_{j=1}^E 1/P_j}
\end{aligned} \tag{2.30}$$

Evidentemente la ecuación (2.30) se hace igual a la ecuación (2.20), correspondiente del caso paralelo, cuando el número de niveles E es uno.

El siguiente paso es realizar la comparación entre las expresiones para el caso paralelo y el jerárquico y tratar de obtener una expresión analítica que nos permita decidir ante un problema dado cuál será la mejor solución. Sin embargo, si tomásemos el valor del factor de calidad requerida en el caso paralelo como dato, dispondríamos de una sola ecuación y $4 \cdot E$ incógnitas, por lo que el sistema tendrá infinitas soluciones.

Por ello, un método más lógico de actuar para comparar las dos soluciones antes un problema dado sería el siguiente:

- se determina la solución paralela y se calcula su factor de calidad requerida.
- se diseña el sistema jerárquico y se calcula su factor de calidad requerida.
- se comparan ambos resultados. Si estamos interesados en una solución jerárquica y su factor de calidad requerida es mayor, se asignan valores diferentes a los parámetros del sistema (a , b , P y α) y se repite el proceso hasta que el factor sea menor que el de la solución paralela.

No obstante, es posible obtener una expresión analítica si lo que se conoce es el valor del factor de calidad requerida del sistema paralelo y los valores de los parámetros de $E-1$ niveles del sistema jerárquico de E niveles. En este caso, imponiendo la condición de que, por ejemplo, el factor de calidad requerida en el sistema jerárquico sea menor que el del sistema paralelo, se obtiene la siguiente expresión para la obtención de los valores de los parámetros del último nivel:

$$\frac{N_E \cdot a_E + (N_E - 1) \cdot K_2 \cdot P_E}{P_E \cdot \sum_{j=1}^E 1/P_E} < \frac{N \cdot b}{N_E \cdot b_E} \cdot [N \cdot a + (N - 1) \cdot K_2] - \sum_{i=1}^{E-1} \frac{N_i \cdot a_i + (N_i - 1) \cdot K_2 \cdot P_i}{P_i \cdot \sum_{j=1}^E 1/P_E} \quad (2.31)$$

5.2 INTRODUCCIÓN DE ELEMENTOS DE PREPROCESADO

En el punto anterior hemos estudiado tanto el sistema jerárquico como el sistema paralelo para el procesamiento de sensores. En este apartado vamos a presentar una mejora sobre el sistema jerárquico consistente en la introducción de elementos de preprocesado anteriores a los niveles de forma que se mejoran las prestaciones del sistema al reducir el procesamiento en los mismos.

5.2.1 Regiones de Interés

Cuando el número de canales a procesar es muy grande, la capacidad de cálculo y el ancho de banda necesario en los niveles de procesamiento pueden llegar a ser demasiado elevados.

En el campo del procesamiento de imágenes, muchas veces se acude a sistemas de procesamiento jerárquicos en los que sucesivos niveles realizan el procesamiento con resoluciones cada vez mayores [33]. También en este campo, sobretodo en el análisis de imágenes en movimiento, debido a que existe gran correlación entre ellas, se utiliza una técnica consistente en localizar las zonas, *regiones de interés* o *region of interest (RoI)*, de la imagen que han cambiado de un momento a otro, y realizar el procesamiento sólo en éstas, con la consecuente reducción en el tiempo de procesamiento.

La idea es aplicar el concepto de región de interés a un sistema jerárquico de procesamiento de sensores con gran volumen de información para aligerar la carga computacional en los niveles donde esto sea posible.

Podemos definir la región de interés Ω como un conjunto de canales procedentes de los sensores del sistema. La región de interés puede representar un espacio de medida uni, bi o tridimensional del entorno físico sensed. En el caso más general de un espacio tridimensional podemos expresarla como:

$$\Omega = \begin{bmatrix} c_{111} & \cdots & c_{11n} \\ \vdots & & \vdots \\ c_{1m1} & \cdots & c_{1mn} \\ c_{211} & \cdots & c_{21n} \\ \vdots & & \vdots \\ c_{2m1} & \cdots & c_{2mn} \\ \vdots & & \vdots \\ c_{p11} & \cdots & c_{p1n} \\ \vdots & & \vdots \\ c_{pm1} & \cdots & c_{pmn} \end{bmatrix} \quad (2.32)$$

donde n , m y p es el número de canales en cada uno de los ejes x , y , z .

El tamaño de la región de interés será:

$$\dim(\Omega) = n \cdot m \cdot p \quad (2.33)$$

En cada toma de datos, se localizarán un número determinado de regiones de interés. Llamaremos $\overline{n_{RoI}}$ al número medio de regiones de interés por adquisición de datos.

Si empleamos las regiones de interés, en cada adquisición las etapas de filtrado sólo procesaran los canales de las regiones de interés para tomar la decisión acerca del suceso. En cada nivel del sistema jerárquico puede aplicarse el concepto de región de interés y utilizar sólo esos canales para realizar la decisión. En general, el número medio de canales a procesar será menor que el total. La fracción de canales a procesar frente a canales totales es:

$$\delta = \frac{\dim(\Omega) \cdot \overline{n_{RoI}}}{N_{ch}} \quad (2.34)$$

done N_{ch} es el número total de canales del sistema de sensores.

Al introducir regiones de interés, hace falta añadir a la arquitectura de procesado los módulos necesarios para el cálculo de las mismas, que denominaremos *constructores de regiones de interés*. Estos módulos se dispondrán entre los niveles de la jerarquía, y como éstos, los podemos modelar por un ancho de banda y una capacidad de procesador necesarios (figura 2.27). El ancho de banda de cada módulo constructor situado entre dos niveles es igual al del nivel correspondiente, y su capacidad de procesado podemos expresarla como una fracción γ_i de la total del problema.

Si suponemos que el constructor (*RoI builder* o *RB*) en cada nivel i está formado por un sistema paralelo con unidades de ancho de banda AB_{upRB} y capacidad de procesado CP_{upRB} , el número de unidades N_{RBi} unidades necesarias será:

$$N_{RB_i} = \max\left(\frac{\gamma_i \cdot CP}{CP_{RB_i}}, \frac{AB \cdot P_i}{AB_{RB_i}}\right) \quad (2.35)$$

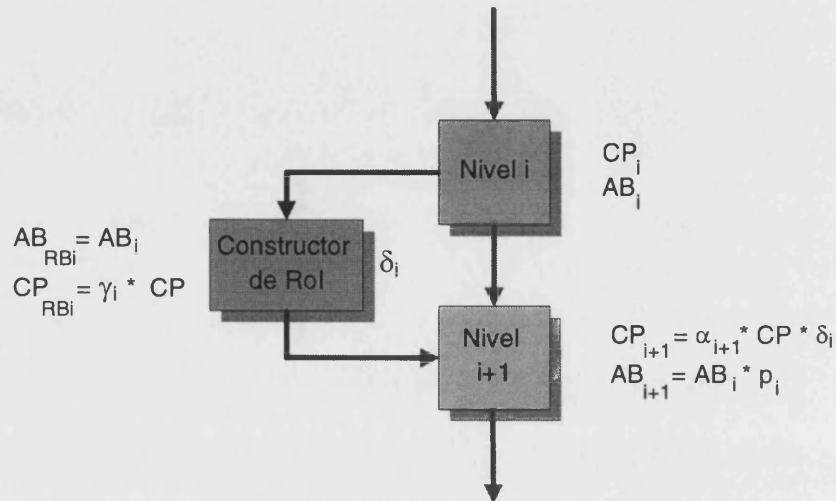


FIGURA 2.27 Disposición del constructor de RoI en la arquitectura jerárquica.

El factor de calidad requerida del constructor se expresará como:

$$Q_{RB_i} = \sum_{j=1}^{N_{RB_i}} AB_{upRB} \cdot \sum_{j=1}^{N_{RB_i}} CP_{upRB} = \quad (2.36)$$

$$= N_{RB_i}^2 \cdot AB_{upRB} \cdot CP_{upRB} = N_{RB_i}^2 \cdot a_{RB_i} \cdot b_{RB_i} \cdot AB \cdot CP \quad 1 \leq i < E$$

donde, al igual que se ha hecho antes, a_{RB_i} y b_{RB_i} , son las relaciones entre la capacidad de procesado y el ancho de banda de cada unidad del constructor y los valores totales del problema. Obsérvese que no existe constructor del nivel E al E+1, de ahí los límites para el índice i .

El ancho de banda de los niveles de la jerarquía queda también modificado al tener que aceptar ahora también los datos relativos a la región de interés, sin embargo, en este estudio vamos a despreciar esta aportación al suponer que estamos tratando con un problema donde la tasa de datos es muy elevada comparada con la debida a este hecho.

La figura 2.28 muestra el esquema modificado para el empleo de regiones de interés. Hay que hacer notar que no es siempre necesario introducir constructores entre todos los niveles de la jerarquía, sino que esto es función de la aplicación concreta.

Puesto que se reduce el número de canales a procesar, también lo hará la capacidad de procesado necesaria en el nivel de la jerarquía donde se empleen regiones de interés, aunque no el ancho de banda puesto que cada nivel debe ser capaz de leer todos los canales y no sólo aquéllos seleccionados por el constructor. El factor de reducción es igual a la fracción δ por lo que el factor de calidad de ese nivel queda afectado en un factor $\delta < 1$, es decir:

$$Q'_i = \delta \cdot Q \quad (2.37)$$

Al variar la capacidad de procesamiento puede que varíe el número de unidades necesarias en cada uno de los niveles, dependiendo de si era este parámetro o el ancho de banda el que lo determinaba. En general, el número de unidades de procesamiento, N'_i , al introducir regiones de interés es:

$$N'_i = \max \left(\frac{\delta_{i-1} \cdot CP_i}{CP_{ij}}, \frac{P_i \cdot AB}{AB_{ij}} \right) = \max \left(\frac{\delta_{i-1} \cdot \alpha_i \cdot CP}{CP_{ij}}, \frac{P_i \cdot AB}{AB_{ij}} \right) \quad (2.38)$$

para $i > 1$, con $\delta_0 = 1$ (puesto que no existe constructor del nivel "0" al nivel "1"). Observemos también que se ha supuesto que se están empleando unidades de procesamiento de iguales características a las que se empleaban en el sistema sin regiones de interés.

El factor de calidad requerida del fusionador se ve modificado por la utilización de regiones de interés puesto que varía en número de unidades de procesamiento en los niveles aunque no el tiempo disponible para el procesamiento y transmisión de la información.

Con todo, el factor de calidad de cada nivel de la jerarquía queda modificado y pasa a valer:

$$\begin{aligned} Q'_{P_i} &= \frac{AB'_{proc_i}}{2 \cdot t_i + t_{RB_i}} \cdot \left(\frac{N_{RB_i}^2 \cdot a_{RB_i} \cdot b_{RB_i} \cdot AB \cdot CP}{N_{RB_i} \cdot b_{RB_i} \cdot AB} \cdot t_{RB_i} + \right. \\ &\quad \left. + \frac{N_i'^2 \cdot a_i \cdot b_i \cdot AB \cdot CP}{N_i' \cdot b_i \cdot AB} \cdot t_i + \frac{N_i' \cdot (N_i' - 1) \cdot K_2 \cdot P_i^2 \cdot AB \cdot CP}{N_i' \cdot K_1 \cdot P_i \cdot AB} \cdot t_i \right) = \quad (2.39) \\ &= AB \cdot CP \cdot \frac{N_i' \cdot b_i}{2 \cdot t_i + t_{RB_i}} \cdot (N_{RB_i} \cdot a_{RB_i} \cdot t_{RB_i} + N_i' \cdot a_i \cdot t_i + (N_i' - 1) \cdot K_2 \cdot P_i \cdot t_i) = (K_{t_i} = t_{RB_i} / t_i) \\ &= AB \cdot CP \cdot \frac{N_i' \cdot b_i}{2 + K_{t_i}} \cdot (N_{RB_i} \cdot a_{RB_i} \cdot K_{t_i} + N_i' \cdot a_i + (N_i' - 1) \cdot K_2 \cdot P_i) \end{aligned}$$

donde K_{t_i} es la relación entre el tiempo de procesamiento del constructor y el del nivel, y $K_{t_E} = 0$ ya que no existe constructor del nivel E al E+1.

Obsérvese que se ha considerado, igual que en el caso sin regiones de interés, que el procesamiento y la fusión operan en modo pipeline.

El factor de calidad requerido para el sistema jerárquico con regiones de interés lo obtenemos a partir de la expresión anterior, y queda:

$$Q_{H_{Rol}} = AB \cdot CP \cdot N_1 \cdot b_1 \cdot \sum_{i=1}^E \frac{(N_i' \cdot a_i + (N_i' - 1) \cdot K_2 \cdot P_i + N_{RB_i} \cdot a_{RB_i} \cdot K_{t_i})}{(2 + K_{t_i}) \cdot P_i \cdot \sum_{i=1}^E 1/P_i} \quad (2.40)$$

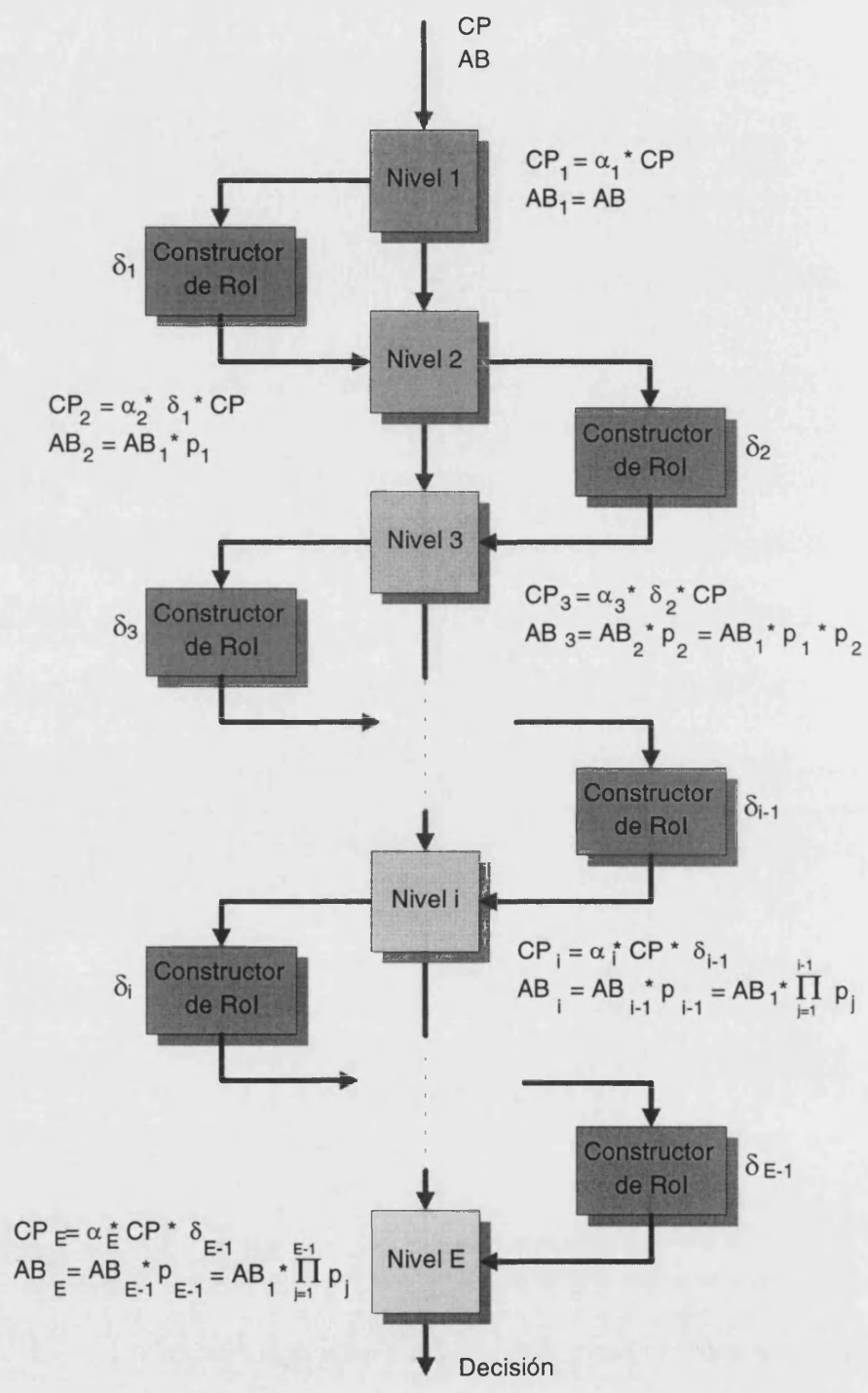


FIGURA 2.28 Procesado jerárquico con regiones de interés.

Podemos ahora comparar los factores de calidad requerida para los sistemas jerárquicos con y sin regiones de interés. Estamos interesados en ver cuándo el sistema con regiones de interés tiene un factor de calidad requerida menor que el sistema sin región de interés.

Comparando la ecuación (2.30) con la (2.40) para un determinado nivel j , puesto que si se cumple para todos los niveles, se cumplirá para la suma, tenemos:

$$Q_{H_{Rol}} < Q_H \Rightarrow \Rightarrow \frac{N'_j \cdot a_j + N_{RB_j} \cdot a_{RB_j} \cdot K_{t_j} + (N'_j - 1) \cdot K_2 \cdot P_j}{(2 + K_{t_j}) P_j \cdot \sum_{s=1}^E 1/P_s} < \frac{N_j \cdot a_j + (N_j - 1) \cdot K_2 \cdot P_j}{2 \cdot P_j \cdot \sum_{s=1}^E 1/P_s} \quad (2.41)$$

Si $K_{t_j} \rightarrow 0$, es decir, el constructor emplea muy poco tiempo en identificar los canales de las regiones de interés con relación al tiempo de procesado del nivel correspondiente, entonces podemos simplificar la expresión anterior, quedándonos:

$$N'_j \cdot a_j + (N'_j - 1) \cdot K_2 \cdot P_j < N_j \cdot a_j + (N_j - 1) \cdot K_2 \cdot P_j \quad (2.42)$$

Reordenando

$$(N_j - N'_j) \cdot (a_j + K_2 \cdot P_j) > 0 \quad (2.43)$$

condición que se cumple siempre puesto que a_j , K_2 y P_j son, por definición, no negativos y $N_j > N'_j$ al reducirse el número de canales a procesar en el caso del sistema con regiones de interés. Por lo tanto en este caso, el sistema con regiones de interés tendrá siempre un factor de calidad requerida menor que sin regiones de interés.

En el caso de que $K_{t_j} \rightarrow 0$, podemos despejar K_{t_j} , obteniéndose:

$$K_{t_j} \begin{matrix} < \\ > \end{matrix} \frac{2 \cdot (N_j - N'_j) \cdot (a_j + K_2 \cdot P_j)}{2 \cdot N_{RB_j} \cdot a_{RB_j} - (N_j \cdot a_j + (N_j - 1) \cdot K_2 \cdot P_j)} \quad (2.44)$$

donde el sentido de la desigualdad depende de si el denominador de la expresión es positivo (desigualdad menor que) o negativo (desigualdad mayor que). En el caso de que el denominador sea negativo, la condición de "mayor que" se cumple siempre ya que, como $N_j > N'_j$, la expresión a la derecha de la desigualdad será negativa y, por definición, $K_{t_j} > 0$, por lo que el sistema con regiones de interés tendrá un factor de calidad requerida menor que el sistema sin regiones de interés.

Podemos ahora, tomando el denominador, evaluar cuándo es positivo o negativo. Si así lo hacemos tenemos que:

$$N_{RB_j} \cdot a_{RB_j} \begin{matrix} < \\ > \end{matrix} \frac{N_j \cdot a_j + (N_j - 1) \cdot K_2 \cdot P_j}{2} \quad (2.45)$$

Si nos fijamos, la expresión a la izquierda de la desigualdad no es más que el cociente entre la capacidad de procesado total del constructor y la total del problema.

De igual manera, a la derecha de la expresión lo que se tiene es la mitad de la suma de las capacidades relativas a la total de los procesadores del nivel y del fusionador.

Por lo tanto, si la capacidad relativa del constructor es menor que la mitad de la del nivel (considerando los procesadores y el fusionador), el denominador será negativo y se cumplirá que el sistema con regiones de interés tiene un factor de calidad requerida menor. Si, por el contrario, la capacidad relativa del constructor es mayor que la mitad de la del nivel, dependerá del valor de K_t , el que el factor de calidad requerida del sistema jerárquico sea menor o mayor que el del sistema paralelo.

Si el número de unidades del constructor está limitado por la capacidad de procesado de cada una de ellas, la ecuación (2.45) se transforma para ofrecernos el valor del factor γ correspondiente a la capacidad de procesado del constructor con relación a la total del problema:

$$\gamma_j \begin{matrix} < \\ > \end{matrix} \frac{N_j \cdot a_j + (N_j - 1) \cdot K_2 \cdot P_j}{2} \quad (2.46)$$

Si el número de unidades del constructor está limitado por el ancho de banda lo que obtenemos es la relación entre la capacidad de procesado relativa y el ancho de banda relativo de cada unidad que forma el constructor:

$$\frac{a_{RB_j}}{b_{RB_j}} \begin{matrix} < \\ > \end{matrix} \frac{1}{P_j} \cdot \frac{N_j \cdot a_j + (N_j - 1) \cdot K_2 \cdot P_j}{2} \quad (2.47)$$

Las tres ecuaciones (2.45), (2.46) y (2.47) son equivalentes y los valores obtenidos permiten acudir a la ecuación (2.44) y obtener el nuevo número de unidades de cada nivel a partir del valor de K_t , que puede ser evaluado una vez se conoce la capacidad de procesado de las unidades del constructor. El nuevo valor para el número de unidades del nivel se relaciona con el factor de reducción debido al empleo de regiones de interés mediante la ecuación (2.38).

Con esto queda demostrado que bajo ciertas condiciones es posible encontrar un sistema jerárquico con regiones de interés con unos requerimientos de calidad menores que el sistema jerárquico sin ellas.

5.2.2 Agrupamiento de datos

Otra de las mejoras que se pueden introducir en el sistema es intentar evitar la dispersión de los datos a procesar en cada elemento de procesado de cada nivel en la jerarquía. La utilización de regiones de interés implica que todos los canales de cada una de ellas deberán ser procesados de manera conjunta puesto que la elaboración de la decisión se hará sobre la base de las relaciones existentes entre los valores de los canales.

Si implementamos los niveles de la jerarquía como sistemas paralelos, puede ocurrir que los canales de la región de interés queden repartidos en varios procesadores, haciendo necesaria la intercomunicación entre los mismos, aumentando el tiempo necesario para el procesado y disminuyendo las prestaciones del sistema.

Una solución a este problema es intentar agrupar los datos de forma que se pueda maximizar la probabilidad de que todos los canales de una región de interés se envíen a una misma unidad de procesado del sistema paralelo dentro del nivel.

Para realizar esto se puede estudiar el problema para averiguar los canales que con una mayor probabilidad van a formar parte de una región de interés. Esto supone que el proceso físico tiene un cierto sesgo y que se produce de manera más probable en unos ciertos subespacios del espacio de medida.

Si esto no es así, entonces existen dos opciones:

- Emplear la información de las regiones de interés para realizar un rutado dinámico de los datos. Esta solución es óptima pero implica la realización del multiplexaje de los canales y el empleo de retardos para disponer de la información de la región de interés antes de rutar los canales.
- Agrupar los canales de forma estática, pero siguiendo alguna relación con el fenómeno físico que se está observando. Esta solución implica que en ocasiones se producirá la necesidad de intercambiar datos entre los procesadores, pero el retardo introducido será menor que en el caso de rutar dinámicamente.

El primer caso es el estudiado anteriormente en el que se ha supuesto que todos los datos estaban en cada procesador y no hacía falta una comunicación entre ellos. En el segundo caso, el hecho de que se deban compartir datos producirá retardos en el procesado. Si se quiere mantener el tiempo global de procesado en el nivel, el efecto del retardo es la necesidad de aumentar bien la capacidad de procesado de las unidades empleadas o su número si se mantiene constante la capacidad.

Tomando como referencia nuestro sistema jerárquico con regiones de interés, la capacidad de procesado total CP_i que ha de cubrirse con un sistema paralelo en el nivel i de la jerarquía se puede expresar como:

$$CP_i = \delta_{i-1} \cdot \alpha_i \cdot \frac{D \cdot op}{t_i} \quad 1 \leq i \leq E \quad (2.48)$$

donde D es el número de datos a procesar, op es el número de operaciones a realizar por dato, t_i es el tiempo en el que se ha de realizar el trabajo en el nivel i y δ el factor de reducción debido al empleo de regiones de interés.

Para cada unidad de este sistema paralelo con N_i unidades, la capacidad de procesado es:

$$CP_{ij} = \delta_{i-1} \cdot \alpha_i \cdot \frac{D/N_i \cdot op}{t_i} \quad 1 \leq j \leq N_i \quad 1 \leq i \leq E \quad (2.49)$$

Si ahora existe la posibilidad de que haya que intercambiar datos, variará el número de operaciones a realizar en función de la probabilidad de que haya que realizar ese intercambio. Sin embargo, si queremos mantener el número de unidades, deberá aumentarse la capacidad de procesado de cada una de ellas. La relación entre la nueva capacidad, CP'_{ij} , y la antigua es:

$$CP'_{ij} = \delta_{i-1} \cdot \alpha_i \cdot \frac{D/N_i \cdot op + p_{ai} \cdot D/N_i \cdot op_a}{t_i} = CP_{ij} \cdot \left[1 + p_{ai} \frac{op_a}{op} \right] \quad \begin{matrix} 1 \leq j \leq N \\ 1 \leq i \leq E \end{matrix} \quad (2.50)$$

donde op_a es el número de operaciones a realizar para llevar a cabo el acceso para tomar datos de otras unidades del nivel i y p_a es la probabilidad de tener que efectuar este acceso; para simplificar el cálculo se supone igual para todas las unidades.

Como se observa, el incremento en la capacidad es proporcional al cociente entre el número de operaciones totales realizadas al acceder a otra unidad y el número de operaciones cuando no requieren acceso.

Si lo que hacemos es mantener la capacidad de procesado de las unidades, entonces deberemos incrementar su número. Para calcular dicho incremento hacemos:

$$CP_{ij} = \delta_{i-1} \cdot \alpha_i \cdot \frac{D/N_i^{(cp)} \cdot op + p_{ai} \cdot D/N_i^{(cp)} \cdot op_a}{t_i} = \delta_{i-1} \cdot \alpha_i \cdot \frac{D \cdot op}{N_i^{(cp)} \cdot t_i} \quad \begin{array}{l} 1 \leq j \leq N_i' \\ 1 \leq i \leq E \end{array} \quad (2.51)$$

Despejando el nuevo número de unidades según la capacidad de procesado, $N_i^{(cp)}$, obtenemos:

$$N_i^{(cp)} = N_i^{(cp)} \cdot \left[1 + p_{ai} \frac{op_a}{op} \right] \quad 1 \leq i \leq E \quad (2.52)$$

donde $N_i^{(cp)} = CP_i / CP_{ij}$.

En cualquiera de los dos casos, el ancho de banda requerido aumenta puesto que ahora no sólo ha de ser suficiente para el transporte de los datos hacia las unidades de procesado, sino que debe dar cabida a las transferencias entre éstas para el intercambio de los datos no disponibles.

Cada unidad deberá disponer de suficiente ancho de banda para permitir no sólo sus transferencias de petición de datos a otras unidades sino las de otras unidades a ella. Para simplificar el problema supondremos que las peticiones entre unidades se reparten de forma uniforme y que la probabilidad de realizar un acceso para pedir datos a otra unidad es la misma para todas ellas. Si esto no fuera así, el análisis se complicaría al tener que introducir la función de distribución de la probabilidad de realizar una petición, así como la del destino de la misma.

En este caso, cada unidad realiza $D \cdot p_a$ accesos repartidos entre las $N-1$ unidades restantes, y recibe de las $N-1$ otras unidades la misma cantidad, donde p_a es la probabilidad de realizar una petición externa. De esta manera, el ancho de banda requerido será:

$$\begin{aligned} AB'_{ij} &= P_i \cdot \frac{D \cdot B / N_i^{(ab)} + p_{ai} \cdot D \cdot B / N_i^{(ab)} + p_{ai} \cdot D \cdot B / N_i^{(ab)}}{t_1} = \\ &= P_i \cdot \frac{D \cdot B / N_i^{(ab)} + 2 \cdot p_{ai} \cdot D \cdot B / N_i^{(ab)}}{t_1} = \frac{D \cdot B \cdot P_i}{t_1 \cdot N_i^{(ab)}} \cdot (1 + 2 \cdot p_{ai}) = \frac{D \cdot B \cdot P_i}{t_1 \cdot N_i^{(ab)}} \quad \begin{array}{l} 1 \leq j \leq N_i^{(ab)} \\ 1 \leq i \leq E \end{array} \end{aligned} \quad (2.53)$$

de donde podemos despejar el nuevo número de unidades según en ancho de banda, $N_i^{(ab)}$:

$$N_i^{(ab)} = N_i^{(ab)} \cdot (1 + 2 \cdot p_{ai}) \quad (2.54)$$

con $N_i^{(ab)} = AB_i / AB_{ij}$.

Por lo tanto, el número de unidades necesarias, N'' , será el mayor de los resultados de las ecuaciones (2.52) y (2.54). Es decir:

$$N_i'' = \max(N_i^{(cp)}, N_i^{(ab)}) \quad (2.55)$$

Así pues, al introducir un agrupamiento de datos estático estamos modificando el número de unidades del sistema y por lo tanto el factor de calidad requerida del mismo.

La nueva expresión del factor de calidad requerida del sistema jerárquico con agrupamiento de datos la obtenemos sin más que substituir en la ecuación (2.40) el número de unidades por el valor resultante de la ecuación (2.55), es decir:

$$Q'_{H_{Rol}} = AB \cdot CP \cdot N_1 \cdot b_1 \cdot \sum_{i=1}^E \frac{(N_i'' \cdot a_i + (N_i'' - 1) \cdot K_2 \cdot P_i + N_{RB_i} \cdot a_{RB_i} \cdot K_{t_i})}{(2 + K_{t_i}) P_i \cdot \sum_{j=1}^E 1/P_j} \quad (2.56)$$

Si además de agrupar los datos, se realizase cualquier tipo de preprocesado (dar formato, detección y corrección de errores, etc.) en los mismos, se introduciría un elemento más en serie con los procesadores, por lo que el factor de calidad requerida aumentaría. Depende del caso en concreto el que esa modificación pudiese suponer que el sistema superase el factor de calidad requerida del caso sin regiones de interés.

5.3 DESALEATORIZADORES

En el sistema jerárquico, la existencia de etapas en las que los datos son filtrados implica que la tasa de entrada no es igual a la tasa de salida, sino que se relacionan por la probabilidad de que el dato pase de una etapa a la siguiente.

En la entrada de la etapa, no se puede predecir cuándo va a llegar un dato de la etapa anterior, sino solo la frecuencia media. Por ello, en el sistema jerárquico, a la entrada de la etapa siempre se sitúa un buffer encargado de absorber la diferencia de velocidades entre los datos que le llegan y el procesado de los mismos.

Un problema a estudiar es cuál debe ser el tamaño de este buffer que desaleatoriza la llegada de los datos. Para resolverlo hay que acudir a la teoría de colas y modelar el sistema como se muestra en la figura 2.29 [34].

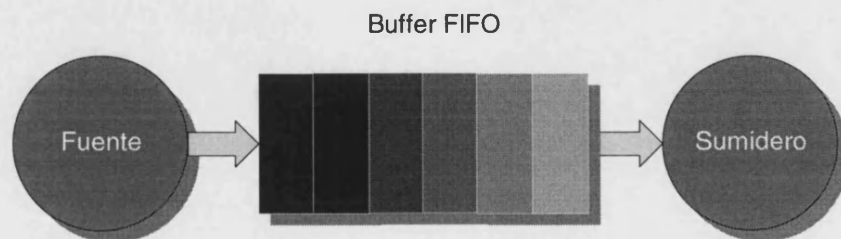


FIGURA 2.29 Modelo de estudio del buffer desaleatorizador.

En este modelo, la fuente transfiere datos al buffer de forma que la media sea constante cuando se considera un periodo de tiempo suficientemente largo, por lo que si el buffer tiene n posiciones, la probabilidad media de que haya i elementos almacenados en el buffer no depende del tiempo. Llamaremos P_0 a la probabilidad de que el buffer esté vacío y P_q con $q \leq n$ a la probabilidad de que haya q elementos en el buffer uno de los cuales se transfiere al sumidero. El sumidero toma un dato del buffer cada cierto tiempo τ .

Para determinar las probabilidades consideramos dos instantes separados por una diferencia de tiempo igual al tiempo que tarda el sumidero en leer un nuevo dato. La probabilidad f_k de que durante este intervalo de tiempo lleguen k datos viene dada por la distribución de Poisson [35]:

$$f_k = \frac{(\lambda\tau)^k e^{-\lambda\tau}}{k!} \quad (2.57)$$

donde λ es la tasa media de llegada de datos.

Como las probabilidades no dependen del tiempo, tenemos las siguientes relaciones:

$$\begin{aligned} P_0 &= P_0 f_0 + P_1 f_0 \\ P_1 &= P_0 f_1 + P_1 f_1 + P_2 f_0 \\ P_2 &= P_0 f_2 + P_1 f_2 + P_2 f_1 + P_3 f_0 \\ &\dots \end{aligned} \quad (2.58)$$

es decir, por ejemplo, la probabilidad de que el buffer esté vacío es igual a la probabilidad de que esté vacío y no llegue ningún dato más la probabilidad de que haya un dato, que será retirado por el sumidero, y no llegue ninguno más, con lo que después de que el sumidero lea el dato, el buffer estará vacío.

De estas relaciones obtenemos:

$$\begin{aligned} P_1 &= (P_0 - P_0 f_0) / f_0 \\ P_2 &= (P_1 - P_0 f_1 - P_1 f_1) / f_0 \\ P_3 &= (P_2 - P_0 f_2 - P_1 f_2 - P_2 f_1) / f_0 \\ &\dots \end{aligned} \quad (2.59)$$

En general:

$$\left\{ \begin{aligned} P_1 &= (P_0 - P_0 f_0) / f_0 \\ & (P_{i-1} - P_0 f_{i-1} - \sum_{j=1}^{i-1} P_{i-j} f_j) \\ P_i &= \frac{\quad}{f_0} \quad \text{para } 2 \leq i \leq n \end{aligned} \right. \quad (2.60)$$

La tasa de salida, cuando hay algo almacenado en el buffer, viene determinada por la frecuencia con que el sumidero toma los datos. Así pues:

$$tasa_{sal} = (1 - P_0) \cdot 1/\tau \quad (2.61)$$

y si definimos:

$$x = \text{tasa}_{ent} / (1/\tau) \quad (2.62)$$

obtenemos que:

$$\text{tasa}_{sal} / \text{tasa}_{ent} = (1 - P_0) / x \quad (2.63)$$

Puesto que la suma de todas las probabilidades es 1, podemos reescribir la ecuación (2.63) de la siguiente manera:

$$\text{tasa}_{sal} / \text{tasa}_{ent} = \frac{1 - P_0 / \sum_{j=0}^n P_j}{x} \quad (2.64)$$

Esta relación puede calcularse fácilmente a partir de las relaciones P/P_0 obtenidas de la ecuación (2.60).

Puesto que el buffer tiene un tamaño limitado de n datos, existe la posibilidad de que éste se encuentre lleno cuando llega un nuevo dato, con lo que se perdería. La probabilidad de que esto suceda puede obtenerse a partir de la relación (2.64), sin más que observar que la tasa de datos perdidos es:

$$\begin{aligned} \text{tasa}_{perd} &= \text{tasa}_{ent} - \text{tasa}_{sal} \Rightarrow \\ \Rightarrow P_{perd} &= \text{tasa}_{perd} / \text{tasa}_{ent} = 1 - \text{tasa}_{sal} / \text{tasa}_{ent} \end{aligned} \quad (2.65)$$

Como la probabilidad de pérdida está en función de la longitud del buffer y de las tasas de entrada y salida de datos, es posible diseñar el buffer adecuado para tener un determinado porcentaje de pérdidas.

De esta manera se obtiene una regla de diseño para los buffer desaleatorizadores necesarios si se desea sincronizar el sistema jerárquico. En la bibliografía se puede encontrar la resolución de este mismo problema cuando la distribución de llegada de los datos no es de Poisson [35]. Si, por algún motivo, esta distribución no fuese fácilmente expresable analíticamente, se puede acudir a la simulación discreta del sistema mediante alguno de los lenguajes actualmente existentes como ModSim [36].

6. BIBLIOGRAFÍA

- [1] Wilkinson, B. *Computer Architecture. Design and performance*. Prentice Hall, 1991.
- [2] Flynn, M. J. *Very high speed computing systems*. Proc. IEEE, vol. 54, págs. 1901-1909, 1966.
- [3] Feng, T.-Y. *Some characteristics of associative/parallel processing*. Proc. Sagamore Computing Conf., págs. 5-16, 1972.
- [4] Händler, W. *The impact of classification schemes on computer architecture*. Proc. Int. Conf. on Parallel Processing, págs. 7-15, 1977.

- [5] Reddi, S. S. y Feurstel E. A. *A conceptual framework for computer architecture*. Computing Surveys, n°2, págs. 277-300, 1976.
- [6] Skillicorn, D. B. *A taxonomy for computer architectures*. IEEE Computer, n° 11, págs. 46-57, 1988.
- [7] Dasgupta, S. *A hierarchical taxonomic system for computer architectures*. IEEE Computer, n° 3, págs. 64-74, 1990.
- [8] Varma, A. y Raghavendra, C. S. *Interconnection networks for multiprocessors and multicomputers. Theory and practice*. IEEE Computer Society Press, 1994.
- [9] Hwang, K. *Advanced Computer Architecture*. McGraw-Hill, 1993.
- [10] Hwang, K. y Briggs, F. A. *Arquitectura de procesadores y procesamiento paralelo*. McGraw-Hill, 1984.
- [11] Minsky, M. y Papert, S. "On some associative, parallel and analog computations", en E. J. Jacks, *Associative Information Technologies*. Elsevier North Holland, 1971.
- [12] Kuck, D. J. et al. *The effects of program restructuring, algorithm change and architecture choice on program parallelism*. Proc. Int. Conf. Parallel Processing, págs. 129-138, 1984.
- [13] Amdahl, G. M. *Validity of the single-processor approach to achieving large scale computing capabilities*. AFIPS Conference Proceedings, vol. 30, págs 483-485. AFIPS Press, 1967.
- [14] Gustavson, J. L. *Re-evaluating Amdahl Law*. Communications of the ACM, vol 32, n° 5, mayo 1988, págs. 532-535. Association for Computing Machinery, 1988.
- [15] Enslow, P. H. *Multiprocessor and parallel processing*. Wiley-Interscience, Nueva York, 1974.
- [16] IEEE. *VMEbus Specification 1014-1987*.
- [17] IEEE. *Futurebus+: Logical Layer Specification, 896.1-1991*. EEUU, 1991.
- [18] Fujitsu America, Inc. *VPP500 vector parallel processor*. EEUU, 1992.
- [19] Cray Research, Inc. *Cray Y/MP functional description manual*. EEUU, 1989.
- [20] Sperry Univac. *The Univac series 1100 announcement*. 1982.
- [21] Pfister, G. F. et al. *The IBM research parallel processor prototype (RP3): introduction and architecture*. Proceedings International Conference on Parallel Processing, págs. 764-777, 1985.
- [22] Gottlieb, A. et al. *The NYU Ultracomputer-Designing an MIMD shared memory parallel computer*. IEEE Trans. Computers, C-32(2), págs. 175-189, 1983.
- [23] IEEE. *SCI, Scalable Coherent Interface, Std. 1596-1992*. EEUU, 1992.
- [24] Gustavson, D., Li Q. *Local-area multiprocessor: the Scalable Coherent Interface*. SCIZZL, Universidad de Santa Clara, EEUU.
- [25] Mora, F. *Aplicación de SCI en la adquisición de datos del detector ATLAS*. Tesis doctoral. Universidad Politécnica de Valencia, 1997.
- [26] IEEE. *Control State Registers, Std. 1212-1991*. EEUU, 1991.
- [27] Dolphin Interconnect Solutions, Inc. *A backside link (B-Link) for Scalable Coherent Interface (SCI) Nodes*. Junio, 1995.
- [28] Handel, R. et al. *ATM Networks: concepts, protocol, applications*.
- [29] Cuenca, P. *Redes de interconexión en sistemas distribuidos con tecnología ATM*. Tesis de Licenciatura, Universitat de València, 1996.

- [30] Fahmy, S. *A survey of ATM switching techniques.*
- [31] X3T9.3 Task Group de ANSI. *Fibre Channel physical and signalling interface (FC-PH) Rev. 4.2*, 1993.
- [32] Fibre Channel Association, *Fibre Channel: Connection to the future*, ISBN 1-878707-19-1, 1994.
- [33] Nagin, P. A. et al. "*Region relaxation in a parallel hierarchical architecture*", en Onoe, M. et al., *Real-Time parallel computing image analysis*, págs. 37-61. Plenum Press, Nueva York, 1981.
- [34] Øverås, H. *Dead-time losses in a buffered data recording system.* NIM n°104, págs. 85-91, 1972.
- [35] Cooper, R. B. *Introduction to queueing theory.* The Macmillan Company, EEUU, 1972.
- [36] CACI Company, *ModSim II The language for object oriented programming.* EEUU, 1993

CAPÍTULO

3

APLICACIÓN DE TÉCNICAS DE REGIONES DE INTERÉS Y AGRUPAMIENTO DE DATOS AL EXPERIMENTO ATLAS/LHC DEL CERN

1. INTRODUCCIÓN	1
2. EL EXPERIMENTO ATLAS/LHC	1
2.1 Descripción	3
2.2 Requerimientos de la aplicación. Posibles soluciones	5
2.2.1 Solución paralela	6
2.2.2 Solución jerárquica	7
2.2.2.1 El primer nivel	8
2.2.2.2 El segundo nivel	10
2.2.2.3 El tercer nivel	11
2.2.2.4 Evaluación de la arquitectura jerárquica	13
3. INCORPORACIÓN DE REGIONES DE INTERÉS Y AGRUPACIÓN DE DATOS	13
3.1 Regiones de interés (RoI)	14
3.2 Agrupamiento de datos (ROD)	18
3.3 Opciones tecnológicas para la implementación de los niveles	21
3.3.1 El primer nivel	21
3.3.2 El segundo nivel	22
3.3.3 El tercer nivel	25
4. BIBLIOGRAFÍA	26

1. INTRODUCCIÓN

En los capítulos anteriores se ha realizado una exposición de los conceptos básicos de la integración de sensores y de las arquitecturas para la adquisición de los datos procedentes de los mismos. En el capítulo 2 se propuso una mejora sobre las arquitecturas jerárquicas empleadas cuando el volumen de información a adquirir es muy elevado. Dicha mejora, basada en la inclusión de Regiones de Interés, permitía reducir el factor de calidad requerida del sistema.

En este capítulo se va a estudiar la introducción de esta mejora a una aplicación donde tradicionalmente se han venido aplicando arquitecturas jerárquicas para la adquisición de los datos. Esta aplicación se sitúa dentro del campo de la experimentación de física de altas energías. Se introducirá primero las características más generales de este campo para luego estudiar la introducción de la mejora planteada al experimento ATLAS/LHC que se prepara en las instalaciones del Laboratorio Europeo de Física de Partículas (CERN) en Ginebra.

2. EL EXPERIMENTO ATLAS/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 tiene carga eléctrica entera y no experimentan todas las fuerzas. La tabla 3.1 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 3.1 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 3.2 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 en el que actúa.

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

TABLA 3.2 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. 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 de 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*.

El problema de esta producción artificial es que generalmente las partículas buscadas tienen probabilidades de producirse extremadamente bajas, o, como se conoce en este campo, una baja *sección eficaz* de producción [1] al no poder recrear exactamente las mismas condiciones de densidad de energía. La sección eficaz es un concepto simbólico acerca de la probabilidad de que las áreas espaciales ocupadas por dos partículas o por una partícula y un núcleo de material, se intersecten en una colisión. Si, por ejemplo, un haz de neutrones incide sobre 1 cm^3 de material, la sección eficaz efectiva del blanco será $n\sigma$, siendo n el número de átomos por cm^3 y σ la

probabilidad de interacción entre un neutrón y un núcleo (sección eficaz microscópica).

Puesto que la sección eficaz de producción es extremadamente baja para algunas partículas, en estos experimentos se intenta que se produzca el mayor número de colisiones por segundo con la ayuda de una frecuencia de colisión y una luminosidad elevadas.

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 la 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.

Para la lectura del resultado de cada colisión se emplean grandes detectores de partículas, compuestos de hasta centenares de millones de sensores y canales electrónicos. Por su estructura, este tipo de sistemas de sensores pueden clasificarse como redes de sensores distribuidos espacialmente. La organización de los sensores dentro de cada subdetector es específica en función de la aplicación concreta para la que se haya diseñado.

La razón del gran número de canales es la necesaria segmentación física del detector para obtener informaciones más precisas, espacialmente hablando, de las variables a sensar, así como de reducir la cantidad de información a adquirir de cada canal. Los canales electrónicos van asociados a los diferentes subdetectores que componen el detector, cada uno de los cuales se encarga de medir una característica destinada a la identificación de las partículas resultado de la colisión. Sin embargo, también debido a esa segmentación, el volumen de información a leer en cada colisión puede llegar a ser realmente elevado (del orden de algunos megabytes).

2.1 DESCRIPCIÓN

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 3.2); 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].

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 50.000 millones de pesetas).

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 3.1, 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.

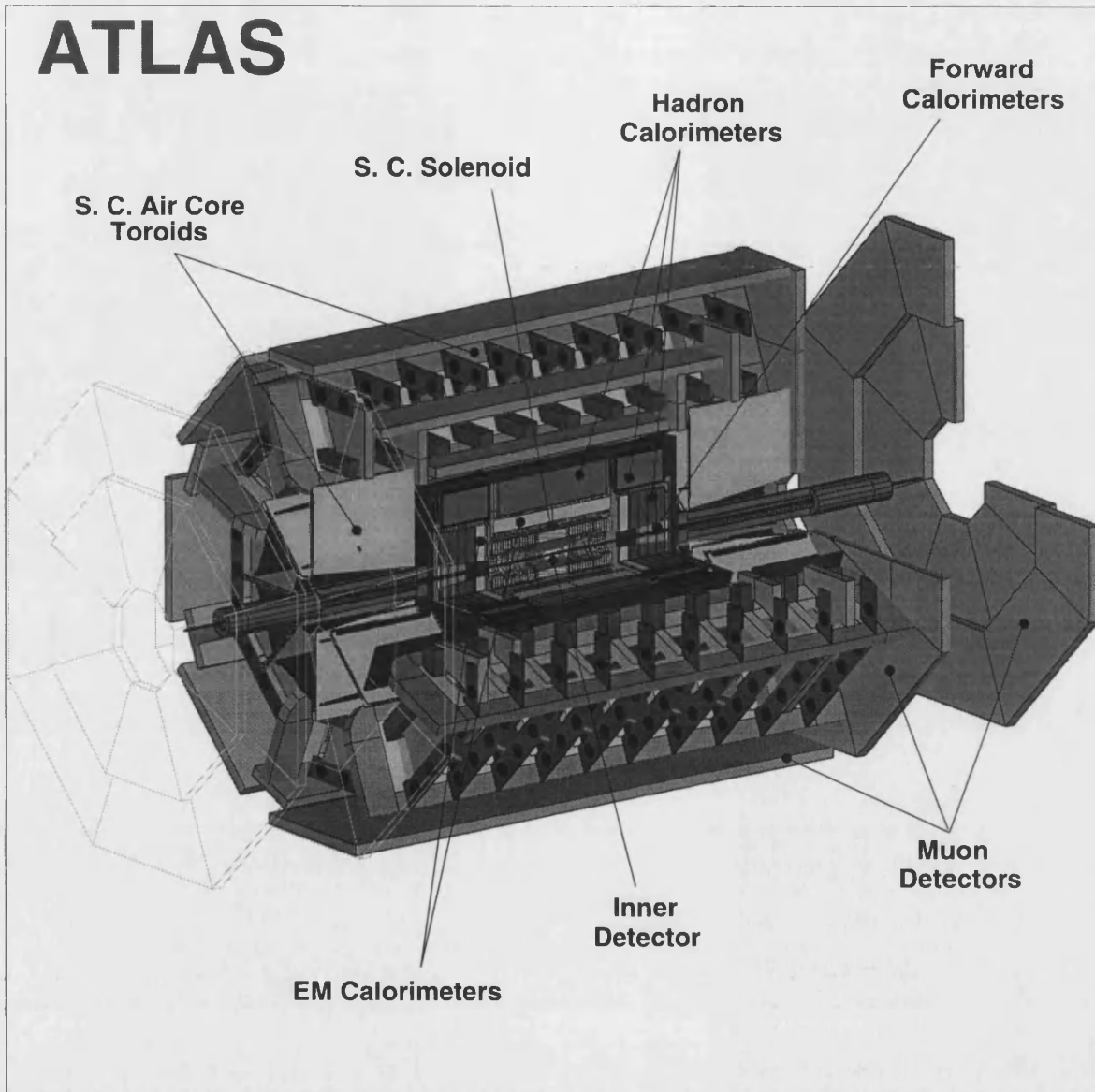


FIGURA 3.1 El detector ATLAS.

2.2 REQUERIMIENTOS DE LA APLICACIÓN. POSIBLES SOLUCIONES

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. Pese a ese gran número, sólo existen un cierto número limitado de tipos de sensores diferentes. Se trata, como ya vimos en el capítulo 1, de un sistema de sensores físicamente replicados para cubrir el espacio a analizar.

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 3.3 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	140·10 ⁶
	Silicon Strips	2'9·10 ⁶
	GaAs	0'8·10 ⁶
	TRT	0'42·10 ⁶
Muones		200
	MDT	300.000
	CSC	100.000
	RPC	400.000
	TGC	490.000
TOTAL	147.144.100	1280

TABLA 3.3 Canales electrónicos en cada subdetector y sus partes para el experimento ATLAS/LHC [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 PowerPC posee una capacidad de procesado de unos 100 MIPS, lo que implicaría que sería necesario un sistema con ¡500 millones de procesadores! para poder tratar el problema, como veremos en el siguiente apartado).

2.2.1 Solución paralela

La solución paralela, figura 3.2, consistiría en la implementación de un sistema con un determinado número de procesadores y un fusionador para obtener la decisión acerca del suceso.

Se podría estimar el número de procesadores tomando como una unidad de procesado la de un dispositivo de 100 MIPS y 200 Mbytes/s. La razón de estos valores no es otra que la de obtener resultados con dispositivos actualmente disponibles. En el futuro, segura que la tecnología incrementará las prestaciones de estos dispositivos, mejorando las prestaciones del sistema. El número de procesadores necesario sería el máximo de los obtenidos al dividir la capacidad de procesado total y el ancho de banda total entre los respectivos valores de la unidad de procesado, ecuación 2.16, es decir:

$$N = \max\left(\frac{CP_T}{100}, \frac{AB_T}{200}\right) = \left(\frac{5 \cdot 10^{10}}{100}, \frac{5 \cdot 10^7}{200}\right) = 5 \cdot 10^8 \quad (3.1)$$

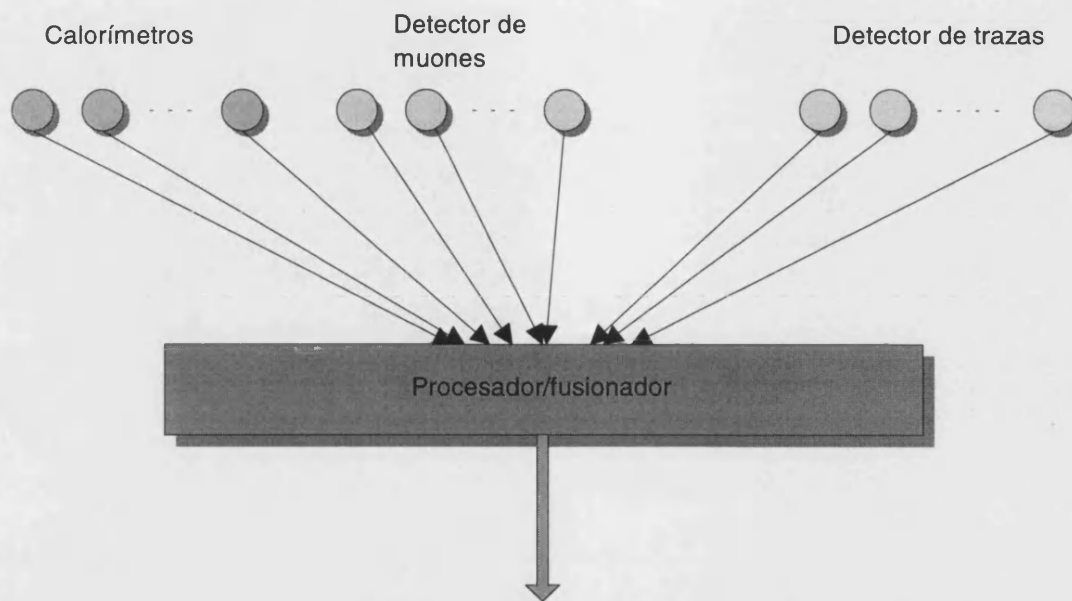


FIGURA 3.2 Integración de sensores en ATLAS/LHC mediante una solución paralela.

Para el cálculo del factor de calidad requerida hemos de estimar también el valor del factor K . En nuestro caso, el tamaño medio del dato es de 2 bytes correspondientes a convertidores A/D de 16 bits cuyo rango se ajusta a los valores de señal a leer con la sensibilidad adecuada; por otro lado, el tamaño de la decisión de cada unidad de procesado será menor, digamos de 1 byte, en el que 1 bit indicará la decisión y el resto indicarán la unidad de procesado e información de control. El detector dispone de unos 150 millones de canales electrónicos a leer (tabla 3.3) y hay que realizar unas 10^9 operaciones por canal para la obtención del resultado final, estimadas según los algoritmos que han de ejecutarse [6]. Con esos valores, $K_1 = 3'33 \cdot 10^9$ y $K_2 = 6'67 \cdot 10^{18}$.

Teniendo en cuenta que las relaciones entre la capacidad de procesado y el ancho de banda de cada unidad y los totales del problema son: $a = 100/5 \cdot 10^{10} = 2 \cdot 10^9$ y $b = 200/5 \cdot 10^7 = 4 \cdot 10^6$ y sustituyendo en 2.19, el valor del factor de calidad requerida resulta $2'5 \cdot 10^{21}$ MIPS x Mbytes/s. El hecho de que sea mayor que el factor de calidad requerida del problema se debe a la introducción del fusionador, que incrementa la capacidad de procesado necesaria.

2.2.2 Solución jerárquica

Como ya se ha visto, la implementación del 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 3.3. A cada uno de estos niveles se le conoce como *nivel de trigger*.

Veamos a continuación con algo más de profundidad una pequeña descripción de cada uno de ellos y sus principales características.

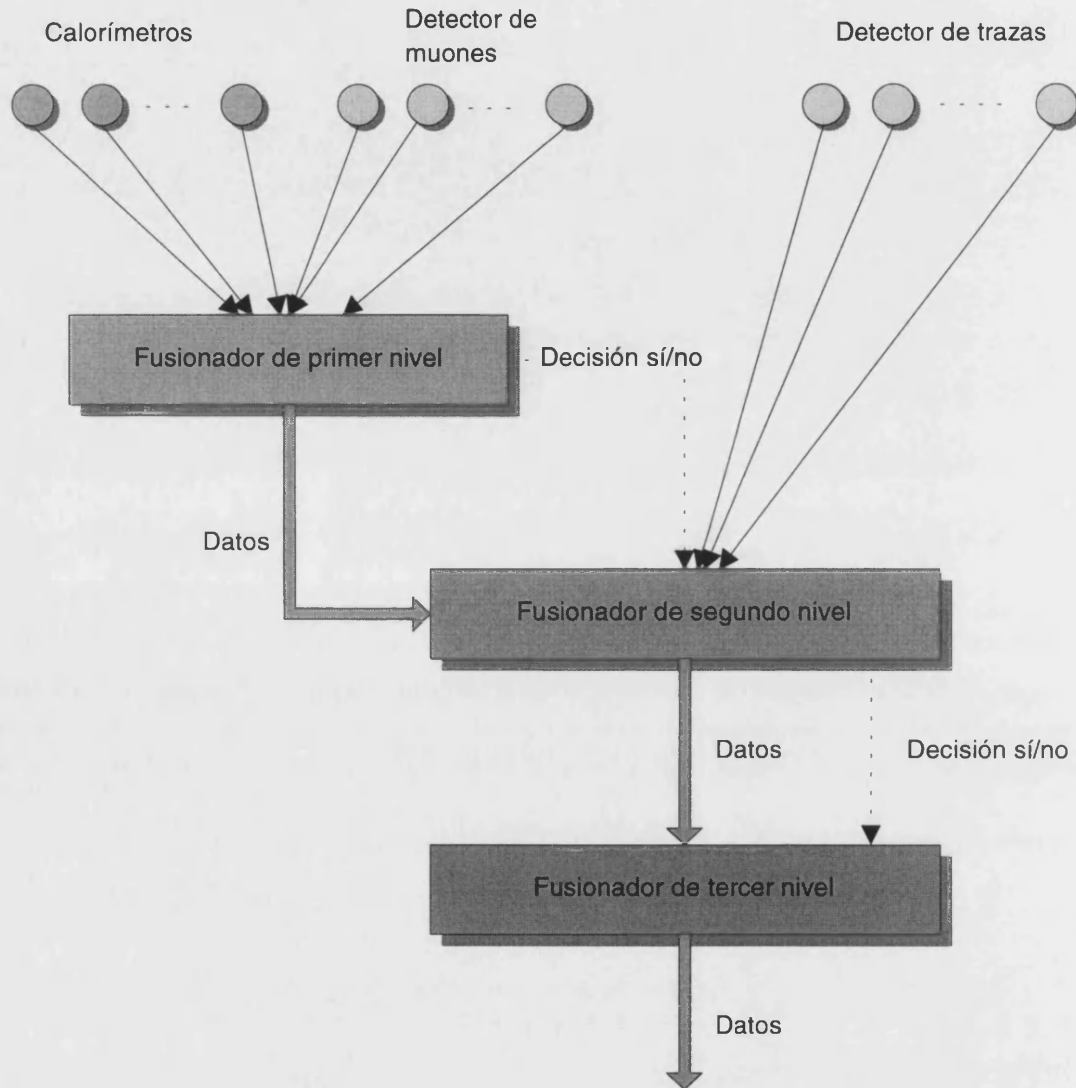


FIGURA 3.3 La arquitectura de selección de tres niveles de ATLAS/LHC.

2.2.2.1 El primer nivel

El primer nivel de trigger aceptará los datos a la frecuencia de colisión del LHC (40 MHz) y realizará una decisión "sí/no" cada $2 \mu\text{s}$ sobre un volumen de datos de unos 380 Kbytes correspondientes sólo al subdetector de muones y a los calorímetros (ver tabla 3.3) 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 procesamiento necesaria en este nivel se estima en unos $45 \cdot 10^6$ MIPS.

La figura 3.4 muestra el diagrama de bloques del nivel 1. En este nivel sólo se realiza el procesamiento 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.

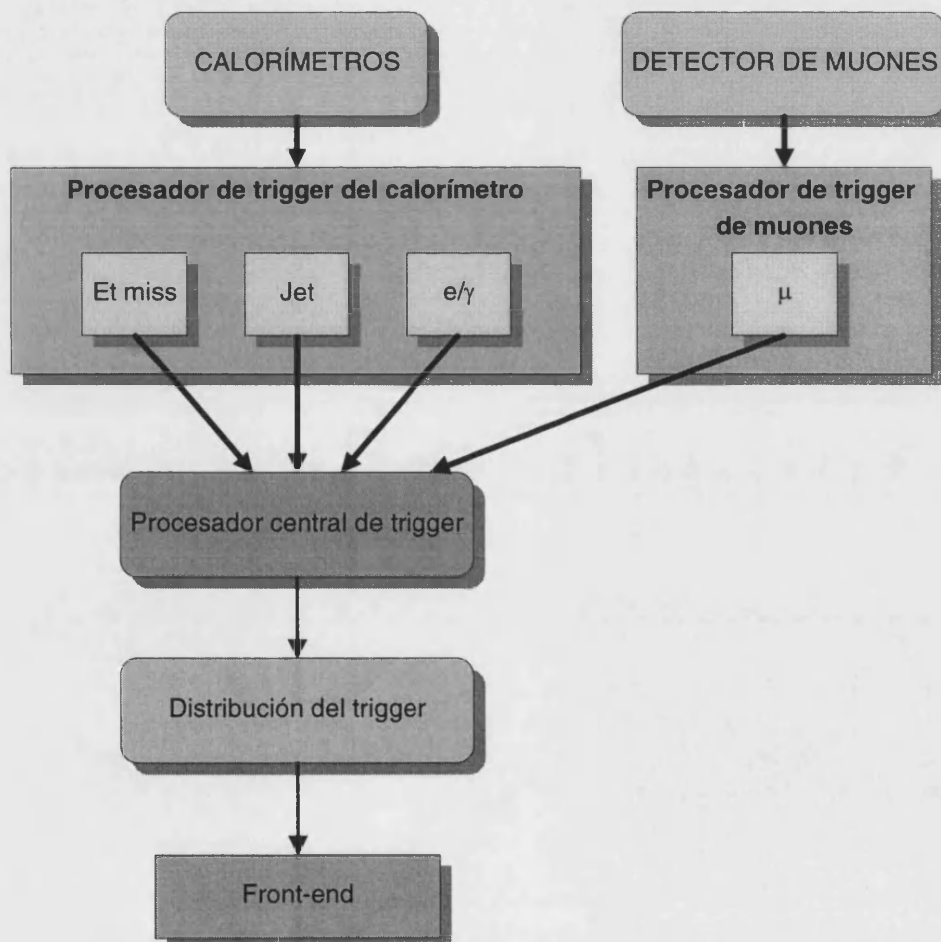


FIGURA 3.4 Diagrama de bloques del primer nivel de trigger.

2.2.2.2 El segundo nivel

El segundo nivel trabaja ya con la información de todos los subdetectores y realiza un procesado más detallado sobre los mismos. 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 [6]. 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.

- **El buffer desaleatorizador**

El tamaño de los buffers desaleatorizadores se podría calcular mediante las expresiones obtenidas en el capítulo 2. Sin embargo, en este caso existe una condición que no se daba en aquel estudio. Para poder dar tiempo a que la señal producida en los detectores en una colisión desaparezca del todo y no se superponga con la de colisiones sucesivas, se ha impuesto la condición de que después de una decisión afirmativa del primer nivel se veten las posibles sucesivas durante 100 ns (4 cruces de haz). Esto hace que el generador de datos no pueda ser modelado estrictamente como uno de Poisson, como se hizo en el capítulo 2, puesto que lo que se desea es generar una secuencia con distribución de Poisson (que simula un nivel de trigger 1 aceptado) y vetar los puntos sucesivos de la secuencia separados por menos de 100 ns. Por ello, se ha realizado el estudio de la longitud del desaleatorizador mediante simulación discreta.

Este tipo de simulación permite conocer la evolución de un sistema en tiempo diferido a partir de un instante inicial, definiendo la duración de cada una de las operaciones que pueden realizarse en él. La simulación, entonces, avanza en intervalos discretos iguales a la duración definida de la operación que se realiza.

Al no disponerse de un paquete comercial para realizar esta simulación, se ha desarrollado una librería de funciones para la simulación discreta en lenguaje C++ que ha permitido estudiar el problema. La figura 3.5 muestra los resultados de la probabilidad de pérdida de datos para diferentes longitudes del buffer y diferentes periodos de lectura de datos de los buffers. Los datos de entrada siguen una distribución de Poisson de media 10 μ s pero incluyendo la condición de que el intervalo entre dos datos sea mayor o igual que 100 ns.

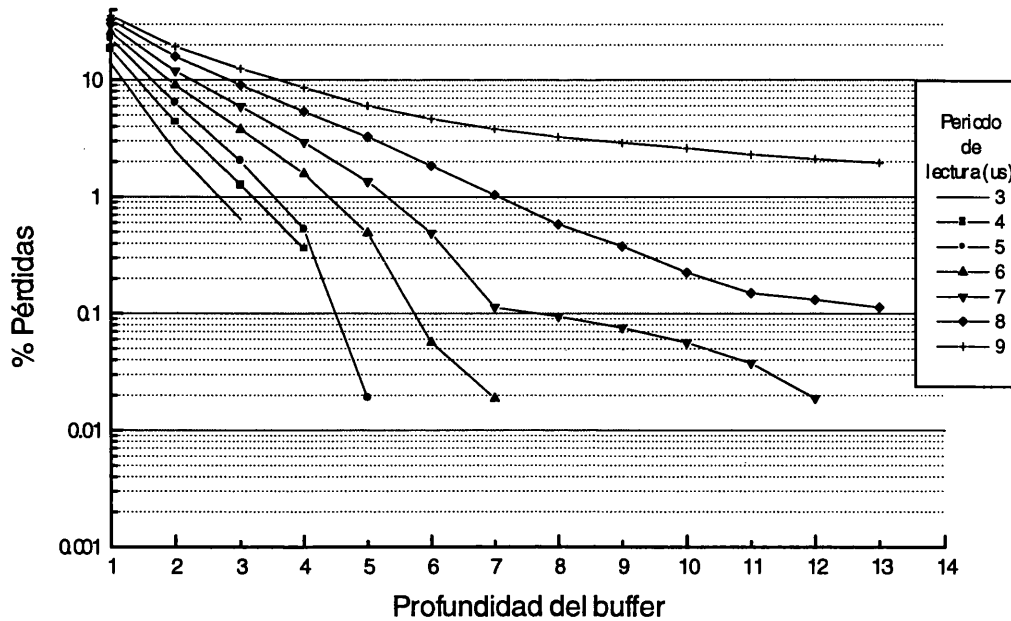


FIGURA 3.5 Porcentaje de pérdidas en función del tamaño y el periodo de lectura para el buffer desaleatorizador.

Como se observa, a mayor periodo de lectura (menor frecuencia de salida de los datos del desaleatorizador), mayor tamaño del buffer para conseguir el 1% de pérdidas estipulado. Los puntos de interrupción de las curvas indican que para el siguiente valor de la profundidad del buffer, el porcentaje de pérdidas es nulo (observar la escala logarítmica en el eje de ordenadas).

Para un periodo de lectura de 9 μ s, el tamaño del buffer para conseguir el 1% de pérdidas es mayor que 13, tamaño que se ha tomado como máximo para los buffers del experimento. Por ello, el periodo de lectura se escoge menor que esta cantidad.

2.2.2.3 El tercer nivel

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 3.6. 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 procesamiento 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 [6]. Ya que se espera una tasa de 1000 sucesos/s, la potencia de procesamiento del tercer nivel está próxima a 10^6 MIPS.

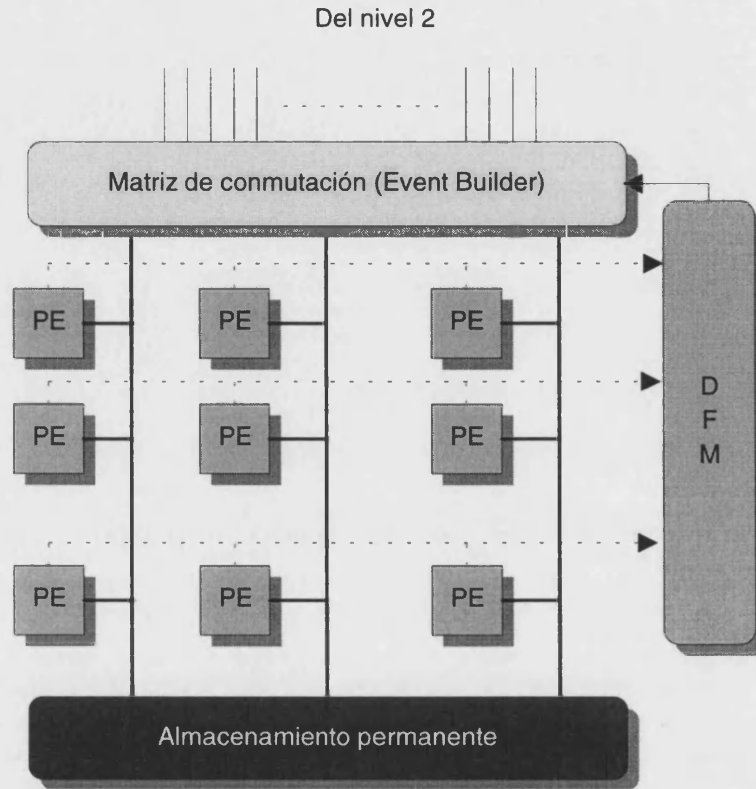


FIGURA 3.6 Diagrama de bloques del tercer nivel de trigger.

El sistema de procesamiento 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 3.7, constará de un segmento de entrada que recibe los datos de cada suceso completo y proporciona la información al DFM, un elemento de procesamiento que ejecuta los algoritmos de análisis y un segmento de salida que recibe los sucesos seleccionados y comprimidos del elemento de procesamiento y los envía hacia el sistema de almacenamiento permanente.

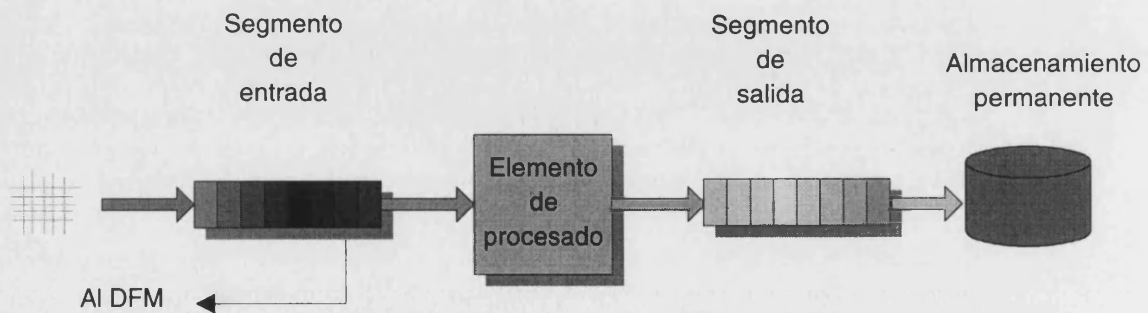


FIGURA 3.7 Esquema de cada una de las máquinas procesadoras del tercer nivel.

2.2.2.4 Evaluación de la arquitectura jerárquica

Veamos a continuación el valor del factor de calidad requerida para el sistema jerárquico tal y como se ha planteado en los epígrafes anteriores.

Los valores de N_i , el número de procesadores de cada nivel, se obtienen de la ecuación 2.23 tomando como referencia unidades de 40 MIPS para el primer nivel y 100 MIPS para el segundo y el tercero. La diferencia está en el hecho de que para el primer nivel se espera utilizar dispositivos hardware tales como FPGAs. El valor de 40 MIPS se considera representativo de la capacidad de procesamiento de estos dispositivos. En todos los casos se supone un ancho de banda de 200 Mbytes/s por ser éste un valor actualmente disponible tanto para los procesadores como para los dispositivos hardware.

Por lo tanto $\alpha_1 = 8 \cdot 10^{-10}$ y $\alpha_2 = \alpha_3 = 2 \cdot 10^{-9}$. Además, puesto que el ancho de banda del primer nivel no es igual al del problema (al procesarse sólo los canales del calorímetro y del subdetector de muones), no podemos suponer que $P_1 = 1$ en el cálculo del número de unidades, sino que en este caso $P_1 = 0.3$ aproximadamente. Sin embargo, para en la expresión del factor de calidad requerida sí hay que tomar $P_1 = 1$ ya que en ella el valor de P_1 aparece en el término del fusionador y hace referencia al tiempo que tiene el mismo para realizar su tarea.

Los valores de K_1 y K_2 empleados son iguales a los del caso paralelo puesto que no depende de la solución sino de características de la aplicación.

Sustituyendo valores en la ecuación 2.27 obtenemos un valor del factor de calidad requerida para cada nivel (tabla 3.4). El valor del factor de calidad requerida del sistema jerárquico de $6 \cdot 10^{18}$. Como se observa el valor es menor que para el caso paralelo. En concreto, el factor de reducción es de unas 400 veces.

Así pues, mediante esta arquitectura jerarquizada en tres niveles se logra reducir la tasa inicial de datos y acomodarla para que se pueda registrar de modo permanente en soporte magnético con un factor de calidad requerida menor que en el caso paralelo.

	AB _{entrada} (Mbytes/s)	CP (MIPS)	p_i	N_i	Q (MB/s x MIPS)
Nivel 1	$14.84 \cdot 10^6$	$45 \cdot 10^6$	1/400	$1.12 \cdot 10^6$	$5 \cdot 10^{15}$
Nivel 2	$10^4 \cdot 10^5$	$32 \cdot 10^6$	1/100	$32 \cdot 10^4$	10^{18}
Nivel 3	$10^3 \cdot 10^4$	10^6	1/10	10^4	$4.5 \cdot 10^{18}$

TABLA 3.4 Principales parámetros de los niveles de trigger.

3. INCORPORACIÓN DE REGIONES DE INTERÉS Y AGRUPACIÓN DE DATOS

Tras el estudio del problema y a la vista del resultado obtenido para el factor de calidad requerida del sistema jerárquico, se empezó a evaluar la posibilidad de reducir la capacidad de procesamiento de alguno de los niveles debido al gran número de procesadores que serían necesarios para su implementación.

Se consideró que el primer nivel, debido a la rapidez con que debía trabajar, se implementaría mediante dispositivos lógicos (FPGA, ASIC, lógica discreta) y no mediante procesadores digitales. De esta manera se lograría la capacidad de procesamiento necesaria a un coste no demasiado elevado (menor que si se implementase mediante procesadores digitales).

Los niveles segundo y tercero, por la naturaleza de las operaciones a realizar, no podían ser implementados mediante lógica sino que debían utilizarse procesadores digitales. El tercero, por disponer de mayor tiempo para el procesado, un ancho de banda requerido no excesivamente elevado, tener que realizar operaciones más orientadas hacia el direccionamiento de datos, disponer de conexiones sencillas hacia periféricos de almacenamiento pasivo y ser utilizado por los usuarios del sistema (con conocimientos básicos acerca del funcionamiento del mismo), podría implementarse sobre máquinas comerciales con interface gráfico, interconectadas entre sí mediante redes de alta velocidad.

Sin embargo, el segundo nivel planteaba el problema de una gran capacidad de cálculo y un gran ancho de banda. En este punto, se planteó la posibilidad del empleo de regiones de interés para descargar al segundo nivel, hacer más sencilla su implementación y mejorar las prestaciones del sistema. Además también se pensó en la posibilidad de agrupar los datos en los procesadores del segundo nivel de forma que se aligerase el procesado.

3.1 REGIONES DE INTERÉS (ROI)

La introducción de regiones de interés en la adquisición de datos del detector implicó la definición de las mismas en términos de los subdetectores que componen ATLAS. El estudio se centró en la determinación de qué subdetectores podrían ofrecer la información sobre los límites de la o las regiones de interés de forma que se asegurase que la información contenida en éstas era útil para esta función.

De estos estudios se concluyó que los subdetectores implicados en la definición de la región de interés serían los calorímetros (tanto el electromagnético como el hadrónico) y el detector de muones.

La utilización de regiones de interés supuso la modificación de la arquitectura del segundo nivel. Esta modificación se materializó en la introducción de dos etapas de procesado: una como nivel local o extracción de características y otra como nivel global.

La extracción de características trabajaría en paralelo para cada subdetector con la información recibida de la/s región/es de interés a estudiar. El resultado de este procesado sería una información con un mayor contenido físico sobre la presencia de determinadas partículas en dicha región.

La decisión global se haría reuniendo las características extraídas y procesándolas adecuadamente.

La estructura del segundo nivel con la introducción de las regiones de interés puede observarse en la figura 3.8. En ella vemos como los datos procedentes del primer nivel se almacenan en los Read Out Buffers o ROBs, de donde los procesadores del nivel local leen los datos correspondientes a las regiones de interés indicadas por el constructor. Una vez procesadas, a través de una red de interconexión, las características (*features*) son enviadas a los procesadores globales para obtener la decisión.

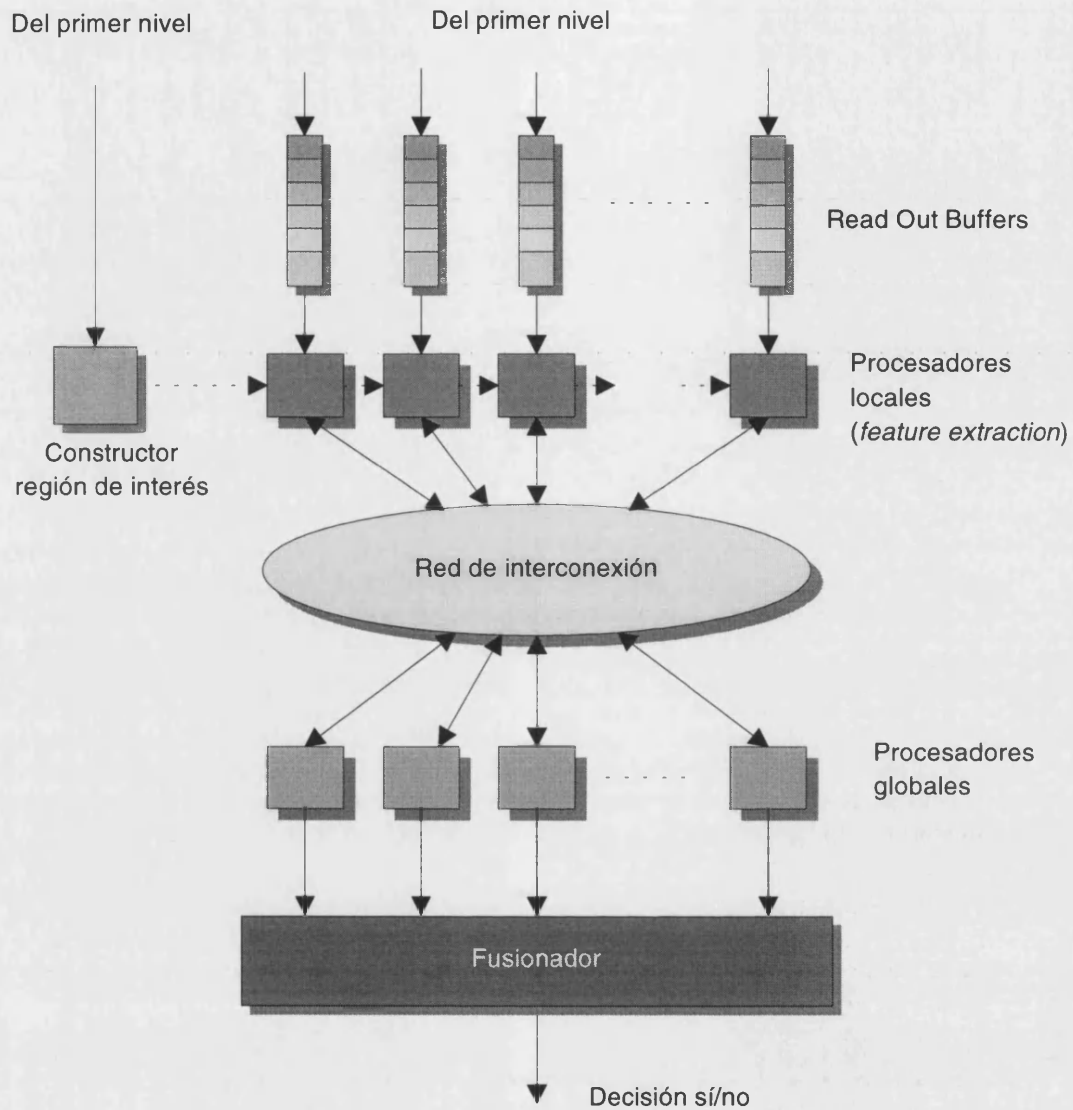


FIGURA 3.8 Esquema del segundo nivel tras la introducción de regiones de interés.

Se realizaron simulaciones [10] para estimar el número de regiones de interés por cada cruce de haz en función de los resultados simulados de la colisión. La figura 3.9 muestra el histograma resultado de esas simulaciones. Como se observa el número de regiones de interés que más se da es cuatro y cinco, aunque a veces puede llegar a ser nueve. Evidentemente, un número elevado de regiones de interés implica un mayor procesado, por lo que se decidió tomar un 5 RoIs por suceso como valor representativo.

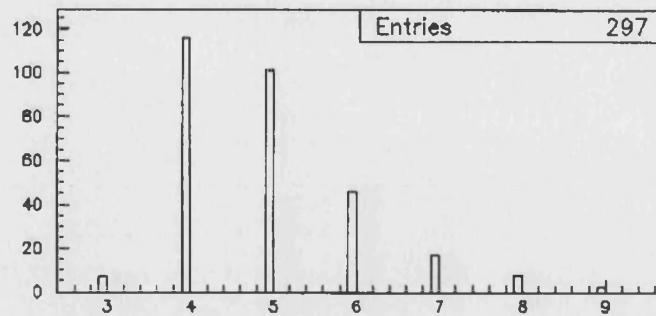


FIGURA 3.9 Distribución del número de RoI por suceso.

Se estudió después el número de canales que compondrían cada región de interés. Para el caso de los calorímetros, el tamaño de la región de interés varía en función de la partícula que se detecta (electrones, fotones o jets de partículas). El tamaño máximo sería el de un área de 12 x 12 celdas, mientras que el mínimo sería de 4 x 4.

El procesado necesario para la localización de las regiones, conocido como *cluster finding*, consiste en la definición de un área de 4 x 4 celdas en el calorímetro electromagnético en la que se realiza la suma de la energía de cada par de celdas en el área interior de 2 x 2 celdas. Esta suma se compara con un umbral de cluster. Además se exige que la suma de la energía en las 12 celdas alrededor del área de 2 x 2 sea menor que un umbral de aislamiento electromagnético (un valor mínimo de energía) y que la energía de las 16 celdas del calorímetro hadrónico por detrás del área de 4 x 4 sea menor que un umbral de aislamiento hadrónico. Una vez acabado el procesado en este área, se desplaza en ambas coordenadas una celda y se repite el procesado. La figura 3.10 muestra el algoritmo propuesto.

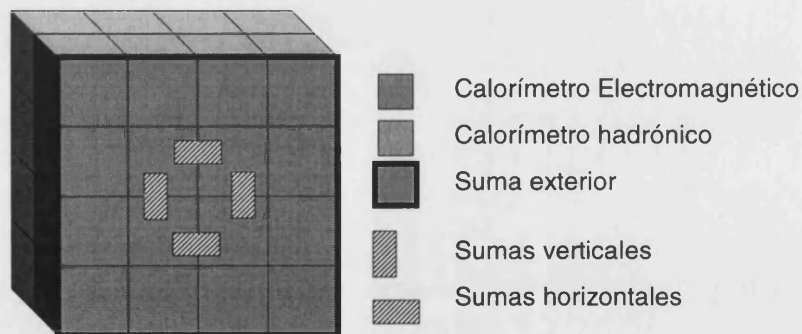


FIGURA 3.10 Algoritmo para la determinación de la RoI.

En el caso del detector de muones, la región de interés está formada por áreas de 6 x 6 celdas mínimo y 60 x 60 máximo en función de la zona del detector donde se defina la región. El algoritmo empleado para la definición de la RoI en el caso de los muones consiste en identificar los puntos del detector con señal por encima del umbral y enviar la información correspondiente.

Entre los dos calorímetros y el subdetector de muones, el número de canales electrónicos por región de interés está alrededor de 135.000 [13]. Como el número

medio de regiones por suceso son 5, podemos determinar ya el factor de reducción δ_1 (según la ecuación 2.34) que relaciona el número de canales a procesar cuando existen regiones de interés y el total para el segundo nivel de trigger:

$$\delta_1 = \frac{135.000 \cdot 5}{150 \cdot 10^6} = 4'5 \cdot 10^{-3} \quad (3.2)$$

Es decir, al emplear regiones de interés estamos procesando alrededor del 0'5 % del número total de canales lo que supone un gran ahorro en la capacidad de procesado necesaria.

Para evaluar la capacidad de procesado del constructor de la región de interés se ha considerado que este elemento se implementará mediante dispositivos ASIC para disponer de una gran velocidad y debido a que las operaciones a realizar no serán complejas. Se estima que la capacidad de procesado necesario en el constructor estará alrededor de $12 \cdot 10^3$ MIPS [11] (se necesitan alrededor de 256 dispositivos ASIC de 40 MIPS para los calorímetros más una cantidad sin determinar todavía para el detector de muones), por lo que el valor del factor γ_1 de reducción en la capacidad de procesado respecto a la total del problema es:

$$\gamma_1 = \frac{12 \cdot 10^3}{5 \cdot 10^{10}} = 2'4 \cdot 10^{-7} \quad (3.3)$$

El número de unidades necesarias para el constructor lo obtenemos de la ecuación 2.35, resultando en un valor de 500. El valor de K_{t2} lo obtenemos de los datos que se encuentran en [11], resultando $K_{t2} = 0'0875$.

Teniendo en cuenta el valor de δ_1 calculado anteriormente podemos calcular el nuevo número de unidades del nivel mediante la ecuación 2.38, resultando ser de 1440 unidades de procesado.

Acudiendo a la ecuación 2.40 y sustituyendo todos los valores obtenemos el valor de nuevo factor de calidad requerida. A efectos de comprobar la mejora y puesto que tanto el primer nivel como el tercero no han cambiado, (no ha variado su aportación en el factor de calidad requerida) nos limitaremos a comparar los factores de calidad requerida del segundo nivel únicamente.

Según la tabla 3.4 el factor de calidad requerida del segundo nivel para el caso sin regiones de interés vale 10^{18} . Si evaluamos el mismo concepto para el caso con regiones de interés obtenemos un valor de $6'3 \cdot 10^{15}$. Como se observa, la reducción es notable, un poco más de dos órdenes de magnitud, debido a la disminución del número de canales a procesar.

La tabla 3.5 muestra la comparación entre las dos opciones (con regiones y sin regiones). Hay que observar que, debido al hecho de que se introducen las regiones de interés, el número de unidades de procesado en el nivel desciende, con lo que el factor de calidad requerida disminuye. Al no ser compensada esta disminución por la introducción del constructor de regiones de interés, el efecto total es la disminución del factor de calidad requerida.

	AB_{entrada} (Mbytes/s)	CP (MIPS)	N_i	Q (MB/s x MIPS)
Sin regiones	10^4-10^5	$32 \cdot 10^6$	$32 \cdot 10^4$	10^{18}
Con regiones	10^4-10^5	$144 \cdot 10^3$	1440	$6'3 \cdot 10^{15}$

TABLA 3.5 Comparación de los parámetros del segundo nivel entre la arquitectura con regiones y sin regiones.

3.2 AGRUPAMIENTO DE DATOS (ROD)

El segundo nivel de trigger trabajará con datos correspondientes a segmentos físicos bien definidos del detector (regiones de interés). Aunque se podría tener un sistema capaz de acceder a porciones de los datos distribuidas de forma arbitraria a lo largo de un gran número de procesadores, es extremadamente importante minimizar el número de localizaciones en las cuales se sitúan los datos requeridos. Esto facilitaría la recolección de los datos y haría el sistema más barato y fiable.

Por lo tanto, será necesario para el segundo nivel de trigger que los datos se agrupen de forma que se adapten lo más posible a la segmentación en regiones de interés o subconjuntos de éstas. Además el número total de ROBs (Read Out Buffers, definidos en el epígrafe anterior) tiene que minimizarse. Esto requiere que los buffers almacenen tantos canales como sea posible, por lo que es necesario que antes exista algún elemento que realice este agrupamiento. Este dispositivo de agrupamiento de datos es el que recibe el nombre de Read Out Driver o ROD.

De las dos opciones apuntadas en el capítulo 2 para el agrupamiento de los datos, hemos escogido la estática por dos motivos:

- Es más sencilla de implementar puesto que el agrupamiento puede realizarse físicamente agrupando los cables o fibras hacia cada procesador.
- Los fenómenos físicos que se estudian tienen una distribución geométrica predecible y por tanto el agrupamiento puede ser casi óptimo [12].

A la hora de evaluar la bondad de la introducción de este elemento (para nosotros, cómo modifica el factor de calidad requerida) lo importante es cuantificar la probabilidad de que los datos requeridos por el procesador local no se encuentren en su ROB y tenga que ir a otro a buscarlos.

Para ello se ha realizado un estudio del sistema de procesado de segundo nivel buscando identificar el número de ROB por RoIs y la probabilidad de que la RoI estuviese en el buffer ROB adecuado [13]. En este estudio se ha supuesto ya que la implementación del sistema consistirá en una serie de clusters de procesadores interconectados entre sí, por lo que se ha evaluado también la probabilidad de que el dato esté en algún ROB dentro del cluster cuando no lo esté en el correspondiente. La tabla 3.6 muestra los resultados de este estudio.

Del mismo, se desprende que la probabilidad media de encontrar los datos en el ROB es de un 6 % mientras que la de encontrar los datos en el cluster es de un 43 %, aproximadamente.

Detector	ROBs/RoI	Frecuencia en %	
	(media)	ROB	Cluster
SCT	4'6	3'5	43
TRT	2	3'1	40
CALORÍMETRO (electrones y fotones)	4	2'7	35
CALORÍMETRO (jets)	19	16'1	94
MUONES	4	4	6

TABLA 3.6 Número de ROBs por RoI y probabilidad de encontrar los datos de la RoI en el ROB y en el cluster para los diferentes subdetectores.

En el estudio realizado en el capítulo 2 se suponía que todos los procesadores se conectaban entre sí a través de una única red de interconexión, de ahí la introducción de una sola probabilidad de efectuar un acceso externo. En este caso, existen dos probabilidades diferentes, la primera, referida al ROB, puede asemejarse a la que se introdujo en el análisis del capítulo 2 y la segunda, referida al cluster, que determina la proporción de accesos a clusters diferentes del ocupado por el ROB.

La distinción de las probabilidades y, por tanto, de los tiempos de accesos a los datos en función de la localización de los mismos, implica la modificación de las expresiones calculadas anteriormente para introducir este hecho. Dicha modificación es sencilla puesto que sólo hay que introducir un término dependiente de la probabilidad de acceso a otros clusters y evaluar las operaciones a realizar en cada caso. De cualquier modo, mantenemos la suposición de que la capacidad del cálculo de las unidades de procesado no varía y que lo que se hace es aumentar su número para acomodarse a la nueva situación.

Según la ecuación 2.58, y añadiendo ahora la posibilidad de accesos remotos con probabilidad p_r , tenemos que:

$$\begin{aligned}
 CP_{ij} &= \delta_{i-1} \cdot \alpha_i \cdot \frac{D/N_i^{(cp)} \cdot op + p_{ai} \cdot D/N_i^{(cp)} \cdot op_a + p_{ri} \cdot D/N_i^{(cp)} \cdot op_r}{t} = \\
 &= \delta_{i-1} \cdot \alpha_i \cdot \frac{D \cdot op}{N_i^{(cp)} \cdot t} \quad \begin{array}{l} 1 \leq j \leq N'_i \\ 1 \leq i \leq E \end{array} \quad (3.4)
 \end{aligned}$$

donde op_r es el número de operaciones necesarias para realizar el acceso remoto. Despejando $N_i^{(cp)}$ obtenemos una expresión semejante a la del capítulo 2:

$$N_i^{(cp)} = N_i \cdot \left[1 + p_{ai} \overline{op_a} + p_{ri} \overline{op_r} \right] \quad 1 \leq i \leq E \quad (3.5)$$

Procediendo de manera análoga para el número de unidades en función del ancho de banda, tenemos:

$$N_i^{(ab)} = N_i^{(ab)} \cdot (1 + 2 \cdot p_{ai} + 2 \cdot p_{ri}) \quad (3.6)$$

Y, también, el número de unidades al incluir transferencias remotas es:

$$N_i'' = \max(N_i'^{(cp)}, N_i'^{(ab)}) \quad (3.7)$$

El factor de calidad requerida del nivel i puede obtenerse con la misma ecuación 2.39 teniendo en cuenta la variación en el número de unidades.

De igual manera, el factor de calidad requerida del sistema completo sigue la expresión 2.40, con la misma salvedad expresada anteriormente.

Veamos, pues, cual es el nuevo valor del factor de calidad requerida para el segundo nivel si aplicamos agrupamiento de datos y teniendo en cuenta los valores medios de las probabilidades de acceso a otro ROB o a otro cluster.

Para ello necesitamos conocer el nuevo número de unidades que depende del número de operaciones por dato a realizar suponiendo que no se existen accesos externos y el que hay que realizar para efectuar el acceso.

A partir de la capacidad de procesado del segundo nivel obtenida anteriormente podemos calcular el número de operaciones totales realizadas durante los 10 μ s de procesado del segundo nivel de trigger. Como la capacidad de procesado es $1'44 \cdot 10^6$ MIPS, el número de operaciones será $1'44 \cdot 10^7$.

Para obtener el número de operaciones por dato sólo hay que saber el número de datos. Este valor se estima a partir del volumen de información total que se procesa en este segundo nivel, es decir, la cantidad de información correspondiente a 5 regiones de interés en todos los subdetectores. Este valor es de unos 27 Kbytes de información[13]. Como cada dato se toma de 2 bytes, el número de datos es de 13824. Dividiendo el número de operaciones entre este valor obtenemos 1042 operaciones por dato.

El número de operaciones por dato a ejecutar para realizar el acceso remoto tanto dentro como fuera del cluster lo podemos suponer igual a uno. Los valores de las probabilidades de acceso externo son: $p_{e2} = 1-6/100$ y $p_{r2} = 1-43/100$.

Sustituyendo en la ecuación (3.5), al emplear agrupamiento de canales, el número de unidades ha de incrementarse y pasa a valer:

$$N_2'^{(cp)} = 1440 \cdot [1 + (1 - 6/100) \cdot 1/1042 + (1 - 43/100) \cdot 1/1042] \cong 1442 \text{ procesadores} \quad (3.8)$$

En función de la variación del ancho de banda, ecuación (3.6), el número de unidades necesarias es:

$$N_2'^{(ab)} = 500 \cdot [1 + 2 \cdot (1 - 6/100) + 2 \cdot (1 - 43/100)] = 2010 \text{ procesadores} \quad (3.9)$$

Y por lo tanto, el número de unidades será el mayor de los dos valores obtenidos, es decir, $N_2'' = 2010$. Este valor es del mismo orden de magnitud que el previsto por la colaboración a partir de estudios ajenos a éste [6], lo que corrobora la exactitud de los cálculos realizados por este procedimiento.

Sustituyendo valores en la ecuación 2.40 para $i = 2$, el nuevo valor del factor de calidad requerida del nivel se incrementa a $8'8 \cdot 10^{15}$ MIPS x MB/s, es decir, aproximadamente 1'4 veces más que en el caso ideal en el que no hay agrupamiento

de datos porque se supone que los procesadores tienen todos los datos necesarios en sus memorias.

Este incremento es normal puesto que el cálculo realizado sin agrupamiento de datos supone que los datos necesarios para el procesamiento los tiene cada procesador en su memoria y por lo tanto, no existe comunicación entre los procesadores. Al introducir el agrupamiento lo único que hacemos es aproximarnos más al caso real en el que, de alguna manera, existirá una comunicación de datos entre los procesadores. Esto, evidentemente, influye en el ancho de banda requerido para los procesadores y por lo tanto en el factor de calidad requerida, como ha quedado patente.

La tabla 3.7 resume todo el proceso de mejora del segundo nivel.

	AB_{entrada} (Mbytes/s)	CP (MIPS)	N_i	Q (MB/s x MIPS)
Sin regiones	$10^4 \cdot 10^5$	$32 \cdot 10^6$	$32 \cdot 10^4$	10^{18}
Con regiones	$10^4 \cdot 10^5$	$144 \cdot 10^3$	1.440	$6'3 \cdot 10^{15}$
Con regiones	$4'02 \cdot 10^4 \cdot 4'02 \cdot 10^5$	$144'4 \cdot 10^3$	2.010	$8'8 \cdot 10^{15}$
+				
agrupamiento (RODs)				

TABLA 3.7 Resumen de los principales resultados de aplicar regiones de interés y agrupamiento de datos en el segundo nivel.

3.3 OPCIONES TECNOLÓGICAS PARA LA IMPLEMENTACIÓN DE LOS NIVELES

Vamos a ver a continuación, una vez determinada la arquitectura a emplear en cada nivel, sobretodo en lo referente al segundo, algunas opciones para la implementación de cada uno de los niveles.

Puesto que la tecnología avanza muy rápidamente, la opción final sobre qué dispositivos serán empleados en el experimento no está decidida, y se espera realizarla, a la vista de los diferentes resultados obtenidos de las pruebas con las diferentes soluciones, a finales de 1999.

3.3.1 El primer nivel

El primer nivel será eminentemente hardware, es decir, estará implementado mediante circuitos digitales de propósito específico (FPGAs, lógica discreta e integrada o ASICs) donde correrán los algoritmos de procesamiento.

El procesamiento será necesariamente encauzado (pipelined), debido a la elevada frecuencia de entrada de datos, y muchas de las operaciones se realizarán en paralelo. Este tipo de procesador tendrá un conjunto fijo de algoritmos pero será programable en lo referente a sus parámetros. La transmisión de los datos a los procesadores y el procesamiento encauzado serán sincrónicos con el reloj de 40 MHz del LHC.

Muchas de las opciones barajadas incluyen la realización de los circuitos a medida, ASICs, y su producción para el experimento. Esta opción tiene la ventaja del ajuste perfecto a la aplicación y la posibilidad de emplear tecnologías rápidas, como el AsGa, muy poco extendidas en dispositivos lógicos discretos.

El principal problema de cualquier solución para este nivel se encuentra en su ubicación dentro del experimento. Para evitar pérdidas de señal y retrasos, gran parte del primer nivel se situará en o alrededor del detector. Esto hará que los dispositivos tengan que trabajar bajo la influencia de dosis de radiación, por lo que su resistencia a la misma es un parámetro importante.

Los transistores bipolares son inherentemente resistentes a la radiación. Los efectos producidos por la radiación se traducen en un incremento de la generación de corriente en la base, que reduce la ganancia de corriente. La magnitud de la generación de corriente es proporcional al volumen de la base, por lo que, transistores con anchuras de base menores, es decir mayores velocidades, son más resistentes a la radiación. En el seno de la colaboración se han desarrollado preamplificadores bipolares integrados para el detector interno y se han probado con éxito después de ser irradiados [6].

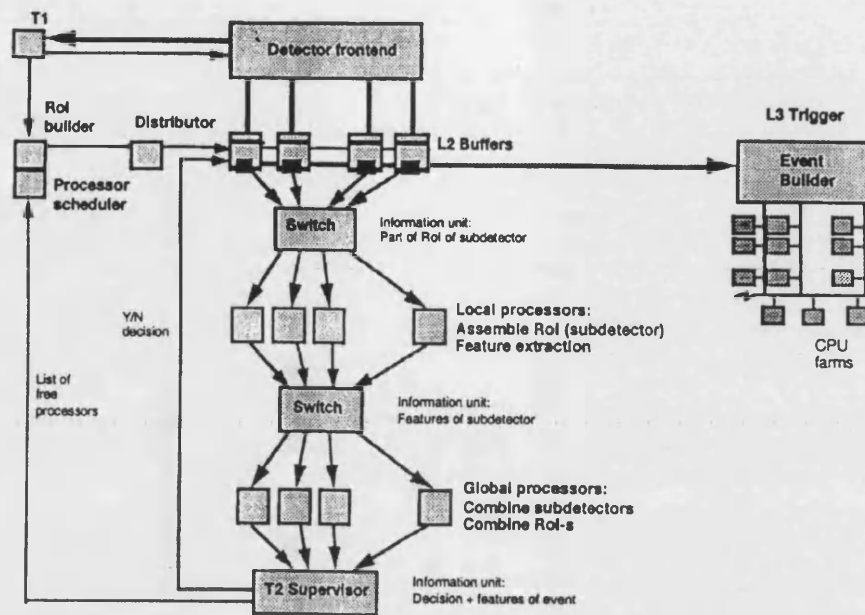
En cuanto a la tecnología CMOS, existen dos tipos de procesos resistentes a radiación: *bulk* y *silicon-on-insulator (SOI)*. Los circuitos desarrollados con estas tecnologías (memorias, preamplificadores) han probado mantener su funcionalidad más allá de 100 KGy¹. Las dosis esperadas en el experimento ATLAS oscilan entre 0.02 y 2300 KGy/año [5], lo que quiere decir que los dispositivos podrían funcionar entre 5.000 años y 2 semanas en función de su ubicación. Evidentemente, en la instalación de la electrónica se tendrá en cuenta este factor.

3.3.2 El segundo nivel

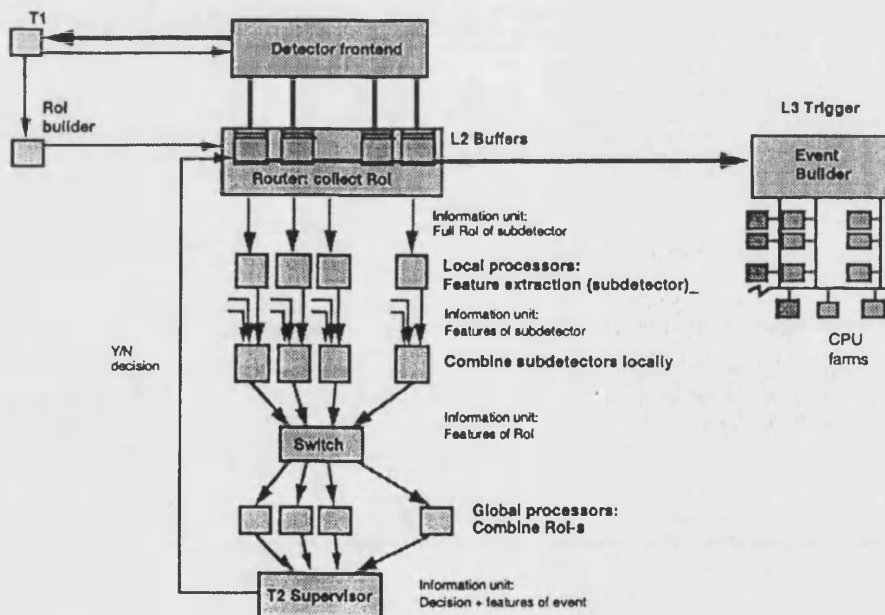
De los tres niveles, el segundo es el que presenta un interés mayor desde el punto de vista tecnológico. En efecto, mientras que el primer nivel realiza operaciones de selección basadas en umbrales (supresión de ceros), implementadas mediante dispositivos discretos o integrados *ad hoc* y el tercer nivel hace uso de la potencia de cálculo de las estaciones de trabajo gracias a que la decisión puede tardar un poco más, el segundo debe elaborar una decisión casi definitiva en un tiempo no demasiado elevado y sobre un volumen de información relativamente grande. Por ello, este nivel hace uso de sistemas multiprocesadores para el procesado en tiempo real de los datos, y redes de interconexión de alta velocidad.

Para la implementación de la arquitectura de este nivel, se han propuesto diferentes soluciones. Puesto que es un proyecto a largo plazo, todavía continúan los trabajos de investigación y desarrollo, aunque básicamente se barajan dos opciones, conocidas como *data-driven architecture* y *farm-based architecture*, mostradas en la figura 3.11.

¹ KGy = kilo Gray, medida de la dosis de radiación absorbida por un cuerpo. 100 Gy = 1 rad = 10⁻² J/kg.



"farm-based"



"data-driven"

FIGURA 3.11 Opciones data-driven y farm-based para el segundo nivel de trigger.

En la arquitectura *farm-based*, los datos almacenados en los buffers de segundo nivel reciben, a través del supervisor de nivel 2, una cabecera que les indica a cuál de entre los procesadores locales disponibles se deben dirigir. Puesto que los datos pueden venir de distintos detectores, los procesadores tienen que poder ejecutar distintos algoritmos en función de los datos. Por ello, tanto a nivel de procesador local como global, se piensa en procesadores flexibles con gran potencia de cálculo, como procesadores digitales de señal (DSP).

La matriz de conmutación de nivel 2 es la encargada de rutar la información de cada subdetector a uno o varios procesadores para realizar el procesamiento local. Una vez realizado éste, toda la información extraída ha de ser enviada a procesadores globales para realizar la decisión “sí/no”.

Los requerimientos de ancho de banda para la red local/global se calculan en 100 MBytes/s puesto que sólo hay que enviar información resumida del análisis local a los procesadores globales.

En la arquitectura *data-driven*, los buffers son los encargados de recoger la información de cada región de interés en cada detector. Después, toda esa información se envía a los procesadores locales que se encuentran asignados a cada subdetector. De esta manera, el procesador realiza siempre el mismo algoritmo, y puede optimizarse para ello. En esta opción se piensa en procesadores sistólicos implementados con dispositivos programables FPGA, de los que ya existe algún prototipo construido y evaluado [14]. Esta opción mantiene, sin embargo, la solución de procesadores generales para la evaluación global.

En lo referente a las posibles opciones para los procesadores locales, para la arquitectura farm-based, ya se han realizado pruebas con el procesador digital de señal TMS320C40 [15][16][17] y con otros dispositivos, como procesadores de vídeo [18] (MaxVideo [19]), procesadores matriciales de flujo de datos (DAVIS[20]), sistemas SIMD masivamente paralelos (MasPar [21]), procesadores VLSI (Blitzen [22]), procesadores de señal empotrados (iWarp [23]) e incluso nuevos desarrollos como el ASP (Associate String Processor [24]). La tabla 3.8 muestra los resultados en cuanto a tiempo de ejecución de los algoritmos de extracción de características para el calorímetro y el detector de trazas [25].

Las conclusiones de este estudio son:

- las arquitecturas encauzadas con un flujo de datos casi síncrono con el sistema parecen ser el camino natural para conseguir algoritmos que funcionen con frecuencias de 100 KHz.
- el paralelismo masivo implementado con elementos de procesamiento simples no corresponde idealmente con los problemas evaluados, y es difícil de poner en práctica (MasPar). Cuando se reduce el número de procesadores, el concepto SIMD parece ser manejable, aunque actualmente sólo se han implementado los algoritmos más simples. Parece, entonces, que una arquitectura de trigger basada en sistemas SIMD podría aprovechar sólo el chip multiprocesador básico, realizando encauzamiento entre múltiples pequeños sistemas SIMD. Sin embargo, actualmente no existe nada parecido en el mercado.
- la flexibilidad en los algoritmos encuentra una limitación en la parte referida a la comunicación. Varios de los sistemas evaluados (DAVIS, Blitzen, ASP) requieren una práctica especial para aprovechar los detalles específicos de la arquitectura en cuestión. Los algoritmos típicos de extracción de características tendrían que ser, eventualmente, codificados en hard o soft, dejando sólo los parámetros críticos bajo el control del usuario.
- todos los sistemas paralelos comerciales favorecen la comunicación a corta distancia y presentan cuellos de botella cuando se trata de realizar sumas globales de datos. Sólo los procesadores diseñados a propósito (por ejemplo los convolucionadores en los procesadores de imagen) parecen capaces de tratar este problema eficientemente.

Arquitectura/algoritmo	Tiempo de ejecución (μ s)
Pipeline	
	MaxVideo (TRT) 20'0
	MaxVideo (Calo) 51'0
	DAVIS (TRT) 3'0
	DAVIS (Calo) 14'3
SIMD	
	MasPar (TRT) 741
	MasPar (Calo) 782
	Blitzen (TRT) 147
	Blitzen (Calo) 76'6
	ASP (TRT) 33'5
	ASP (Calo) 20'6
Otra	
	iWarp (TRT) 16
	iWarp (Calo) 33

TABLA 3.8 Tiempos de ejecución de los algoritmos de extracción de características para distintos tipos de procesadores.

Para la red de interconexión las soluciones que se investigan se centran en nuevas redes de interconexión de propósito general como ATM [26][27], SCI [28][29][30] o Fiber Channel [31], descritas en el capítulo 2, y switches integrados como el C104 [32].

Estas dos opciones, *farm-based* y *data-driven*, siguen siendo evaluadas para distintas implementaciones. La tecnología actual no permite diferenciar claramente cuál de las dos opciones es más adecuada. Por tanto, el trabajo que actualmente se lleva a cabo consiste extrapolar, a partir de los resultados que se obtienen de las diferentes implementaciones y de la tecnología actual de los procesadores y dispositivos digitales, para intentar obtener una visión de futuro que permita, si es posible, dirimir la cuestión [33].

3.3.3 El tercer nivel

Diversos fabricantes de computadoras proporcionan ya sistemas multiprocesador muy potentes con interfaces a estándares de comunicación como SCI, FibreChannel o ATM. En este sentido se espera que las compañías fabricantes proporcionen en el futuro soluciones basadas en estándares industriales.

Existen actualmente dos modelos alternativos para proporcionar potencia de procesado masiva:

- el modelo *farm*, donde los procesadores individuales (o clusters de procesadores) se complementan mediante una red de interconexión para proporcionar un sistema de computación modular y coherente [34].
- la máquina de procesado masivamente paralelo o MPP que combina procesadores e interconexiones en un sistema único y escalable.

Para el nivel 3, donde los procesadores actuarán individualmente en sucesos enteros, la elección de una u otra arquitectura es indiferente. Las *farms* pueden ser apropiadamente escaladas al nivel requerido y los sistemas MPP soportan particionado lógico en unidades menores.

El trabajo desarrollado hasta ahora ha consistido en el modelado del nivel 3 basándose en FibreChannel o HiPPI [35] y en tecnologías de conmutación.

Se ha evaluado la posibilidad de implementar el sistema mediante un conmutador de un único nivel de 256 x 256 con el resultado de una capacidad máxima de 1200 sucesos/s y una utilización del ancho de banda del 24%. Esta solución adolece de problemas de coste, falta de disponibilidad de grandes conmutadores y problemas en la integración del sistema. También se ha evaluado la opción de un sistema con dos etapas con conmutadores 16 x 16. La eficiencia en la utilización del ancho de banda es mayor que la opción de una sola etapa, alcanzando el 42 % [36].

En este nivel todavía se necesita realizar más trabajo para poder identificar la solución óptima adecuada al problema y al coste previsto para la implementación.

4. 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] Kozlov, V. *Functional simulation of detector, front-end and read-out parts of a LHC-like DAQ architecture*, CERN/RD13-120, 1994.
- [11] Brawn, I. P. et al. *The level-1 calorimeter trigger system for ATLAS*, ATLAS-DAQ-NO-30, 1995.
- [12] Bock, R. et al., *More readout specifications (LVL2 demonstrator program, Addendum a la nota interna DAQ-NO-53*, 1996.
- [13] Bock, R. y LeDu, P. *Detector and readout specifications for the level-2 trigger demonstrator program*, ATLAS/DAQ-NO-53. 1996.
- [14] Noffz, K. H. et al., *Results of On-Line tests of the Enable Prototype*, EAST Note 94-03, 1994.
- [15] Clarke, P. E. L. et al., *Test of components for an asynchronous level-2 trigger for ATLAS*, ATLAS/DAQ-NO-36, 1994.

- [16] Strong, J. *Local processing for a farm-based second level trigger at LHC*, ATLAS/DAQ-NO-21, 1994.
- [17] Kastrup, B. et al., *Study of the FEX subfarm concept with C40s and Digital Memory Channel cluster*, ATLAS/DAQ-NO-68, 1997.
- [18] Bock, R. K. et al., *A commercial image processing system considered for triggering in future LHC experiments*, CERN/EAST nota 94-26, 1994.
- [19] DataCube Inc., *MaxVideo image processing systems*, documentación.
- [20] Schmidt, U. *Data-Driven array processor for video signal processing*, IEEE Transactions on Consumer Electronics, n° 8, 1990.
- [21] MasPar documents: *System overview (PN 9300-0300)*, *MPL Reference Manual (PN 9302-0000)*, *MPL User's guide (PN 9302-0100)*.
- [22] Blevins, D. W. et al., *BLITZEN: A highly integrated massively parallel machine*, Journal of Parallel and Distributed Computing 8/2, pág. 150, 1990.
- [23] Borkar, S. et al., *iWarp: An integrated solution to high-speed parallel computing*, Supercomputing '88, IEEE Computer Society and ACM SIGARCH, Orlando, Florida, pág. 330, 1988.
- [24] Lea, R. M. *The ASP: A cost effective parallel microcomputer*, IEEE Micro, Volumen 8, n° 5, 1988.
- [25] Vermeulen, J. et al. *Evaluating parallel architectures for two real-time applications with 100 KHz repetition rate*, CERN/EAST nota 92-18, 1992.
- [26] Calvet, D. *Using ATM switches model in SIMDAQ*, ATLAS/DAQ-NO-sin asignar, 1994.
- [27] Calvet, D. et al., *Performance analysis of ATM network interfaces for data acquisition applications*, ATLAS/DAQ-NO-64, 1997.
- [28] Clarke, P. E. L. et al., *SCI with DSPs and RISC processors for LHC 2nd Level triggering*, ATLAS/DAQ-NO-19, 1994.
- [29] ATLAS modelling group, *Modelling results of the ATLAS data acquisition and trigger system*, ATLAS/DAQ-NO-sin asignar, 1994.
- [30] RD24 Collaboration, *RD24 Status Report. Application of the SCI to Data Acquisition at LHC*, CERN/DRDC-95,1995.
- [31] Ambrosini, G. et al., *Experience with FibreChannel in the environment of the ATLAS DAQ prototype "-1" project*, ATLAS/DAQ-NO-88, 1998.
- [32] SGS Thomsom Microelectronics, *C104 Asynchronous Packet Switch Data Sheet*.
- [33] Bock, R. K. et al., *Status Report: Embedded Architectures for Second-Level triggering (EAST)*, CERN/DRDC 94-20. RD11 Status Report (EAST Note 94-16), 1995.
- [34] Mornacchi, G. *Preliminary performance study of LVL3-like MP architectures*, ATLAS/DAQ-NO-23, 1994.
- [35] *High Performance Parallel Interface. Mechanical, Electrical and Signalling Specification (HIPPI- PH) X3.183*, 1991.
- [36] Greiman, W. *Design and simulation of FibreChannel based event builders*, CERN/RD13 nota 132, 1994.

CAPÍTULO

4

VERIFICACIÓN EXPERIMENTAL

1. INTRODUCCIÓN	1
2. DISEÑO DE UN MÓDULO DE AGRUPAMIENTO DE DATOS	1
2.1 El detector TileCal	1
2.2 Especificaciones generales del módulo	4
2.3 Especificaciones particulares para TileCal	5
2.4 Diagrama de bloques del prototipo	5
2.4.1 La tarjeta DBV44	7
2.4.2 La tarjeta interface SCI	8
2.4.3 El módulo interface S-LINK	9
2.4.3.1 Solución adoptada. Diseños alternativos	12
2.4.3.2 Cálculos	15
2.4.3.3 DISEÑO DE LA PLACA	17
3. DESCRIPCIÓN DE LAS PRUEBAS Y RESULTADOS	20
3.1 Determinación del número de DSPs	20
3.2 Integración del prototipo	25
3.2.1 Sistema StandAlone	25
3.2.2 Sistema TestBeam	29
3.2.2.1 Descripción del TestBeam para TileCal	30
3.2.2.2 Inclusión del prototipo	34
3.2.3 Resultados	37
4. BIBLIOGRAFÍA	38

1. INTRODUCCIÓN

En este capítulo se describe el trabajo realizado en la implementación de un módulo de agrupamiento y procesado de datos para la verificación experimental de algunas de las conclusiones del capítulo 2. La imposibilidad de trabajar con el sistema de adquisición final, debido a que su puesta en marcha se prevé para el 2004, limita la comprobación a algunos de los aspectos descritos en el segundo capítulo.

Para ello, se procederá al diseño de este módulo adaptándose a las características concretas de uno de los subdetectores del experimento ATLAS/LHC, el calorímetro TileCal.

Comenzaremos con una breve exposición de las características de este subdetector para después enunciar los requerimientos del módulo y posteriormente describir la implementación realizada.

Tras esta exposición, se describirán las pruebas realizadas tanto en los laboratorios del Departamento de Informática y Electrónica de la Universitat de València, como en los del CERN, el Laboratorio Europeo de Física de Altas Energías, en Ginebra. Por último se detallarán los resultados obtenidos de las mismas.

2. DISEÑO DE UN MÓDULO DE AGRUPAMIENTO DE DATOS

A continuación se describirá el diseño de un módulo de agrupamiento ROD para el calorímetro TileCal propuesto en 1995 a la colaboración ante la necesidad de mejorar el sistema de adquisición de datos. Las especificaciones iniciales del módulo se encuentran en [4] y fue en base a éstas con las que se diseñó el sistema. A lo largo de este tiempo, debido a ajustes en el diseño del detector, en los sistemas de trigger y a un mejor conocimiento del problema se ha producido un reajuste aunque las ideas básicas no ha variado.

En primer lugar, se describirá el subdetector en el que se realizarán las pruebas para conocer sus características, puesto que influyen directamente en el diseño del módulo.

Tras esto, se establecerán las especificaciones del módulo para después comentar las distintas opciones tecnológicas barajadas en el diseño. Por último se presentará el diseño del prototipo realizado y las diferentes pruebas realizadas en diferentes bancos de test.

2.1 EL DETECTOR TILECAL [1]

La figura 4.1 muestra un detalle del sistema de calorimetría del detector ATLAS. La estructura está dividida en una parte central y dos laterales. El criostato central alberga el calorímetro electromagnético y el solenoide superconductor de 2 Teslas que ya se comentaron en la descripción general del detector (apartado 2.1 del capítulo 3).

El calorímetro hadrónico TileCal se sitúa fuera de la estructura criogénica. Está dividido en una parte central (barril) y dos laterales (barriles extendidos). La estructura de soporte del calorímetro se emplea también como retorno de flujo del campo magnético del solenoide.

Su forma es cilíndrica con un radio interno de 228 cm y uno externo de 423 cm. La estructura está dividida longitudinalmente en tres secciones, una central (barril) de 564 cm de largo y dos externas (barriles extendidos) de 264 cm de largo cada uno. Entre la parte central y las laterales existe una separación de 0.7 m para la extracción de los conductores del detector interno.

ATLAS Calorimetry

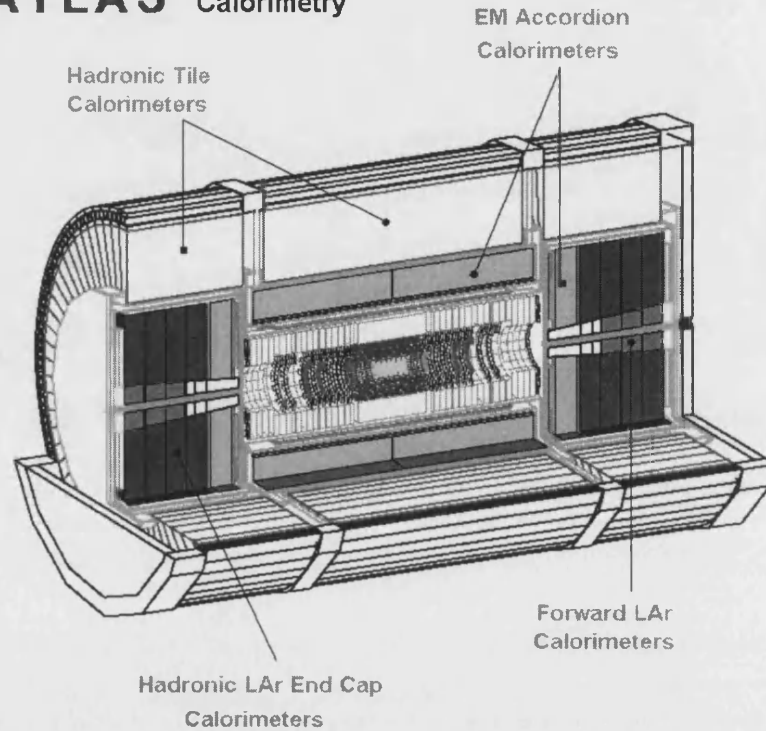


FIGURA 4.1 El sistema de calorimetría de ATLAS. Obsérvese la posición del calorímetro hadrónico TileCal.

Cada parte se divide azimutalmente en 64 sectores. En la parte del barril cada sector está dividido longitudinalmente en dos segmentos o *superdrawers*, mientras que las partes laterales están compuestas de un único *superdrawer*; la estructura de cada *superdrawer* puede verse en la figura 4.2. Todos ellos están compuestos de un material pasivo (hierro), cuya misión es la de "frenar" las partículas, y un material activo (plástico centelleador) que produce luz al paso de las partículas cuya intensidad es proporcional a la energía de la misma. Los fotones producidos son leídos por fibras ópticas y conducidos a unos fotomultiplicadores donde se convierte en señal eléctrica apta para su procesado.

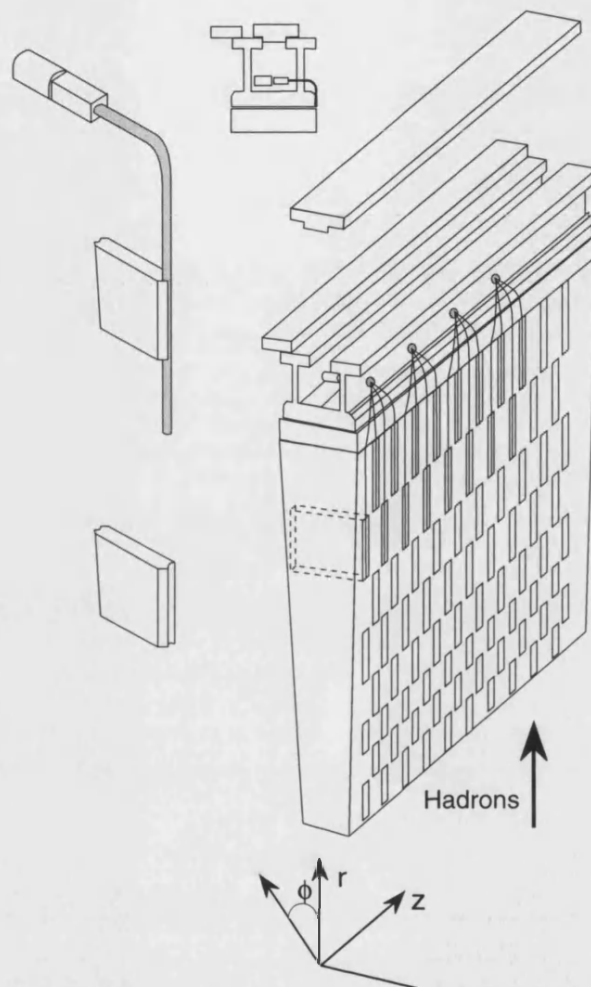


FIGURA 4.2 Estructura interna del calorímetro TileCal. El eje Z da la dirección aproximada de los haces que colisionan.

Para leer todo el detector, el número de fotomultiplicadores necesario es de unos 10.000. La elección del fotomultiplicador concreto no está todavía fijada pero uno de los mejores candidatos es el modelo R5900 de la firma Hamamatsu [2] de dimensiones extremadamente compactas ($28 \times 28 \times 20 \text{ mm}^3$).

Junto a los fotomultiplicadores, situados dentro del propio detector, se sitúa la electrónica de acondicionamiento de la señal o *front end* que consiste en un divisor de tensión para la alimentación del fotomultiplicador, un conformador de señal y un sistema de inyección de carga para el calibrado del detector. Todo este conjunto se encuentra dentro de un cilindro de hierro y μ -metal para proporcionar el adecuado aislamiento frente al campo magnético.

Toda la electrónica anterior se encuentra situada en una gaveta que se inserta en la parte superior de cada uno de los sectores. Además, en esta gaveta se incluye también el sistema de conversión analógico/digital basado en un dispositivo

digitalizador pipeline conocido como *Front End and Readout Microsystem* o FERMI [3]. La salida del digitalizador se lleva hasta los sistemas de *trigger* de primer y segundo nivel. Cada canal ofrecerá una información de 2 bytes, por lo que la tasa de datos a leer en el primer nivel será de 745 Gbytes/s, mientras que a la salida del mismo (entrada del segundo nivel, donde se sitúa el ROD) será de unos 2 Gbytes/s.

2.2 ESPECIFICACIONES GENERALES DEL MÓDULO

Según [5], las especificaciones cualquier módulo ROD para ATLAS, y por lo tanto para TileCal, son las siguientes:

- **Agrupamiento de datos**

El ROD recibe los datos de los buffers desaleatorizadores a una frecuencia de 100 KHz y los multiplexa de forma que se ajusten a las dimensiones de las regiones de interés o subconjuntos de las mismas.

- **Sincronismo**

Deberá detectar y corregir errores sobre los datos recibidos cuando sea posible. Para ello, recibirá las señales de sincronismo (TTC) [6] del acelerador LHC. Dicha señal contiene información acerca del número de suceso y el número de cruce de haz en el que se ha producido el suceso que se está leyendo.

- **Detección y recuperación de errores**

Realiza la detección y recuperación de errores de consistencia de los datos. Para el control de errores, el ROD comprobará que el número de suceso de los datos que le llegan concuerda con el que él ha recibido a través del TTC y es consistente con el número de cruce de haz. Si se detecta un error, el sistema no debe detenerse, sino que el ROD ha de intentar recuperar ese error viendo si las tramas se resincronizan o enviar hacia el segundo nivel los sucesos recibidos marcándolos con un flag de error.

- **Procesado y formateo de datos**

Los datos pueden procesarse en el ROD en función de si se considera necesario en cada uno de los subdetectores. En cualquier caso, a la salida del ROD deberán ajustarse a un formato determinado por la colaboración ATLAS antes de ser enviados a los ROB del segundo nivel. Dicho formato deberá incluir por lo menos la información referente al número de evento (24 bits), número de cruce de haz (12 bits), una palabra de error/estado, un identificador único para cada ROD y el conjunto de los datos. Incluso si no hay datos, el resto de la información debe ser enviada.

- **Enlace óptico**

La salida de datos del ROD será óptica y a una velocidad de 1 Gbit/s. Debido a que la velocidad de este enlace es limitada puede ser necesario un buffer antes del enlace. Puesto que su capacidad será finita, debe protegerse contra el desbordamiento. Si el ROD detecta que este buffer está casi lleno, enviará una señal ROD_BUSY hacia el primer nivel de trigger para que deje de enviar datos.

Los datos deben salir del ROD hacia el ROB a una frecuencia de 100 KHz tanto si se trata de un mensaje con datos como si no los tiene, de forma que se mantenga el sincronismo en el segundo nivel. Se implementará un mecanismo XON/XOFF para informar al ROD de que los buffers de segundo nivel se encuentra casi llenos y de que debe activar la señal ROD_BUSY.

- **Test**

Deberá incluir un modo de test para comprobar el funcionamiento de la cadena de lectura desde el ROD a los buffers de segundo nivel. Para ello, habilitará un mecanismo para enviar sucesos falsos con contenido conocido (un patrón fijo o uno cargado en el sistema a tal efecto).

- **Calibración y monitoraje**

Se considera también posible el acceso a los datos por un canal distinto al normal. El ROD contará con un puerto auxiliar utilizable para monitoraje, calibración, etc. que no debe introducir tiempos muerto ni interferir con el modo de operación normal.

2.3 ESPECIFICACIONES PARTICULARES PARA TILECAL

Como ya se ha comentado, las especificaciones para TileCal han sufrido ligeras modificaciones a lo largo de estos tres años en los que el proyecto está en marcha. Dichas modificaciones se refieren más a cuestiones concretas sobre el número exacto de canales a procesar por ROD o a la tecnología de los enlaces a emplear. Cuestiones como el procesamiento de datos a realizar no han sido determinadas por los estudios físicos hasta hace bien poco.

Todos los requerimientos generales enunciados en el epígrafe anterior son válidos para TileCal, pero para centrar más aún el problema se detallan a continuación aquéllos específicos para este subdetector:

- Cada módulo ROD procesará la información de 4 ó 5 *superdrawers*. Lo que supone un total de entre 52 y 64 RODs para poder leer todo el detector.
- El número de canales del detector por ROD se estima en unos 200. Cada canal contendrá la información correspondiente a los fotomultiplicadores de los *superdrawers* e información acerca del cruce de haz. Esto supone un total de 782 bytes por fibra, por lo que la tasa de datos a leer será de unos 80 Mbytes/s.
- El enlace óptico a emplear para enviar los datos hacia el ROD desde los desaleatorizadores y de éste hacia los ROBs, seguirá el protocolo S-LINK [7] definido en el CERN. La tecnología a emplear será OptoBus de Motorola [8] para el enlace entrante y FibreChannel para el de salida.
- Para el TileCal se piensa realizar un preprocesado sobre los datos. Este procesado tiene como objetivo la obtención de la suma de las energías depositadas en cada uno de los canales procesados. El resultado de este procesado será una información de 4 bytes, suficientes para albergar la información de energía, a la que habrá que añadir el resto del formato definido para ATLAS [9].

2.4 DIAGRAMA DE BLOQUES DEL PROTOTIPO

En el diseño del prototipo se tuvieron en cuenta las especificaciones generales sobre los ROD y, siendo conscientes de que los requerimientos sufrirían variaciones a lo largo del tiempo, se pretendió realizar un sistema que cumpliera con la funcionalidad descrita para el ROD antes que con los detalles más específicos. Se buscó un sistema mediante el cual se pudieran realizar pruebas durante el periodo de desarrollo del detector y que permitiera su integración paulatina en el sistema de adquisición de datos del ATLAS, todavía por implementar. Por ello, el diseño ha sufrido sucesivos refinamientos y reajustes hasta quedar configurado como se muestra en la figura 4.3, donde se representa el diagrama de bloques del prototipo construido.

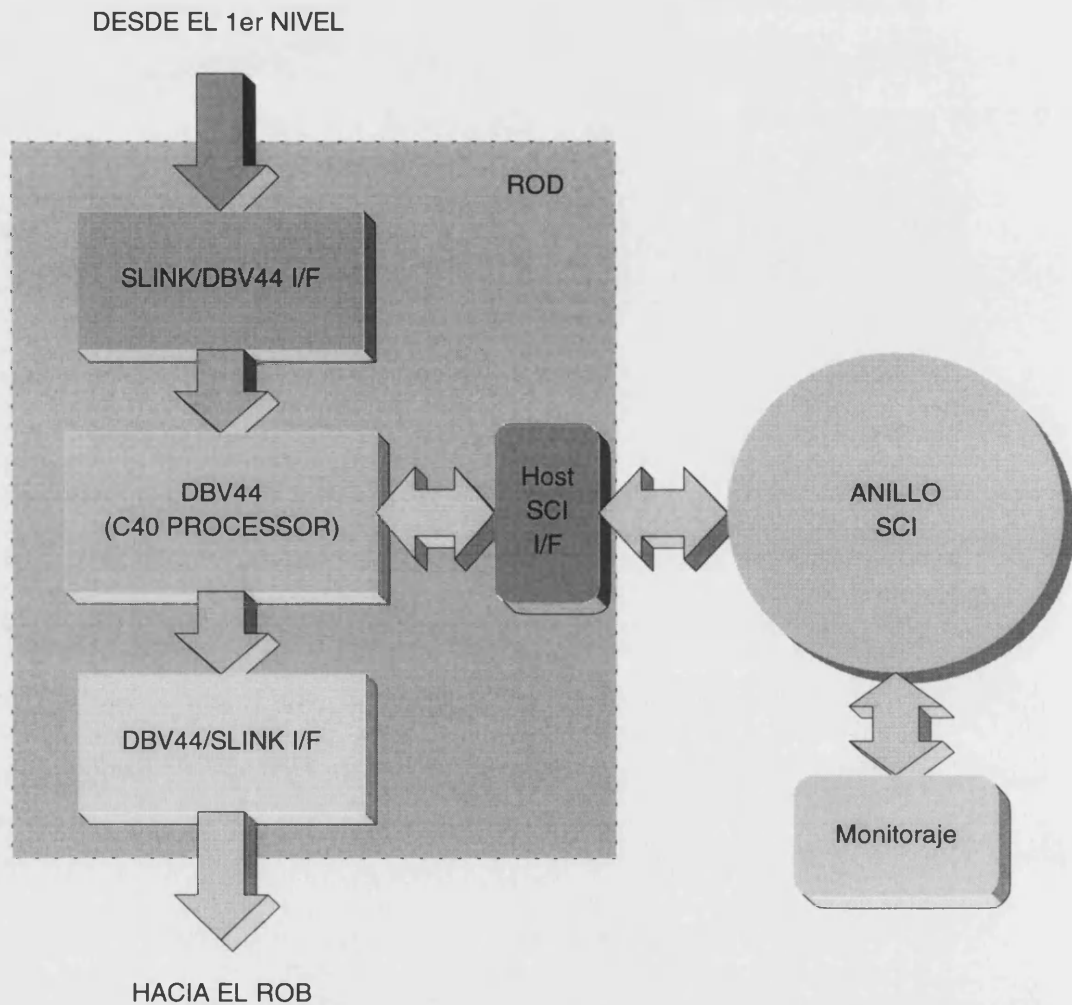


FIGURA 4.3 Diagrama de bloques del prototipo ROD para TileCal.

El sistema está constituido por tres tarjetas VME de la siguiente manera:

- Una tarjeta procesadora encargada de realizar las operaciones necesarias sobre los datos. Esta tarjeta, denominada DBV44 y fabricada por la firma británica LSI Ltd., es comercial y es un sistema modular capaz de albergar hasta ocho procesadores digitales de señal TMS320C40 de Texas Instruments [10]. Para este primer prototipo se calculó un número máximo de canales de 30 para que su procesamiento no excediese de 10 μ s.
- Una tarjeta de interface a SCI. Como se desprende de los requerimientos generales, existe la posibilidad de un acceso auxiliar a los datos procesados en el ROD. Para el detector TileCal, como ya se ha comentado, el volumen total de datos está en torno a los 2 Gbytes/s. El protocolo SCI es, por tanto adecuado para el propósito de este puerto auxiliar. Este interface está basado en una tarjeta comercial PowerPC con un adaptador PCI/SCI.
- Una tarjeta destinada a la entrada y salida de datos a través de S-LINK. Se encarga de realizar el resto de las operaciones para cumplir con los requerimientos

fijos. Entre ellas están la generación de ROD_BUSY o la señal XON/XOFF para el control de flujo entre el ROD y el ROB.

En una primera fase se implementó sólo el enlace de salida hacia S-LINK, al no disponerse todavía de acceso a los datos del detector a través de este protocolo, sino a través de VME.

Vamos a ver a continuación con más detalle cada una de las tarjetas mencionadas anteriormente, comentando, cuando así sea necesario, detalles de su implementación concreta o de las soluciones tecnológicas empleadas.

2.4.1 La tarjeta DBV44

La figura 4.4 muestra una fotografía de la tarjeta procesador DBV44. Dispone de cuatro zócalos de extensión donde es posible conectar hasta 4 módulos TIM [11] con un máximo de ocho procesadores C40.

En la parte superior de la foto se pueden apreciar los dos conectores IDC del bus de expansión dBeX32, de 32 bits de datos y 22 de direcciones, empleado para la conexión a la tarjeta interface S-LINK. Uno de los conectores es de 60 pines de densidad estándar mientras que el otro es de 40 pines de alta densidad. El hecho de que existan dos conectores se debe a que la tarjeta DBV44 también soporta la conexión a este bus de periféricos dBeX (de 32 bits de datos pero sólo 7 de direcciones) y que sólo precisan la conexión al conector de 60 pines.

Este bus está conectado mediante buffers al bus global de uno de los módulos TIM de la placa desde donde es accesible y funciona a una frecuencia de 20 ó 25 MHz en función de la versión de C40 que se emplee (40 ó 50 MHz).

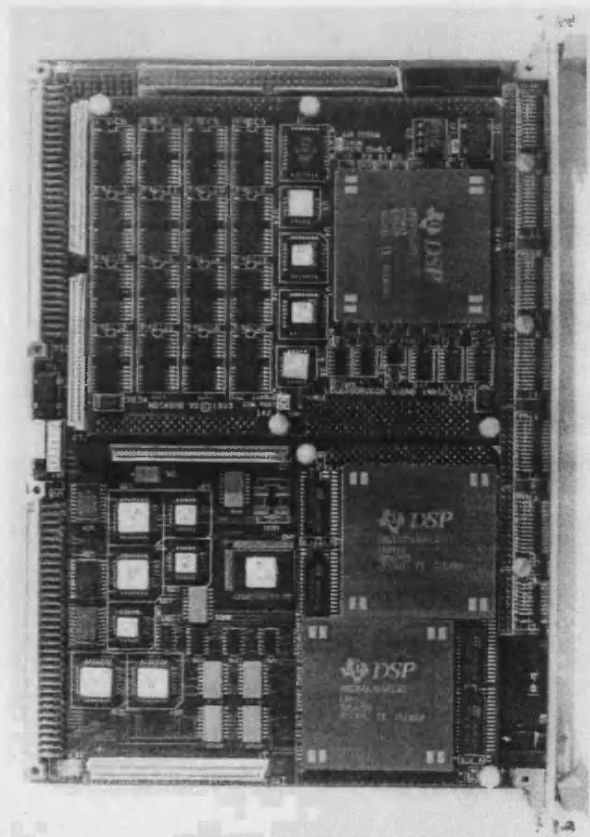


FIGURA 4.4 La tarjeta comercial DBV44 de LSI Corporation.

2.4.2 La tarjeta interface SCI

La tarjeta interface con SCI está basada en un módulo con formato PMC (PCI Mezzanine Card) [12] que realiza la conexión PCI/SCI. Este módulo ha sido desarrollado dentro del proyecto RD24 del CERN [13][14]. La figura 4.5 muestra una fotografía del mismo.

La forma de operar es la siguiente: en cualquier momento, a través del bus VME al que está conectado todo el sistema, se pueden leer los datos que se están procesando en la tarjeta DBV44. Estos datos se almacenan en la memoria de la tarjeta PowerPC y son enviados a través de un anillo SCI hacia un sistema de monitoraje donde se leen.

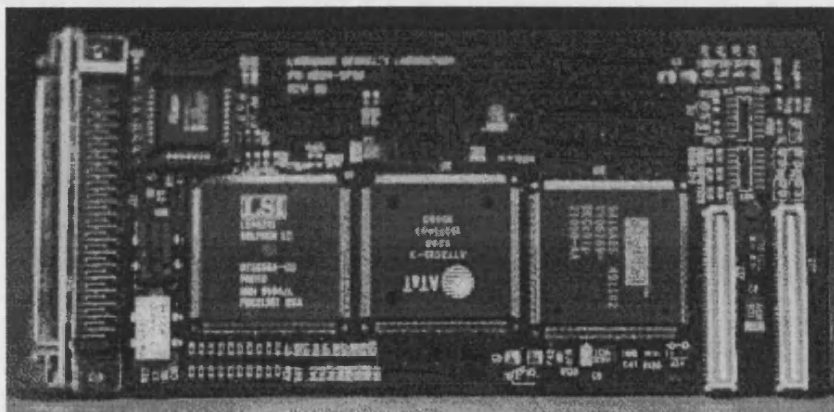


FIGURA 4.5 Módulo PMC PCI/SCI para la conexión a un anillo SCI (versión desarrollada en el seno de la colaboración RD24 del CERN).

La figura 4.6 muestra el esquema del sistema y el flujo de datos entre los diferentes módulos que lo componen.

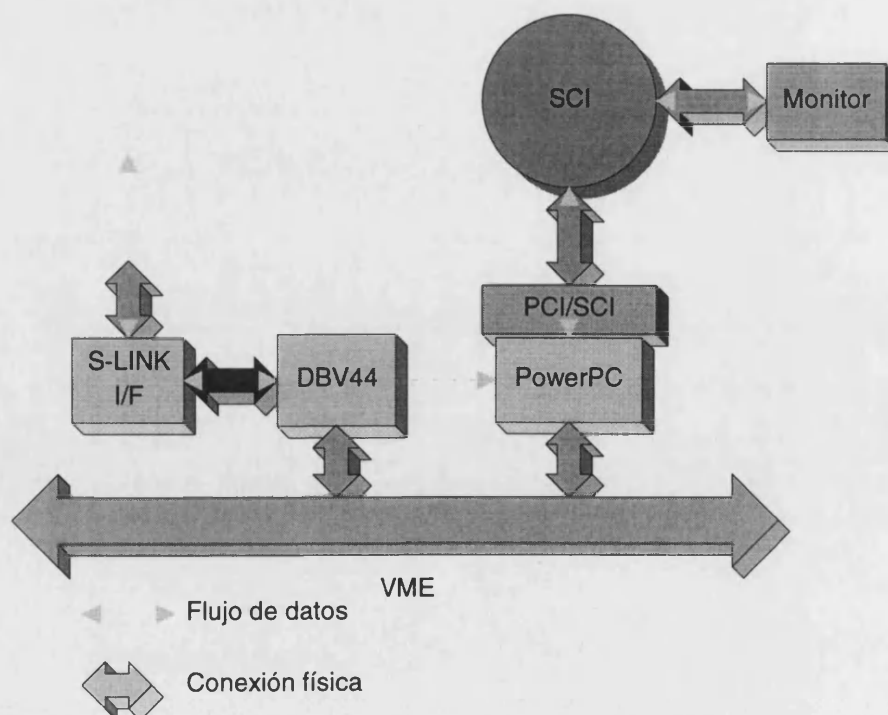


FIGURA 4.6 Diagrama de conexión y flujo de datos del prototipo ROD construido.

2.4.3 El módulo interface S-LINK

Este módulo se encarga de la entrada y salida de datos desde el primer nivel y hacia los buffers del segundo. No es un módulo comercial al tener que adaptarse a los requerimientos de nuestro sistema.

La figura 4.7 muestra el diagrama de bloques de la tarjeta implementada. Como se observa existen tres buses a los que la tarjeta se conecta: el dBeX32, el bus S-LINK y el bus del puerto del C40.

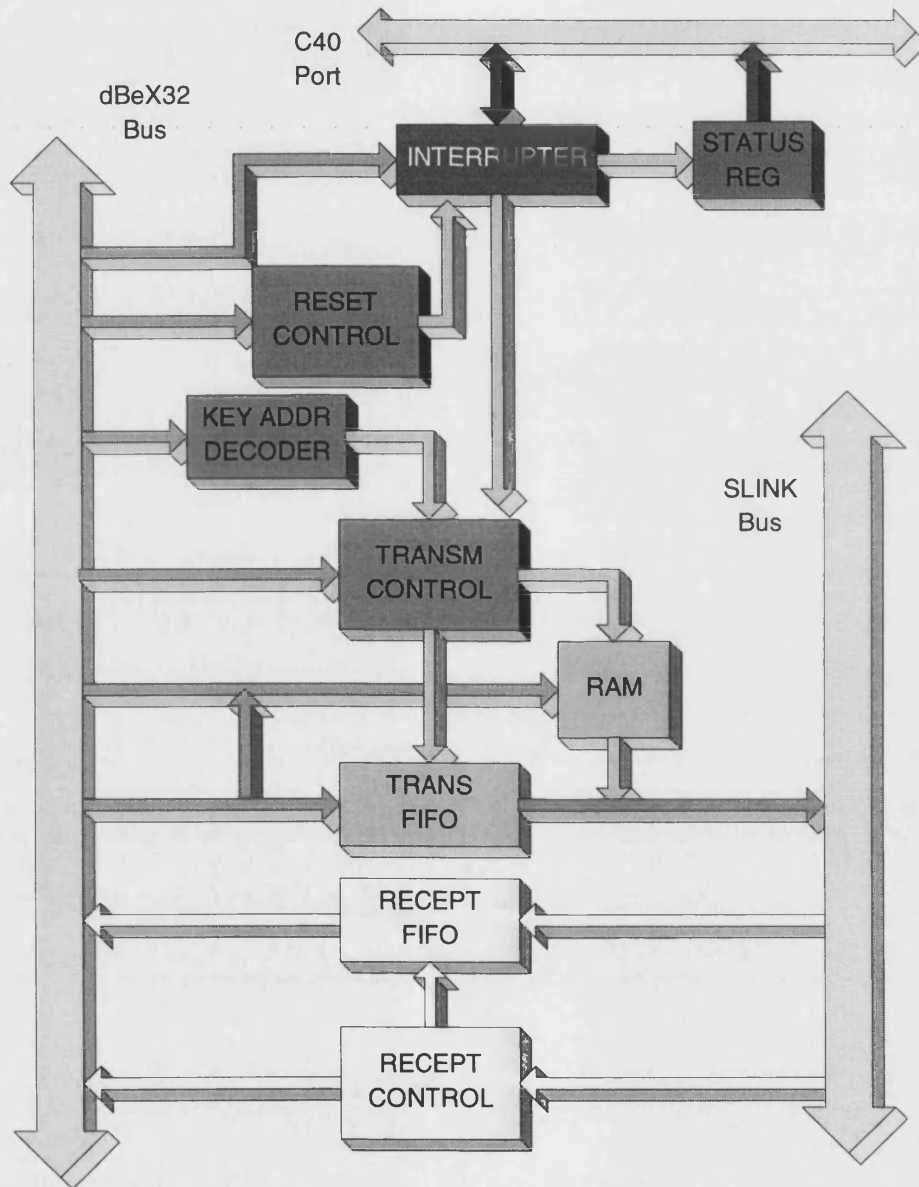


FIGURA 4.7 Diagrama de bloques del interface S-LINK. La parte de recepción no se ha implementado en la primera versión del prototipo.

El bus dBeX32 se emplea para la recepción y envío de datos de y hacia la tarjeta DBV44 y para la recepción de comandos de control desde la misma. A través de este bus es posible realizar la programación de las diferentes opciones de funcionamiento del sistema.

El bus del puerto del C40 se emplea para la generación de interrupciones por parte de la tarjeta interface para reclamar la atención de la tarjeta DBV44 y exigir la correspondiente acción.

Como se describe en [7], el protocolo S-LINK no define el medio físico a través del cual se envían los datos, sino sólo las señales y la temporización que asegura el buen funcionamiento del mismo. El protocolo define cuatro unidades funcionales: el LSC (Link Source Card), el LDC (Link Destination Card), el FEMB (Front-End Motherboard) y el ROMB (Read-Out Motherboard). La figura 4.8 muestra la relación que existe entre estas unidades.

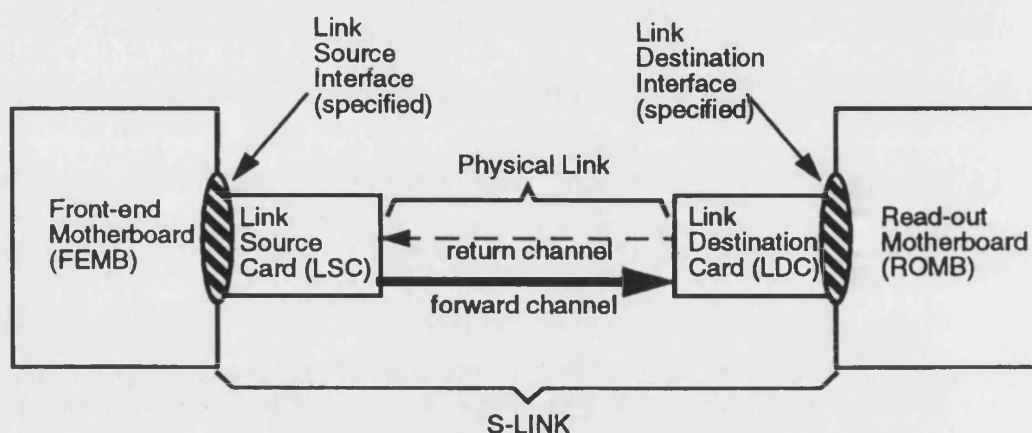


FIGURA 4.8 Módulos funcionales del protocolo S-LINK.

Desde el punto de vista de S-LINK, nuestro sistema es una mezcla de FEMB y ROMB, dejándose libre la elección de la LSC y LDC más adecuada para la aplicación (como ya se definió anteriormente, TileCal empleará OptoBus y FibreChannel como enlaces físicos del ROD).

Alrededor de estos tres buses se dispone la lógica adecuada para la realización de las operaciones necesarias del ROD. Esta lógica incluye:

- Memorias FIFO de transmisión. Encargadas de almacenar los datos recibidos desde la DBV44, que se han de enviar a través de S-LINK, a la espera de la señal de 100 KHz.
- Memoria RAM. Almacena el mensaje a enviar cuando no existen datos. Como se recordará el sistema debe mantener en todo momento el sincronismo. El contenido del mensaje es programable desde la DBV44.
- Bloque de control de transmisión. Encargado de controlar la transmisión de datos a través de S-LINK.
- Control de reset. Encargado de generar la secuencia de reset de la placa cuando así se le indica mediante un comando desde la DBV44 o mediante un pulsador frontal.

- Bloque interrupter. Activa las señales correspondientes en el bus del puerto del C40 para provocar una interrupción en el mismo. Las causas que pueden provocar esta interrupción son:
 - Enlace S-LINK listo para la transmisión.
 - Pérdida de sincronismo en el enlace S-LINK.
 - Transmisión demasiado rápida desde la DBV44. FIFO de salida casi llena.
 - Activación de la señal externa XONF. Parada de la transmisión.
- Registro de estado. Registro de 16 bits que contiene en todo momento la información acerca del estado del sistema. Incluye detalles sobre el estado del S-LINK, de las FIFO, del estado de la máquina de estados que controla los enlaces de salida, etc. Este registro de estado es accesible tanto a través de la interrupción como a través del bus dBeX32.
- Bloque *Key Address Decoding*. Para controlar la tarjeta se escogió esta técnica que consiste en enviar los comandos a través del bus de direcciones, en vez de a través del bus de datos. La ventaja es que se gana velocidad y el sistema es más sencillo en lo referente a la decodificación. La desventaja es que el mapa de memoria se amplía, pero para nuestra aplicación esto no supone un inconveniente pues solamente se tiene un dispositivo conectado al bus.

El funcionamiento de la tarjeta es como sigue. Cada 10 μ s, el sistema comprueba si en la FIFO de salida hay uno o más bloques para transmitir (un bloque consiste en 128 bytes = 30 canales x 2 bytes x 2 bytes (resultado del procesado) + 8 bytes de cabecera). Si los hay, se envía uno a través del S-LINK y el sistema espera al siguiente periodo de 10 μ s; si no lo hay, se envía la información de bloque vacío contenida en la RAM.

En todo momento se comprueba el estado del enlace S-LINK en previsión de pérdidas de sincronismo, el llenado de la FIFO y el estado de la línea externa XONF, que implementa el protocolo XON/XOFF requerido. Ante cualquier anomalía, se provoca una interrupción del procesador C40 para que éste actúe en consecuencia. Además, la tarjeta dispone de una línea de salida RBUSY (ROD_BUSY) activable desde la DBV44 en caso de que se recibiesen datos con demasiada velocidad o se produjese la activación de la línea XONF.

El conjunto de comandos disponible para el control de la tarjeta se puede ver en la tabla 4.1 A cada comando le corresponde una dirección diferente del bus dBeX32. De los 24 bits de direcciones sólo se emplean de A0 a A4 para los comandos, reservándose de A23 a A13 para la selección de la tarjeta a través de su dirección base programable por software. Los bits restantes (A12 a A5) se emplean para direccionar la escritura del mensaje vacío en la RAM (hay que recordar que un mensaje consta de 128 bytes, es decir, 32 posiciones de memoria de 32 bits, aunque se deja la posibilidad de ampliar hasta 255 el tamaño del mismo).

Comando	Dirección	Comentario
addwfifo	00000	escritura de bloques en la FIFO
addrbon	00001	activa la señal ROD_BUSY
addrbof	00010	desactiva la señal ROD_BUSY
addofon	00011	reset de la FIFO
addofof	00100	respuesta al ciclo de reset de la FIFO
addolon	00101	reset del S-LINK

Comando	Dirección	Comentario
addwmem	00110	activa la RAM para la escritura del mensaje vacío
addint	00111	reconocimiento de interrupción por parte del C40
addrxonf	01000	reconocimiento de un cambio de estado el XON/XOFF
addesr	01001	lectura del registro de estado
addolof	01010	respuesta al ciclo de reset del S-LINK

TABLA 4.1 Comandos disponibles para la tarjeta interface S-LINK.

2.4.3.1 Solución adoptada. Diseños alternativos

Veamos a continuación las diferentes alternativas tecnológicas evaluadas para este diseño.

Lógica de control, reset, registro de estado, descodificación e interrupter.

La implementación del diseño se ha basado en dos premisas fundamentales:

Velocidad: se debe procurar una implementación con dispositivos “rápidos” ya que la velocidad es el parámetro fundamental en el diseño.

Sencillez: debido a la limitación de espacio en la placa, se debe procurar una total integración del sistema, reduciendo al máximo el número de componentes y aumentando en lo posible la flexibilidad.

Teniendo en cuenta el punto sobre velocidad se puede escoger una familia lógica estándar de alta velocidad como la ECL, pero esto conlleva una serie de problemas como son:

- Necesidad de una etapa de conversión TTL/ECL: tanto la etapa anterior como posterior trabaja en niveles TTL.
- Fuente de alimentación para ECL.
- Un número elevado de integrados: diseño y etapa de conversión.
- Nula flexibilidad en el diseño.

Para evitar la etapa de conversión y la fuente de alimentación, se puede pensar en utilizar una familia lógica como la AS, con parámetros de tiempo muy similares a la familia ECL. Aún así el número de componentes estándar sigue siendo elevado (a *prorateo* se estima en unos 70 los integrados estándar necesarios para este diseño, sin contar las memorias FIFO y RAM y el resto de componentes discretos).

Así pues, se ha optado por emplear dispositivos programables para reducir al máximo el número de componentes. Con esta premisa, se pueden utilizar los dispositivos programables de varios fabricantes: *Xilinx*, *AMD*, *Altera*, *Lattice*. De todos estos fabricantes se han utilizado los dispositivos de LATTICE varias razones:

- Alta velocidad: son de entre los más rápidos del mercado.
- No necesitan sistema de programación: Son del tipo ISP (*In-System Programming*); es decir se pueden programar con una tensión de 5V en el mismo sistema donde están trabajando, lo que supone un ahorro en el programador, evitar parar por un tiempo prolongado el sistema, y facilidad en la programación.

- Facilidad de compra: a diferencia de otros dispositivos, los de *Lattice* se pueden encontrar con relativa facilidad.
- Sistema de desarrollo compatible con varios fabricantes: para este proyecto se ha utilizado el Synario empleando el lenguaje Abel y un *fitter* para *Lattice*.

El dispositivo empleado en el diseño ha sido el ispLSI2064-80LJ:

Gates: 2000

Speed-Mhz: 81

Speed-Tpd: 15

Macro-Cells: 64

Registers: 64

Inputs&I/O: 68

Puesto que el prototipo debía ser instalado y probado en los laboratorios del CERN, el motivo de la elección de este dispositivo en concreto ha sido puramente logístico: ha sido el más fácil de localizar entre los más rápidos. Esto no quiere decir que sea el óptimo, existen otros dispositivos de *Lattice* que presentan mejores condiciones para este proyecto, pero que por su elevado coste y largo plazo de entrega se han desechado para este prototipo. Como ejemplo de otros dispositivos adecuados para este diseño tenemos los siguientes:

Familia ispLSI3000:

ispLSI3192-100L

Gates: 9000

Speed-Mhz: 100

Speed-Tpd: 10

Macro-Cells: 192

Registers: 384

Inputs&I/O: 192

ispLSI3256E-100L

Gates: 12000

Speed-Mhz: 100

Speed-Tpd: 10

Macro-Cells: 256

Registers: 384

Inputs&I/O: 128

Familia ipsLSI6000: son de las más actuales. Tienen la característica de poseer varios módulos específicos como FIFO y SRAM (de puerto simple o doble). Debido a su reciente aparición son difíciles de conseguir y de un elevado precio.

Memorias FIFO

Para la memoria FIFO de salida, el principal requerimiento era la velocidad de trabajo, ya que su contenido debía ser extraído en menos de 10 μ s para quedar lista para el siguiente bloque.

El dispositivo elegido es el SN74ACT7803/12 de 512x18 bits. Las principales características son:

- Tamaño de palabra de 18 bit, con una capacidad para 512 palabras.
- Reloj de escritura y lectura independiente.
- Tecnología Low-Power Advanced CMOS.
- Flag de “medio llena” y bandera programable de “casi llena/vacía”.
- Configuración a bidireccional y expansión a 36bits.
- Tiempo de acceso de 12 ns.
- Compatible TTL.
- Encapsulado DL de 56 pines.

Al ser el dBeX32 un bus de 32 bits, han sido necesarias dos unidades de forma que de la primera se toman 18 bits y de la segunda 14, dejando sin emplear las 4 restantes.

También se ha empleado un dispositivo ispGDS14, una matriz programable que puede sustituir al conjunto de interruptores-resistencias convencionales a la hora de implementar una lógica de determinación de la dirección base de la placa, con la diferencia de ser programable por software. Las características técnicas de este dispositivo son:

- Tiempo de retardo de 7'5 ns.
- Frecuencia máxima de 50 MHz.
- Consumo típico de 40mA.
- Tamaño de la matriz: 7x7.
- 14 entradas/salidas.
- Encapsulado 20-PLCC.

Memorias RAM

La memoria RAM requiere las mismas prestaciones que la FIFO. Por ello, y por la necesidad de circuitería adicional, se descartaron las memorias dinámicas.

El modelo escogido es la MCM6206D-12 de 32Kx8, cuya capacidad es suficiente para almacenar el bloque vacío. Sus características son:

- Tamaño de palabra de 8 bit, con una capacidad para 32K palabras.
- Totalmente estática.
- Tiempo de acceso de 12ns.
- Control de triestado para eliminar los problemas de contención del bus.
- Compatible TTL.

- Bajo consumo.
- Encapsulado DIP de 28 pines.

Como necesitamos 32 bits, se emplearon 4 unidades de esta memoria.

Osciladores

En el diseño hacía falta generar dos señales de frecuencia bastante dispar. Por una parte se necesitaban 33 MHz para el reloj del protocolo S-LINK, y por otra había que generar una señal de 100 KHz.

Para obtener estas dos señales se emplearon dos osciladores de cristal de cuarzo integrados:

- De 33MHz con una estabilidad de frecuencia de ± 100 ppm, con una tensión de alimentación de 5V y un *fan-out* de 10. La salida es estándar TTL. El encapsulado tipo 14DIP.
- De 1MHz con una estabilidad de frecuencia de ± 100 ppm, con una tensión de alimentación de 5V y un *fan-out* de 10. La salida es estándar TTL. El encapsulado tipo 14DIP.

La señal de 100KHz se ha generado a partir de la de 1MHz empleado un divisor por 10 (74LS56).

Etapas de conversión TTL-NIM/NIM-TTL

La señal de control de entrada XON/XOFF y la señal de control de salida RBUSY debe ser señales compatibles con los niveles NIM. Un nivel lógico 0 en NIM corresponde a una tensión de 0V, mientras que un nivel 1 corresponde a -0.85 V. Por lo tanto, fue necesario realizar la adaptación de los niveles TTL a NIM y viceversa.

2.4.3.2 Cálculos

Potencia consumida

Los datos empleados en cada uno de los dispositivos son los proporcionados por los fabricantes.

1. isPLSI 2064-80LJ:

La corriente consumida por estos dispositivos viene definida por la siguiente ecuación:

$$I_{CC}(mA) = 38 + (n^{\circ}PTs \cdot 0'33) + (n^{\circ}nets \cdot MaxFreq \cdot 0'007) \quad (4.1)$$

donde:

- $n^{\circ}PTs$ = número de términos producto empleados en el diseño.
- $n^{\circ}nets$ = número de señales empleadas en el diseño.
- $MaxFreq$ = máxima frecuencia de trabajo (en MHz).

La tabla 4.2 muestra cada uno de los anteriores parámetros para cada una de la FPGA empleadas.

ISP N°	N°PTs	N°NETS	MaxFreq	ICC(mA)
U19	51	65	50	78
U20	100	62	50	93
U23	32	64	50	71
U22	217	76	50	137

TABLA 4.2 Parámetros de las FPGAs para el cálculo de la potencia.

Luego el consumo total es de 379 mA, es decir inferior a 400 mA, lo que supone una potencia total de 2 W.

2. ispGDS

El consumo (I_{cc}) de este dispositivo es de 40 mA para una V_{cc} de 5 V. La potencia consumida es de 0'2 W.

3. FIFOS

El consumo de las FIFOs en el peor de los casos (todas las señales conmutando) se calcula según la siguiente ecuación:

$$P_i = V_{CC} \cdot [I_{CC}(l) + (N \cdot \Delta I_{CC} \cdot dc)] + \sum (C_{pd} \cdot V_{cc}^2 \cdot f_i) + \sum (C_L \cdot V_{cc}^2 \cdot f_o) \quad (4.2)$$

donde:

- $I_{cc}(l)$ = I_{cc} según tabla proporcionada por el fabricante.
- N = número de entradas direccionadas
- ΔI_{cc} = incremento de I_{cc} .
- dc = ciclo de trabajo de las señales de entrada
- C_{pd} = capacidad de disipación de potencia
- C_L = capacidad de carga de salida
- f_i = frecuencia de entrada.
- f_o = frecuencia de salida.

El valor de $I_{cc}(l)$ para una V_{cc} de 5 V es de 4 mA, el ΔI_{cc} es de 1 mA, el valor de C_{pd} es de 53 pF, C_L se supone de 50 pF, la f_i es de 50 MHz y la f_o es de 33 MHz.

La estructura de la FIFO de salida está compuesta por dos FIFOs, una con 18 entradas y la otra con 14 entradas (para conseguir las 32 entradas de datos). Así pues la potencia total es de 3'5 W.

4. RAM

El consumo de esta memoria es de 140 mA a V_{cc} de 7 V. La potencia consumida por memoria es menor de 0'9 W. Se supondrá en el peor de los casos un consumo total, de las cuatro memorias, de 3'5 W.

5. S-LINK

Según el manual técnico la máxima potencia consumida es de 7'5 W.

6. Etapas de conversión TTL-NIM y NIM-TTL.

Mediante simulación se ha determinado el consumo de estas dos etapas:

La etapa de NIM-TTL tiene un consumo de 0'15 W.

La etapa de TTL-NIM tiene un consumo de 0'15 W. Esta etapa dependerá de una fuente de alimentación distinta al resto del sistema.

7. Osciladores

El consumo de los osciladores de cuarzo se sitúa en el peor de los casos en 50 mA, por lo que la potencia consumida total es de 0'5 W.

El consumo del 74LS56 es de 30 mA, luego la potencia es de 0'15 W.

8. Potencia total

La potencia total consumida por el sistema es de 17'3 W para la alimentación de 5 V, lo que supone un consumo de 3'46 A. Para la alimentación de -2'5 V (obtenida a partir de la de -12 V), la potencia consumida es de 0'15 W, es decir un consumo de unos 60 mA.

Fuente de alimentación

La potencia necesaria para este módulo la suministra el sistema de alimentación de VME. La corriente máxima que puede suministrar viene limitada por la máxima corriente que puede circular por un pin del conector DIN41612.

La máxima corriente depende de la temperatura, ello es debido a que la limitación viene producida por el autocalentamiento del pin del conector, que puede ser dañado si se calienta demasiado. El autocalentamiento se produce por el efecto de la resistencia de contacto. Esta resistencia de contacto puede causar flujos de corriente desiguales si varios pines se conectan juntos. Ello obliga a considerar que la corriente por pin máxima cuando se conecten varios pines en paralelo es inferior a la máxima.

Para una tensión de 5 V con 3 pines en P1/J1 y 3 pines en P2/J2 se tiene una capacidad de corriente de 3'6 A con 1 *backplane* y de 7'2 A con 2 *backplanes*. Para masa (GND) con 7 pines en P1/J1 y 4 pines en P2/J2 se tiene una capacidad de corriente de 10'5 A con 1 *backplane* y de 16'5 A con 2 *backplanes*. La tensión de -12 V con 1 pin en P1/J1 tiene una capacidad de 1'5A con un *backplane*.

La corriente requerida por el sistema es de 3'46 A para la fuente de 5 V y de 60 mA para la fuente de -12 V. Dado que el sistema está sobredimensionado se puede afirmar que sólo es necesario la conexión a P1/J1.

2.44.3.3 Diseño de la placa

Para el diseño de la placa se han utilizado los programas PCBoards de *MicroSim Corporation*, y el *SPECCTRA* de *Cooper&Chyan Technology's*; pertenecientes al entorno de trabajo DesignLab y englobados en el llamado PCBoards Autorouter.

Reglas de trazado

Dado que esta placa trabaja con frecuencias situadas entre los 33 y 50 MHz es conveniente aplicar las reglas de diseño de PCB para Alta Velocidad. Para ello se pueden consultar las referencias [15] [16] [17] [18]. Las reglas aplicadas afectan a parámetros como tamaños de pista, distancias entre pistas, trazado de pistas, condensadores de desacoplo, *layout* de osciladores, planos de masa, etc.

Las características básicas del diseño del PCB son:

- Adecuación al formato VME 6U.
- Diseño a 4 capas: 2 caras de señales, 2 para alimentación y masa.
- Clase 4.
- Diseño mixto: smd e inserción.
- La separación mínima entre conductores con una tensión de 5 V puede ser inferior a 0'004 inch.- 40 mil - (espesor de capa de 35 μ m): se ha empleado de 10 mil.
- El ancho de las pistas de alimentación para una corriente de 4 A no debe ser inferior a 0'06 inch. (espesor de capa de 35 μ m): se ha empleado plano de GND y PWR.
- el ancho de las pistas de señales para una corriente de pico de 100 mA no debe ser inferior a 0'0002 inch. - 2mil - (espesor de capa de 35 μ): se ha empleado 10mil.
- Vías de 0'015 inch. - 15mil - de diámetro o superior: se ha empleado de 20 mil.

Disposición de los componentes

La disposición de los componentes está sujeta a una serie de imposiciones:

- Disposición, tamaño y limitación en altura del S-LINK.
- Disposición de los conectores.
- Disipación térmica.

La placa del S-LINK debe estar situada en la *motherboard* (placa madre) según unas normas específicas; además estas normas indican la máxima altura que deben tener los componentes situados bajo esta placa. Todo ello implica una redistribución de los componentes por el resto de la placa.

Los conectores del dBeX32 deben estar situados en la parte inferior de la placa y por la cara de soldadura para facilitar su conexión con la tarjeta procesadora. El conector del DBV44 para el bus del puerto del C40, debe ir situado en la parte frontal de la placa, es decir en la carátula. Los leds indicadores de "placa accedida", "RBUSY" y "XONF" están situados en la parte frontal. Los conectores tipo LEMO de las señales "RBUSY" y "XONF" también deben estar situados en la parte frontal. Por último el pulsador del "RESET" y el conector de programación de las "ISP" también estarán situados en la parte frontal.

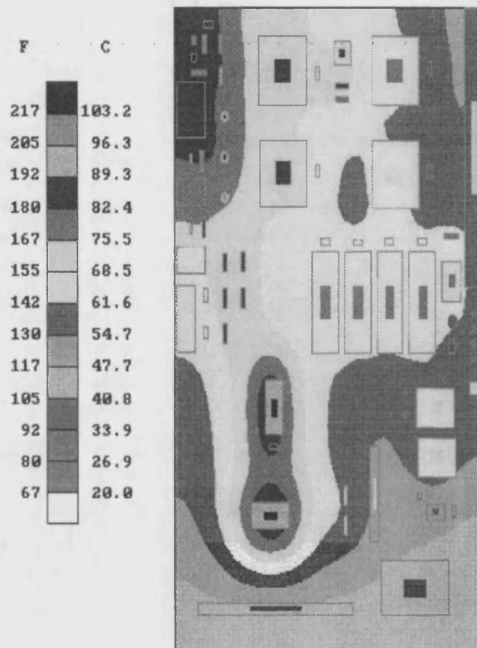
Además, se ha tenido en cuenta el factor de disipación térmica para una distribución óptima de los componentes, de tal forma que dicha disposición haga innecesario el uso de disipadores térmicos y/o ventilación forzada, o que una excesiva disipación térmica provoque calentamientos en el resto del sistema.

Cálculos térmicos

Para realizar una distribución óptima de los componentes, estudiar las variaciones térmicas de la placa y la fiabilidad del sistema, se ha utilizado el programa BETASOT-R de *Dynamic Soft Analysis, Inc.* Este programa utiliza modelos tridimensionales de campos térmicos y evalúa los componentes según el estándar MIL-HDBK-217E. La información acerca del funcionamiento del programa y las bases teóricas del mismo se pueden consultar en [19][20].

En la figura 4.9 se observa la distribución térmica del sistema. El análisis térmico se ha realizado suponiendo el sistema aislado y en recinto cerrado, observándose un promedio de temperatura de funcionamiento de 68°C con unas zonas calientes en la parte superior izquierda (zona de conectores) y en las FIFOs. Este último dato implica que durante la puesta en marcha del sistema se deben vigilar con especial atención estas zonas y en el caso de cumplirse estas previsiones, emplear disipadores o ventilación forzada para disminuir la temperatura de trabajo.

Tras todos los análisis realizados, el aspecto final de la placa ya montada es el mostrado en la figura 4.10.



FFIGURA 4.9 Análisis térmico del sistema.

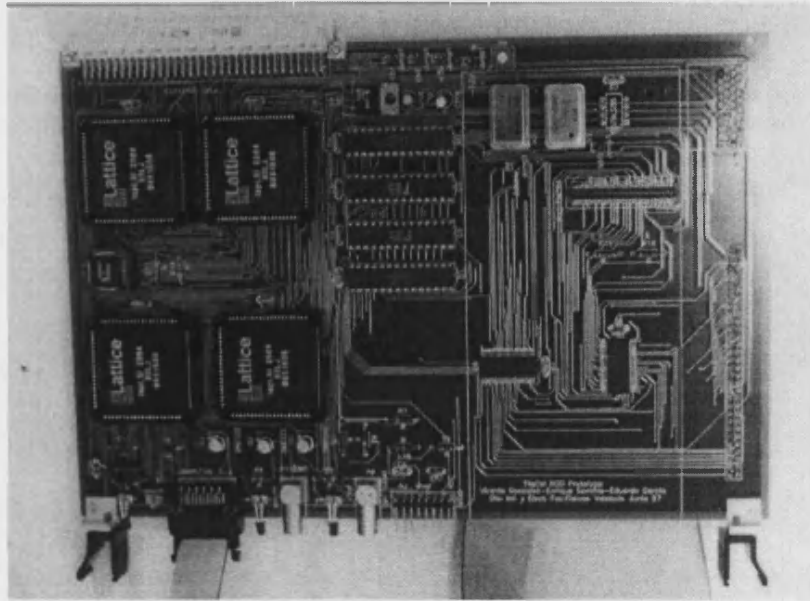


FIGURA 4.10 Aspecto final del módulo interface S-LINK.

3. DESCRIPCIÓN DE LAS PRUEBAS Y RESULTADOS

A continuación se describirán las pruebas realizadas antes y después del diseño y montaje del prototipo. Las pruebas realizadas con anterioridad al diseño se refieren a la determinación del número de procesadores digitales de señal C40 necesarios para lograr el objetivo de maximizar el número de canales procesados en cada módulo. Aunque lo normal sería pensar que cuantos más mejor, la arquitectura interna de la tarjeta DBV44 limita este número.

Las pruebas realizadas con posterioridad se refieren al diseño del software para la adaptación del prototipo al sistema de adquisición instalado en el CERN. Estas pruebas se realizaron en dos fases: una local mediante el empleo de un sistema portátil a escala del sistema instalado en el CERN, y otra en el mismo sistema leyendo datos reales del detector.

3.1 DETERMINACIÓN DEL NÚMERO DE DSPS

Para la determinación del número de DSP la primera tarea ha sido el desarrollar el programa de tratamiento de datos. Dicho programa se ha realizado en ensamblador para poder optimizar el código y ganar velocidad.

El programa implementado realiza un escalado de los datos multiplicando cada uno de ellos por una constante almacenada en una tabla precargada en la memoria del DSP.

El setup empleado para la realización de estas medidas se muestra en la figura 4.11.

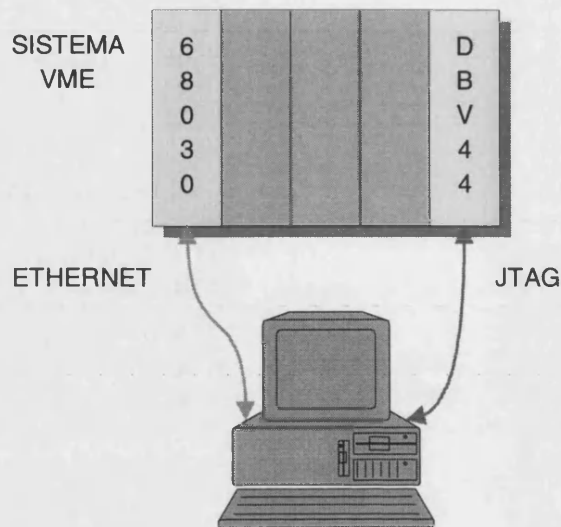


FIGURA 4.11 Setup empleado para la determinación del número de DSPs necesarios.

La programación de los DSP se ha realizado en un PC y después se ha transferido a los procesadores a través del *host* del VME, en este caso un procesador 68030 de Motorola con OS-9 [21] como sistema operativo.

Las primeras pruebas se han realizado con un solo DSP y se ha empleado fibra óptica para simular la lectura de los datos procedentes del primer nivel. El protocolo empleado para la transmisión de los datos por la fibra óptica fue TAXI [22].

Se ha medido el tiempo necesario para procesar un total de 30 canales de 2 bytes más 8 bytes de cabecera, correspondientes al total de canales de adquisición disponibles en un primer momento en las instalaciones del detector. El resultado ha sido de 20'15 μ s. Como se recordará, desde el primer nivel de trigger, el periodo de salida de datos es de 10 μ s por lo que con un solo procesador no sería suficiente.

Por ello, se ha variado la configuración de la tarjeta DBV44 y se han instalado 3 DSP (el procesador del TAXI y dos procesadores, PI y PII situados en un único módulo TIM), repartiéndose las tareas entre ellos. Se emplearon dos DSPs para el procesamiento de los datos y el tercero se empleó para el procesamiento de la cabecera. La disposición de los tres DSP es la mostrada en la figura 4.4 o esquemáticamente en la figura 4.12. La distribución de tareas puede observarse en la figura 4.16.

Uno de los procesadores se encarga de leer los datos desde TAXI y almacenarlos alternativamente en dos bancos de memoria (bloque 1 y bloque 2). Desde cada banco y mediante DMA los datos son transferidos al procesador PI el cual separa la cabecera y la pasa a PII. En ese momento, en paralelo, se procesan los datos y la cabecera. El resultado se envía de nuevo hacia el procesador encargado de recibir y enviar los datos a través de TAXI. Durante el procesamiento de uno de los bloques, se carga el siguiente de forma automática, de manera que en cuanto se termina de procesar uno, ya se dispone de los datos para procesar el siguiente.

Mediante esta configuración fue posible leer y procesar los 30 canales en 9 μ s, inferior a los 10 μ s con que salen los datos desde el primer nivel.

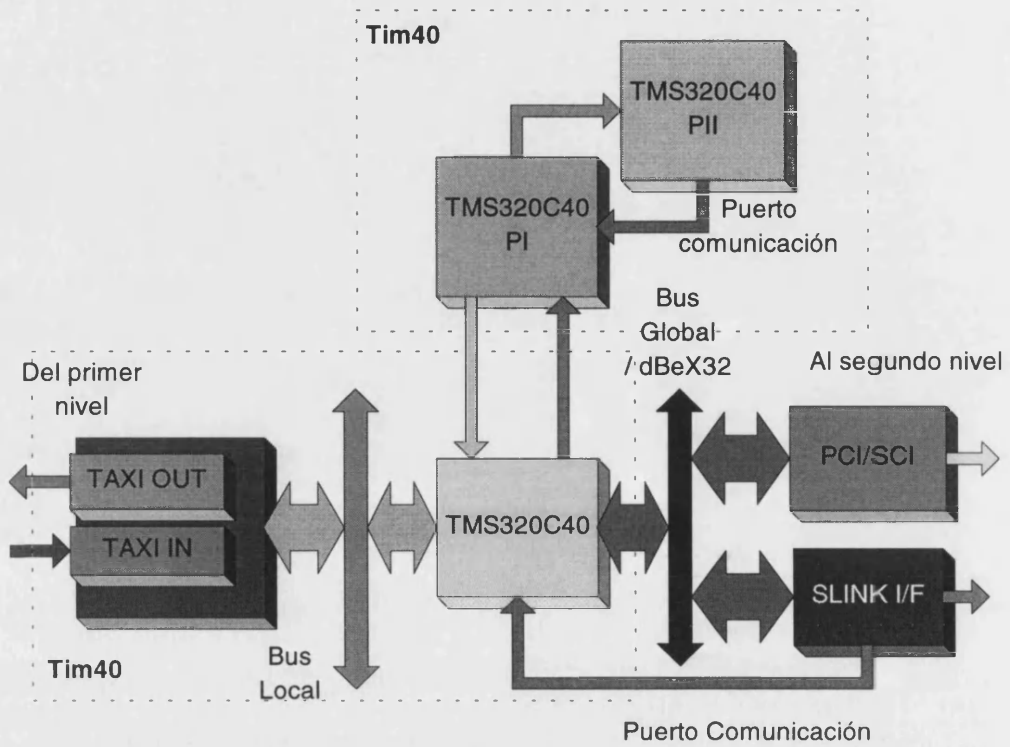
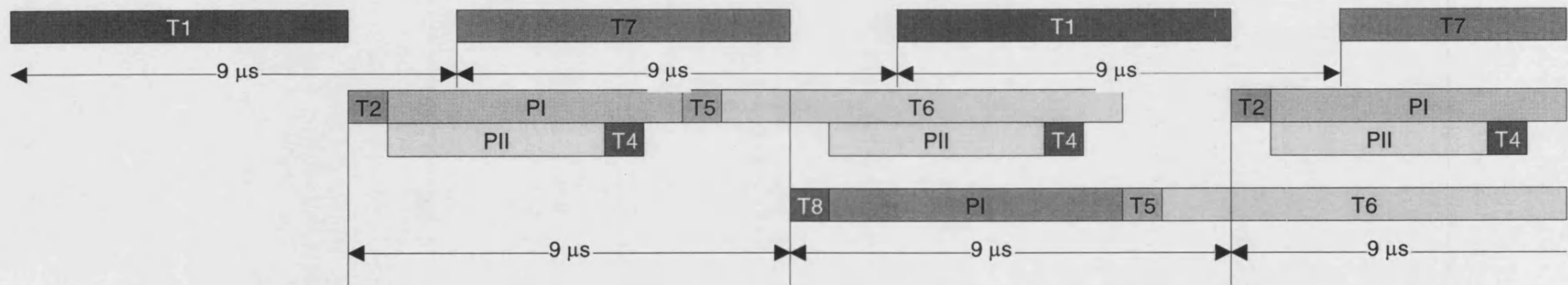


FIGURA 4.12 Disposición de los tres DSP y sus conexiones.



- T1- Lectura de datos desde TAXI a Bloque 1(6'8 μs)
- T2- Datos del C40-TAXI (Bloque 1) a PI (0'8 μs)
- PI- Procesado 30 canales (6 μs)
- PII- Procesado de cabeceras (4'35 μs)
- T4- Envio resultado cabeceras a PI (0'8 μs)
- T5- Envio resultado cabeceras a C40-TAXI (0'8 μs)
- T6- Envio resultado procesado a C40-TAXI (8'2 μs)
- T7- Lectura de datos desde TAXI a Bloque 2 (6'8 μs)
- T8- Datos del C40-TAXI (Bloque 2) a PI (0'8 μs)

FIGURA 4.13 Diagrama temporal de la ejecución de tareas entre los tres DSPs.

A partir de la disposición temporal de tareas podemos obtener el diagrama de paralelismo del sistema a fin de obtener los valores de paralelismo medio, incremento de velocidad (*speedup*) y eficiencia a partir de las expresiones vistas en el capítulo 2. La figura 4.17 muestra el diagrama de paralelismo del sistema.

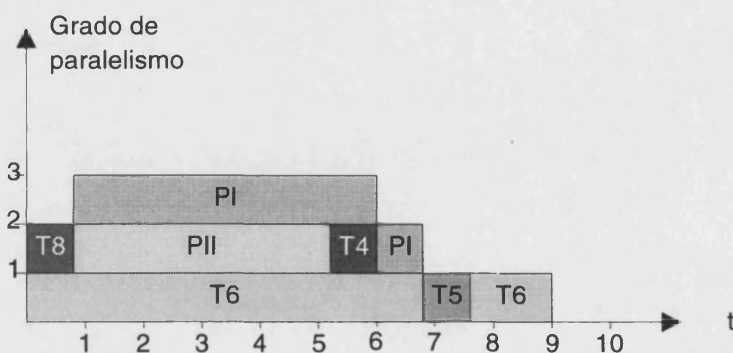


FIGURA 4.14 Diagrama del paralelismo del sistema con tres procesadores.

El paralelismo medio se obtiene mediante la siguiente ecuación [23]:

$$A = \frac{\sum_{i=1}^m i \cdot t_i}{\sum_{i=1}^m t_i} \quad (4.3)$$

donde m es el número máximo de procesadores empleados y t_i el tiempo durante el que el grado de paralelismo es i .

Aplicando esta expresión, el paralelismo medio del sistema es:

$$\begin{aligned} A &= (1 \cdot (T5 + (T6 - T8 - PI))) + 2 \cdot (T8 + (PI - PII - T4)) + 3 \cdot (PI + T4) / 9 = \\ &= (1 \cdot (0'8 + (8'2 - 0'8 - 6))) + 2 \cdot (0'8 + (6 - 4'35 - 0'8)) + 3 \cdot (4'35 + 0'8) / 9 = 20'95 / 9 = 2'33. \end{aligned}$$

Como se observa el paralelismo medio es menor que el máximo número de procesadores al no emplearse todos los procesadores durante todo el periodo de procesado.

Puesto que nuestro sistema no trabaja en modo totalmente secuencial o totalmente paralelo, no podemos aplicar la ley de Amdhal, sino que hemos de acudir a la expresión dada por Hwang (ecuación 2.2) y que repetimos aquí por comodidad.

$$S(n) = \frac{\sum_{i=1}^m W_i}{\sum_{i=1}^m \frac{W_i}{i} \cdot \left\lceil \frac{i}{n} \right\rceil + Q(n)} \quad (4.4)$$

donde, como se vio entonces, W_i es el trabajo que se realiza cuando se usan i procesadores de los n existentes en el sistema y viene definido como $W_i = i \cdot t_i \cdot A$, donde

A es el paralelismo medio del sistema, m es el número máximo de procesadores necesarios para ejecutar la tarea o, lo que es lo mismo, el grado máximo de paralelismo, $Q(n)$ es la carga introducida por las comunicaciones y $\lceil x \rceil$ la operación que devuelve el menor entero mayor o igual que el número real positivo x .

Para nuestro caso, $m = 3$, $n = 3$, $W_1 = 2'2A$, $W_2 = 3'3A$, $W_3 = 15'45A$. Si consideramos $Q(n) = 0$, sustituyendo en (4.4) nos queda un valor de *speedup* igual a 2'33. Como se ve, el valor es idéntico al calculado para el paralelismo medio y coincide con el valor asintótico del *speedup* (valor para el caso en que el número de procesadores disponibles fuera infinito). Esto es así porque no se han considerado las latencias de comunicación y otras penalizaciones.

En la realidad, el *speedup* medido del sistema, cociente entre el tiempo que tardaría la ejecución secuencial de todas las operaciones y los 9 μ s que tarda el sistema cuando se emplean tres procesadores, es de 2'24. La diferencia entre el valor obtenido anteriormente y éste debe atribuirse a las latencias de comunicación que podemos obtener a través de la ecuación (4.4), resultando $Q(n) = 0'35A = 0'8155$.

La eficiencia del sistema según la ecuación (2.6) es de $E(3) = 2'24/3 = 0'75 < 1$ como es normal al no ser un sistema ideal.

En la tabla 4.x se resumen los principales resultados de las medidas realizadas.

	T_1	T_3	S_∞	$S(n)$	$E(n)$
1 procesador	20'15	-	1	1	1
3 procesadores	-	9	2'33	2'24	0'75

TTABLA 4.3 Resumen de los parámetros de *speedup* y eficiencia del sistema procesador.

3.2 INTEGRACIÓN DEL PROTOTIPO

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

El proceso de integración se dividió en dos etapas. La primera 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. Esta tarea se realizó mediante un equipo portátil que reproducía a escala el sistema de adquisición.

Una vez realizada esta tarea, 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 2004. 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.

3.2.1 Sistema StandAlone

Denominamos sistema *StandAlone* al sistema portátil en el que se integró por primera vez el prototipo y donde se adaptó el software de adquisición. La estructura de este sistema se muestra en la figura 4.15.

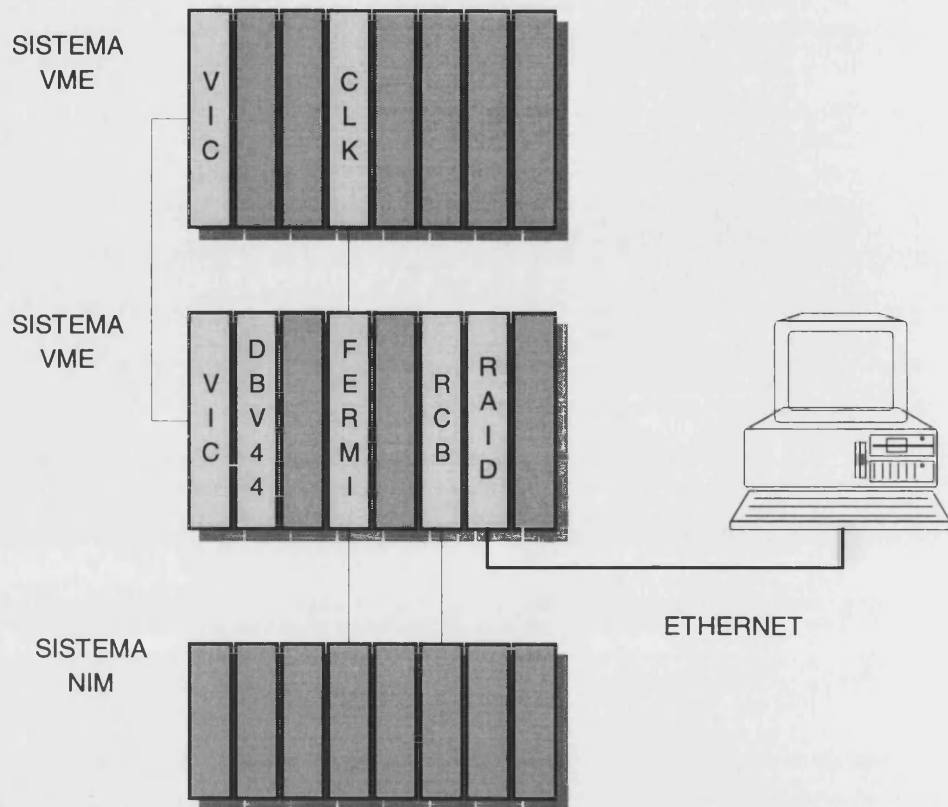
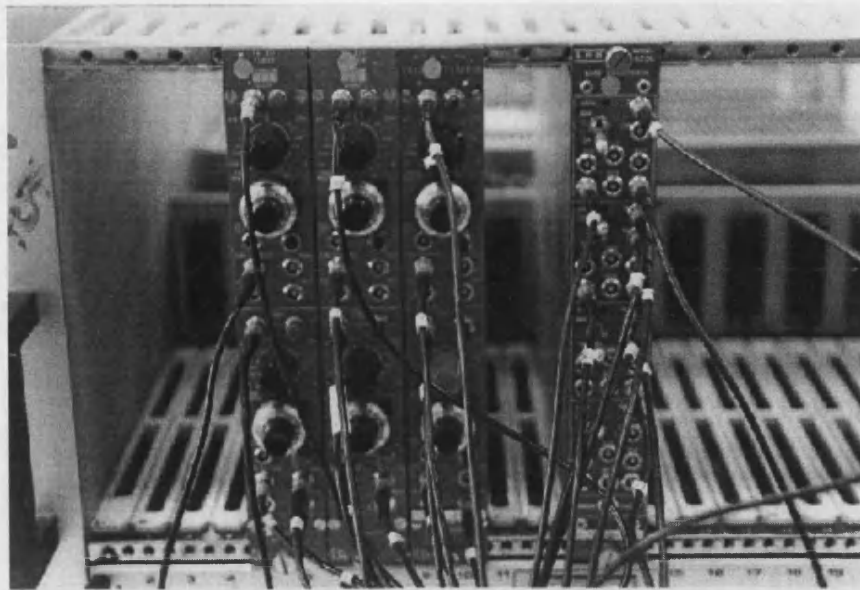


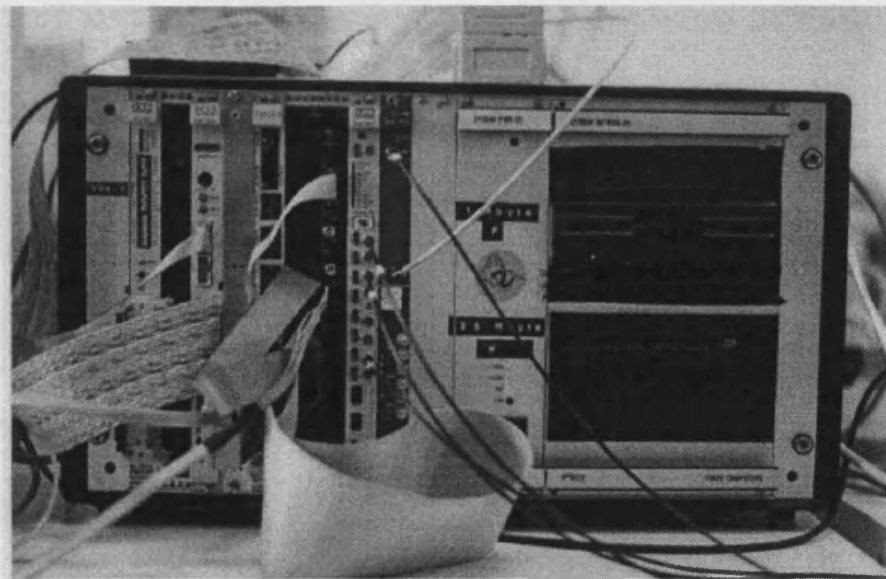
FIGURA 4.15 Setup para la integración StandAlone.

Como se puede observar el setup se compone de dos sistemas VME [24] y un sistema NIM (Nuclear Instruments Measurements) [25] (figura 4.16). El sistema NIM es el encargado de simular, mediante lógica, las señales de control del acelerador donde se está probando el subdetector. Es el encargado de generar la señal que indica cuándo se produce el haz y por lo tanto hay que iniciar la adquisición. Asimismo impide que llegue al sistema un nuevo comienzo de adquisición si no se ha terminado de procesar el anterior.

Los dos sistemas VME están interconectados a través de tarjetas VIC8251F [26] que realizan la comunicación VME-VME de forma transparente. En uno de los chasis, por cuestiones de consumo, se encuentra el generador de reloj de 40 MHz necesario para la digitalización de la información. En el segundo chasis se encuentra la tarjeta procesadora DBV44, la tarjeta digitalizadora de 3 canales a 40 MHz, FERMI [27], una tarjeta generadora de interrupciones VME, RCB 8047 CORBO [28] y la CPU donde opera todo el software de adquisición, una tarjeta RAID8235 [29] con un procesador RISC R3000 y con EP/LX (una adaptación de LynxOS [30] para esta tarjeta) como sistema operativo. El control de todo el sistema se realiza a través de Ethernet desde un host (PC o estación de trabajo) (figura 4.17).



FFIGURA 4.16 Detalle del sistema NIM para la generación de las señales de sincronismo.



FFIGURA 4.17 Detalle del sistema VME.

El sistema funciona como se describe a continuación: la lógica NIM genera la sincronización; una de las señales generadas sirve como disparo para el inicio de la digitalización, otras señales se emplean a través de la tarjeta RCB para la generación de interrupciones VME que indican el comienzo de la toma de datos, la llegada de los haces, la colisión de los mismos y la finalización de ese cruce de haces y de la toma de datos. Ante todas esas interrupciones, el procesador RAID responde de la manera adecuada: leyendo el contenido de la tarjeta digitalizadora, almacenando los datos, etc.

Como primera diferencia respecto al sistema que se empleó para evaluar el número de DSPs encontramos que en el sistema StandAlone los datos no llegan a través de fibra óptica a la tarjeta procesadora, sino que el flujo de datos va de la tarjeta digitalizadora a la CPU y de ahí a la DBV44. Además, los requerimientos temporales de la adquisición durante la fase de prueba del detector no son tan estrictos como lo serán en el futuro y, por lo tanto, se optó por variar la configuración de la tarjeta dejando un solo DSP.

Después de ser procesados, los datos son leídos de nuevo por la CPU y almacenados en disco. Evidentemente, el hecho de que los datos tienen que pasar por la tarjeta procesadora enlentece el sistema. De hecho, los datos deben ser leídos primero por la CPU ya que la tarjeta procesadora DBV44 no dispone de la funcionalidad MASTER para poder hacerlo por ella misma.

Puesto que el sistema se enlentece, el objetivo de esta prueba, a parte de la integración del prototipo, es asegurar que el incremento en el tiempo no limita seriamente la adquisición de datos. Para que esto no sea así, el sistema debe ser capaz de grabar en disco, al menos, 70 colisiones por cada cruce de haz.

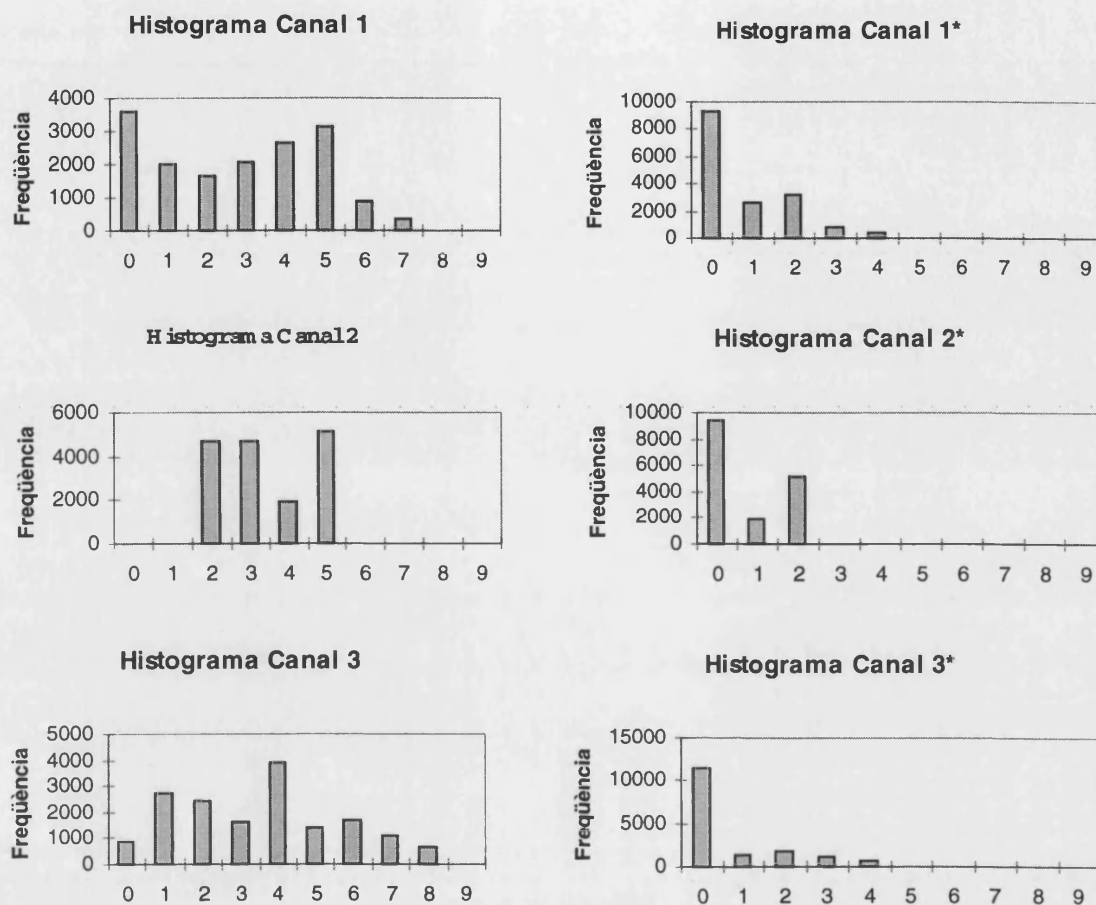
Las pruebas se han realizado para dos algoritmos de procesamiento diferentes. En el primero, se realiza una calibración de los datos consistente en la multiplicación de cada uno por una constante pregrabada en la memoria del DSP. El segundo consiste en la reordenación de los canales siguiendo directrices físicas para hacer más sencillo y rápido el procesamiento posterior. Para emular las condiciones del sistema de prueba del detector (60 canales de datos a leer, 40 muestras por canal y 2 bytes por muestra) se ha establecido un tamaño del bloque de datos de 4800 bytes (1200 palabras de 32 bits) y 8 bytes de cabecera, y se han medido los tiempos de procesamiento para cada uno de los algoritmos. La tabla 4.4 muestra los resultados.

Tipo de procesado	Salida datos	Tiempo por bloque (ms)	Colisiones almacenadas
Calibración	VME	6'3	81
	dBeX32	2'4	119
Ordenación	VME	4'1	99
	dBeX32	1'93	126

TABLA 4.4 Tiempos de procesamiento para diferentes algoritmos.

La columna de **Salida datos** hace referencia a que se ha medido también el efecto de reenviar los datos, una vez procesados, hacia la tarjeta CPU a través del VME, o hacia el bus auxiliar dBeX32 al que se conecta la tarjeta interface S-LINK. Como puede observarse en la columna del número de colisiones almacenadas por cruce de haz, en todos los casos estamos por encima de las 70 requeridas.

Se ha realizado también un tercer algoritmo para reducir el error de cero del digitalizador. En efecto, se comprueba que en ausencia de señal la señal digitalizada no es cero, sino que existe un pequeño error. Por ello, se ha diseñado un algoritmo para minimizarlo basado en la resta, en tiempo real, entre los valores leídos y un valor medio de error obtenido estadísticamente de los datos almacenados previamente. El resultado ha sido una corrección casi total de este error como muestran los histogramas de la figura 4.18 para cada uno de los 3 canales del digitalizador.



FFIGURA 4.18 Resultado del procesado de corrección de cero para el digitalizador.

3.2.2 Sistema TestBeam

El sistema TestBeam es el montado en los laboratorios CERN en Ginebra para la realización de las pruebas de funcionamiento de los diferentes subdetectores. En este tipo de pruebas, se emplean haces de partículas conocidas con determinada energía y se observa la respuesta del subdetector para evaluar sus prestaciones y si cumple con los requerimientos con los que se debe construir.

Para poder verificar estas cuestiones, es necesario leer y procesar la información del subdetector. Esto se realiza mediante sistemas de adquisición y procesado *ad hoc* ya que, las condiciones en las que se realizan estas pruebas no son las mismas que las que se tendrán cuando el detector se instale en el experimento para el que se construye. Sin embargo, aunque no es el principal objetivo del sistema TestBeam, el sistema de adquisición se emplea también para realizar pruebas de los módulos que después compondrán el sistema final, aprovechando que los datos que se leen tienen un sentido físico y que, por lo tanto, puede comprobarse el correcto funcionamiento de los mismos.

A continuación describiremos de manera breve el sistema de TestBeam donde se ha probado el prototipo, las pruebas realizadas y los resultados obtenidos.

3.2.2.1 Descripción del TestBeam para TileCal

La zona experimental donde se instala el detector TileCal para las pruebas de TestBeam es una nave de grandes dimensiones hasta donde puede llegar un haz de partículas procedente de otro de los aceleradores disponibles en el CERN, el SPS (Super Proton Synchrotron).

La primera consideración que hay que tener en cuenta es que, desde el momento en que el acelerador no es aquél en el que se va a emplear el detector, las condiciones de prueba no van a ser las finales. Entonces, ¿para qué nos sirven este tipo de pruebas? La misión de las pruebas en TestBeam es asegurar la funcionalidad de los sistemas empleados y su correcto funcionamiento. Para probar el funcionamiento de un detector se podrían simplemente simular las señales, puesto que son perfectamente conocidas, es más, esto se realiza para calibrar el detector. Sin embargo, siempre es aconsejable emplear señales reales obtenidas de la detección de partículas, incluso si su tasa de producción no es comparable con la final, ya que con esos datos se puede inferir si el sistema funcionará correctamente cuando esté en condiciones reales.

Por otra parte, tampoco se emplean partículas resultantes de la colisión de dos haces, sino un haz que colisiona con un determinado material y que provoca un determinado tipo de partículas con energía perfectamente conocida, lo que permitirá comparar los resultados entre lo medido en el detector y la realidad.

Una tercera consideración es que, ya que la fecha de comienzo del experimento está fijada para el 2004, las pruebas se realizan en algunos casos con porciones del detector, puesto que todavía no está totalmente construido o incluso porque su construcción final depende del resultado de las pruebas.

Por lo tanto, el sistema de TestBeam para TileCal se caracteriza por tener, en primer lugar, una "frecuencia de colisión" diferente a la del experimento ATLAS/LHC, y en segundo lugar (y muy relacionado con la primera diferencia y la tercera consideración) una menor tasa de datos.

Para el TestBeam de TileCal, el "periodo de colisión" es de 14'5 segundos (580 millones de veces superior al del experimento ATLAS/LHC). Al igual que será en el futuro, las partículas llegan al detector agrupadas en lo que se conoce como *bursts*, y su llegada y la finalización de la misma vienen indicadas por señales de sincronización del acelerador: *SOB* (*Start_Of_Burst*) y *EOB* (*End_Of_Burst*). La figura 4.19 muestra la relación entre estas señales.

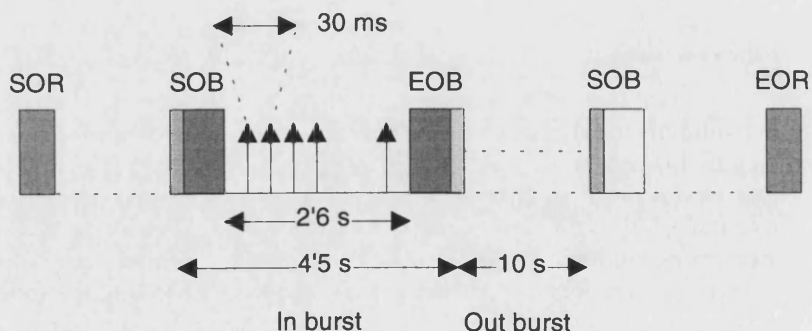


FIGURA 4.19 Relación entre las señales de sincronización del acelerador SPS.

La señal *SOR* (*Start_Of_Run*) comienza la adquisición y es generada por el software; la señal *EOR* (*End_Of_Run*) la finaliza cuando se han adquirido un número determinado de sucesos o cuando se produce un error de funcionamiento. Entre esas dos señales se aceptan los burst que lleguen indicados por las señales *SOB* y *EOB*. Cada señal va seguida o precedida de un tiempo de aproximadamente un segundo destinado a realizar operaciones como la calibración del detector. Entre las dos señales es donde se sitúan las partículas durante un tiempo de 2'6 s.

El haz de partículas, con esta sincronización, atraviesa cuatro dispositivos centelleadores acoplados a fotomultiplicadores, empleados para indicar el paso de partículas y, por lo tanto, la activación del sistema de adquisición. Tras esta etapa, el haz incide en el detector.

En el sistema de TestBeam donde se ha probado el prototipo, el detector está compuesto por dos módulos de 3 metros de longitud correspondientes al barril extendido y cinco módulos de 1 metro, construidos como se vio en el punto 2.1, sobre una mesa capaz de girar en los tres ejes del espacio para observar la respuesta del detector a la incidencia de haces desde diversos ángulos. Los módulos se hallan dispuestos como se muestra en la figura 4.20. El prototipo trabaja sólo con los canales de los dos módulos del barril extendido puesto que son los que emplean los digitalizadores FERMI.

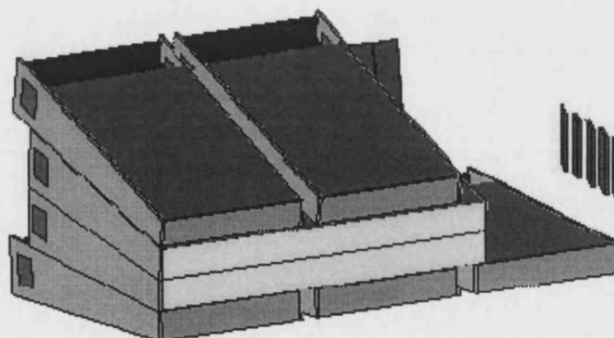


FIGURA 4.20 Disposición de los módulos del detector en el sistema TestBeam.

Junto a los módulos, sobre la mesa, se halla situada la electrónica de calibración del detector y las tarjetas de digitalización para minimizar la atenuación de la señal y el ruido. Los datos leídos se envían a una sala adjunta separada por un muro de cemento como protección contra la radiación.

El diagrama de bloques del sistema de adquisición es el mostrado en la figura 4.21. Como se observa está compuesto de una serie de tarjetas tanto en el estándar VME como en el NIM, y su estructura es semejante a la del sistema StandAlone pero ampliada para poder realizar la lectura de 60 canales del detector, agrupados de 3 en 3 en tarjetas digitalizadoras FERMI.

La conexión entre los sistemas VME se realiza a través de tarjetas VIC8251 y la sincronización con el acelerador se logra a través de un sistema NIM que provoca interrupciones en el sistema VME principal a través del un módulo RCB 8047 CORBO como ya se explicó en el sistema StandAlone.

El control general de la adquisición se realiza en la tarjeta CPU RAID 8235 como se hacía en el sistema StandAlone, y los datos se almacenan en disco local. Además en el sistema TestBeam es posible realizar el monitoraje de los datos conforme se van

adquiriendo. Para ello, se emplea una estación de trabajo HP conectada a la cadena VIC, que permite la comunicación de los diferentes sistemas VME. Además, esta estación de trabajo se encarga del control de todo el sistema de adquisición. Los resultados del análisis *on-line* realizado en la estación de trabajo se almacenan en otro disco local.

La principal diferencia con el sistema StandAlone, aparte del incremento en el número de canales, está en la tarjeta CDR (*Central Data Recorder*) que envía los datos adquiridos a través de Ethernet hasta un sistema remoto para su almacenamiento en cinta magnética con vistas a su posterior análisis *off-line*.

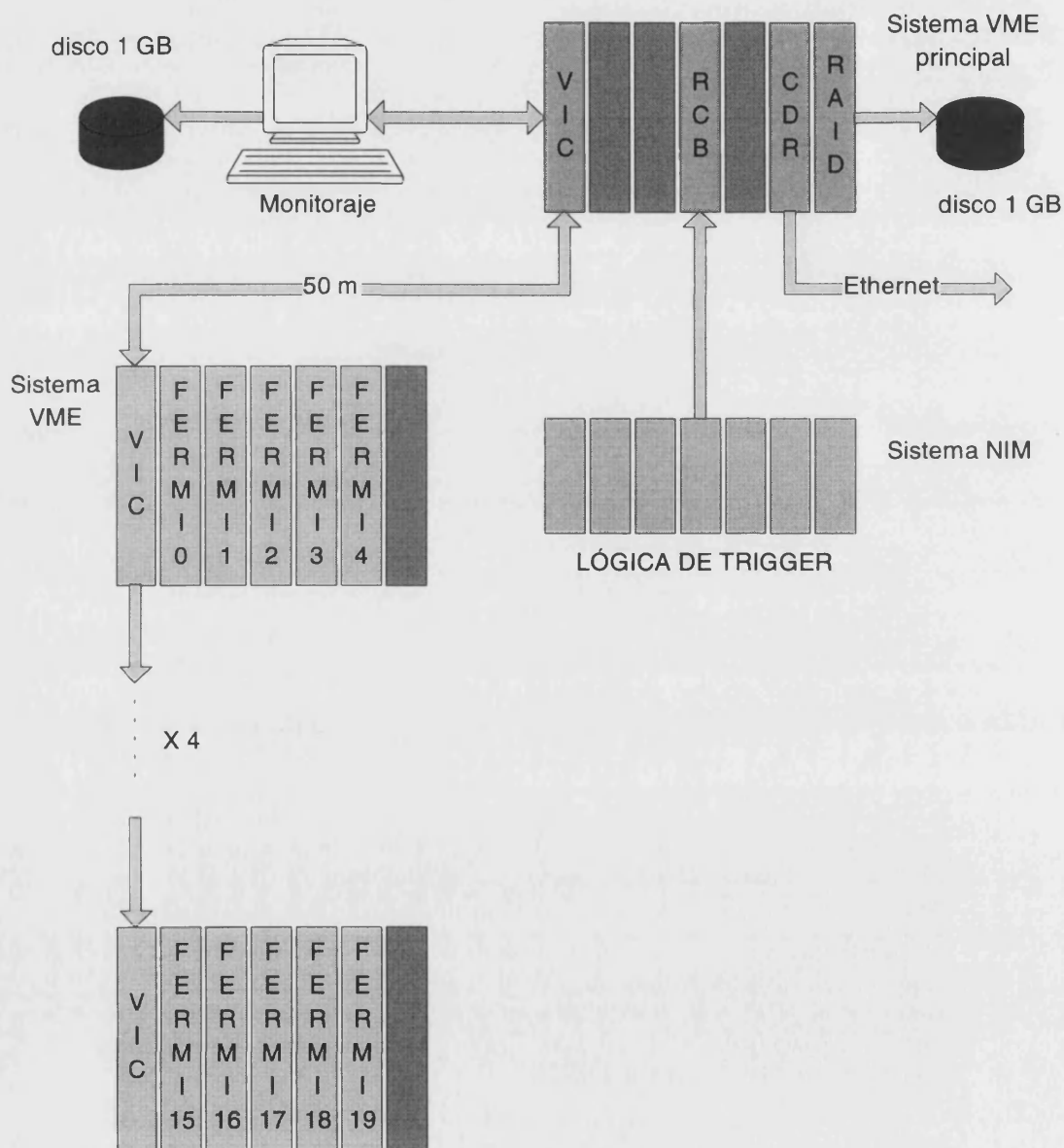


FIGURA 4.21 Diagrama del sistema de adquisición del TestBeam donde se integró el prototipo.

Como también se observa, el medio de transmisión de los datos es totalmente eléctrico sin existir fibras ópticas para enviar los datos desde los digitalizadores. Tampoco en este caso existen niveles de selección, sino que el sistema adquiere todos los datos posibles. El límite está marcado por el tiempo que se tarda en leer todos los digitalizadores, formatear el dato siguiendo una estructura creada a tal efecto, almacenarlo en cinta y enviarlo hacia el CDR. En este sistema no existe ningún tipo de procesado *on-line* de los datos salvo el realizado en la estación de trabajo más orientado a verificar que los datos se están leyendo y a extraer información general sobre la adquisición que al procesado de los mismos. La figura 4.22 muestra fotografías del sistema de adquisición instalado en el CERN.

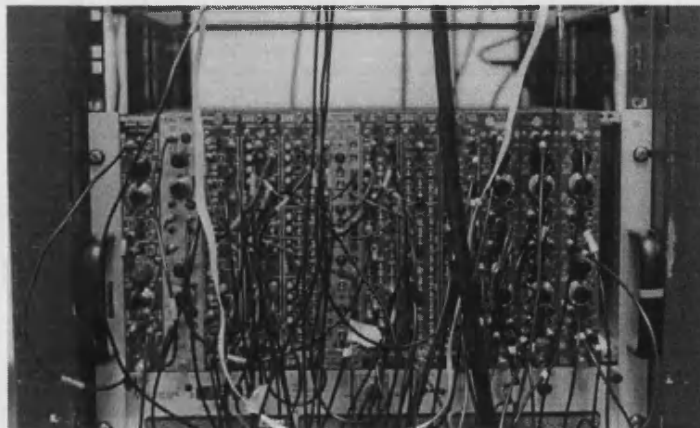
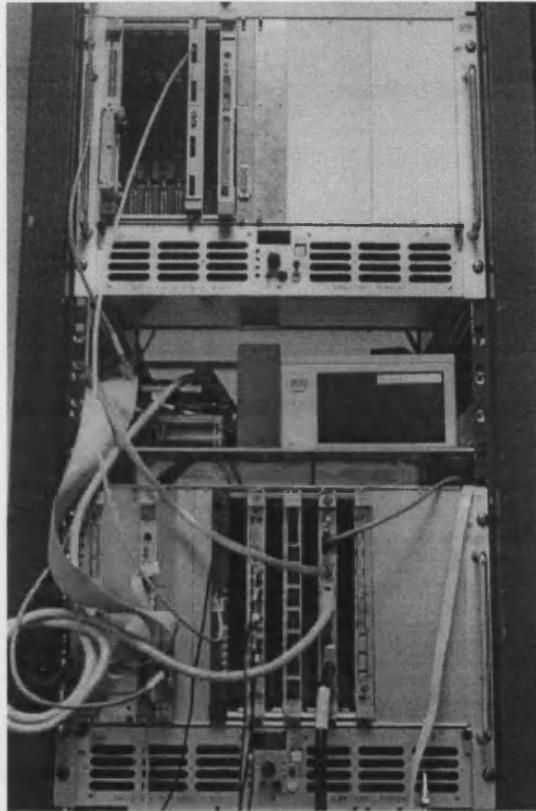


FIGURA 4.22 Detalle de los sistemas VME y NIM instalados en el CERN.

Evaluación del factor de calidad requerida

Como se ha comentado, el sistema lee la información de 60 canales digitalizados, cada uno compuesto de 40 muestras de 2 bytes. Toda esta información se lee cada vez que los cuatro dispositivos centelleadores tienen señal y el sistema no está ocupado guardando algún suceso anterior. En este caso, no existen niveles de jerarquía, y todo el proceso de adquisición lo realiza el procesado RAID quedando los datos almacenados en cinta para su posterior análisis.

Para este caso, se desea adquirir como mínimo 70 sucesos/s. Esto implica que en cada burst, se deberán leer, al menos, 182 sucesos. Como en cada burst el tiempo durante el que hay partículas son 2'6 segundos, el ancho de banda necesario, AB_{ib1} , será:

$$AB_{ib1} = \frac{182 \cdot 60 \cdot 40 \cdot 2}{2'6} \approx 340 \text{ Kbytes/s} \quad (4.5)$$

El sistema deberá leer y procesar cada suceso en $1/70 = 14 \text{ ms}$, para estar listo para el siguiente. Si durante ese periodo los dispositivos centelleadores detectan el paso de partículas, el sistema no las registra al estar ocupado.

La capacidad de procesado requerida la obtenemos a partir del número de canales a procesar (60), el tiempo disponible para hacerlo (14 ms) y el número de operaciones a realizar por cada canal.

Como en este caso no se realiza ningún tipo de procesado sino que simplemente se ordenan los datos según un formato predefinido, el número de operaciones a realizar no es demasiado elevado, y se estima, a partir del código ensamblador empleado, en 400 por canal. Por tanto, la capacidad de procesado necesaria, CP_{ib1} , es de:

$$CP_{ib1} = \frac{60 \cdot 400}{14 \cdot 10^{-3}} \approx 2 \text{ MIPS} \quad (4.6)$$

Por tanto, el factor de calidad requerida del sistema, Q_{ib1} , es:

$$Q_{ib1} = AB_{ib1} \cdot CP_{ib1} = 0'34 \cdot 2 = 0'68 \text{ MIPS} \times \text{MB/s} \quad (4.7)$$

Como ya sabemos, el sistema se ha implementado empleando un sólo procesador RAID con una capacidad de procesado de 18'3 MIPS y un ancho de banda de 100 MB/s []. Por lo tanto, el factor de calidad del sistema empleado, Q_{ib} , es superior al requerido y es igual a:

$$Q_{ib} = 18'3 \cdot 100 \approx 1830 \text{ MIPS} \times \text{MB/s} \quad (4.8)$$

3.2.2.2 Inclusión del prototipo

Puesto que, como hemos visto, no existe la comunicación óptica de los datos sino que se realiza a través de VME, se ha tenido que adaptar el prototipo a las nuevas condiciones. Para ello, y con el fin de no interferir con el sistema de adquisición ya instalado, se ha decidido diseñar una cadena de adquisición parásita paralela a la

existente que obtenga los datos de la primera y los pase a través del prototipo donde se realizará un procesamiento adecuado de los datos.

La adaptación incluye el empleo de una CPU PowerPC 604 bajo LynxOS para la realización del control del sistema de adquisición parásito, además, se ha eliminado la tarjeta interface S-LINK al no existir segundo nivel hacia donde enviar los datos. Esta CPU debe comunicarse con el CPU RAID 8235 y sincronizarse con ella. Esto se ha logrado mediante la realización de un software adecuado sobre la tarjeta PowerPC. Este software, además de lograr el control del sistema, debe ser el encargado de recibir los datos una vez procesados, formatearlos de acuerdo con el estándar CERN y almacenarlos en disco local. Además, igual que ocurre con la estación de trabajo, debe ser posible realizar el monitoreo *on-line* de los datos que se están leyendo.

La estructura modificada del sistema TestBeam se muestra en la figura 4.23. El flujo de datos procede de la siguiente manera:

- La información de SOR, SOB, EOB y EOR es escrita directamente por el procesador RAID en la memoria del PowerPC
- Los datos leídos de los digitalizadores se escriben en la memoria del procesador C40 de la tarjeta DBV44 donde se procesan y de donde son leídos por el PowerPC (recordemos que la tarjeta DBV44 no tiene capacidad para ser MÁSTER del bus VME).

Con la información de cada SOR, SOB, EOB y EOR, el PowerPC genera un fichero donde se guarda esta información con el formato estándar ATLAS. Con toda la información recibida de la DBV44, una vez procesados los datos, se genera otro fichero que también se almacena.

El procesamiento a realizar en la tarjeta DBV44 se ha ajustado a las necesidades de la colaboración. Antes de la inclusión del prototipo, los datos eran almacenados en cinta para su posterior análisis. Éste consiste en la estimación de la energía depositada por las partículas en el detector. Para la realización de este cálculo existen varios métodos:

- Sumar todas las muestras de cada canal dentro de una ventana definida para minimizar el ruido. Para este caso, se sumaban 14 muestras, de la 10 a la 24.
- Usar de cinco a siete muestras alrededor de la muestra de valor máximo y aplicar los coeficientes adecuados para corregir efectos de la forma del pulso obtenido en el fotomultiplicador, etc.
- Usar sólo el pico de la señal. Este es el mejor método pero tiene el problema de que la muestra de valor máximo no tiene porqué ser el pico, debido a que los digitalizadores están continuamente muestreando. El pico puede ser obtenido mediante interpolación parabólica de la señal alrededor de la muestra de valor máximo.

En cualquiera de los casos hay que tener en cuenta el error de cero del digitalizador que deberá ser corregido en todo momento.

Para las pruebas realizadas se determinó que se seleccionaría el valor de la muestra máxima y las dos laterales, y se corregiría el error de cero mediante la estimación del valor medio del mismo a medida que se adquiría la señal. De esta manera, de las 40 muestras por canal tomadas, el procesamiento deja únicamente 5.

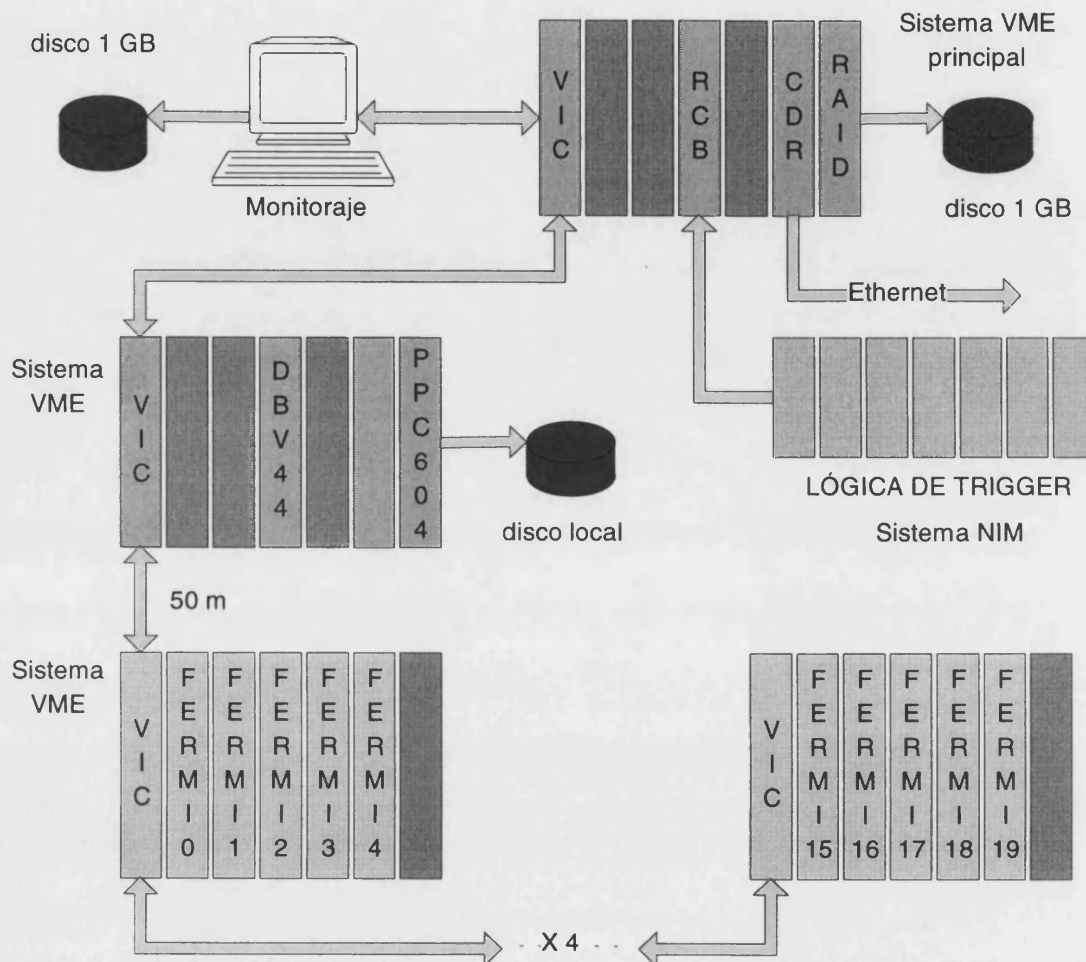


FIGURA 4.23 Diagrama del sistema de adquisición de datos tras la inclusión del prototipo.

Evaluación del factor de calidad requerida

Con la inclusión de la tarjeta procesadora DBV44 y de la CPU PowerPC, el sistema queda modificado. Para ver cómo afecta esta modificación al factor de calidad requerida, vamos a calcularlo ahora en este nuevo caso.

Evidentemente se trata de un sistema serie en el que el flujo de datos, como ya se ha comentado, pasa primero a través de la CPU RAID donde no va a sufrir ningún tipo de procesado ni formateo, sino que se deriva hacia la CPU PowerPC en el caso de que se trate de SOR, SOB, EOB y EOR o hacia la tarjeta DBV44 cuando sean datos leídos del detector. La figura 4.24 muestra esquemáticamente el flujo de los datos.

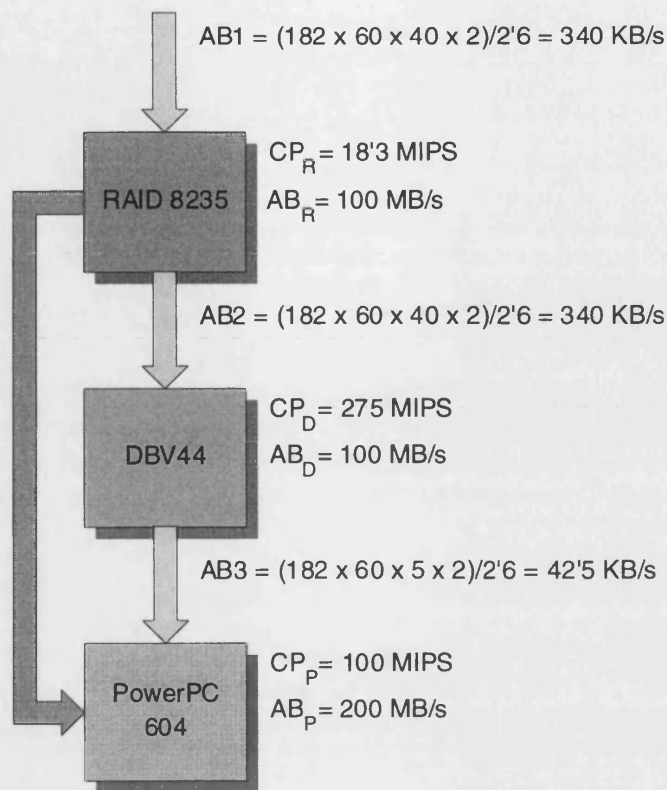


FIGURA 4.24 Diagrama del flujo de datos con los valores de ancho de banda y capacidad de procesamiento de cada unidad.

Puesto que es un sistema serie, por lo que respecta a los datos referentes a las partículas que atraviesan el detector, aplicamos la ecuación 2.14 que repetimos aquí por comodidad.

$$Q_S = \frac{AB_1}{\sum_{i=1}^N t_i} \cdot \sum_{i=1}^N Q_i \cdot \frac{t_i}{AB_i} \quad (4.9)$$

De la expresión (4.9) lo único que nos falta por conocer son los tiempos de cada uno de los elementos de procesado.

El primer tiempo, t_1 , es el correspondiente al procesador RAID. Este elemento se limita a leer los digitalizadores y a enviar los datos hacia la tarjeta procesadora DBV44. El tiempo empleado en realizar estas operaciones es de aproximadamente 100 μs correspondientes a la escritura en memoria de todos los datos de los digitalizadores (60-40 palabras de 2 bytes a 100 MB/s) más la lectura de los mismos para transmitirlos hacia la DBV44 (60-40-2 bytes a 100 MB/s).

El procesado de los datos en la tarjeta DBV44 corresponde al tiempo t_2 y es aproximadamente igual a 165 μs , que se corresponden con la escritura en memoria de los datos, su procesado y la lectura de los resultados para ser enviados hacia la CPU PowerPC.

Por último, el tiempo t_3 corresponde al montaje de los datos según el formato ATLAS y a su grabación en disco en el procesador PowerPC; su valor aproximado es de 500 μs , procedentes de la escritura de los datos en memoria, montaje de la trama

Por último, el tiempo t_3 , corresponde al montaje de los datos según el formato ATLAS y a su grabación en disco en el procesador PowerPC; su valor aproximado es de 500 μ s, procedentes de la escritura de los datos en memoria, montaje de la trama según el formato ATLAS, su escritura en disco y las operaciones de monitoraje. Por lo tanto, el tiempo total de procesado es de 0'8 ms.

Sustituyendo en la ecuación (4.9), el valor del factor de calidad requerida para el sistema tras la inclusión del prototipo es $Q'_{ib} \approx 15.000 \text{ MIPS} \times \text{MB/s}$. Como se observa ha aumentado en relación al sistema inicial debido a la inclusión de más elementos de procesado. Sin embargo, este sistema presenta dos ventajas frente al inicial:

- El tiempo total de adquisición de los datos se reduce sensiblemente por lo que se adquieren más sucesos por burst.
- El preprocesado realizado en los datos simplifica la estimación de energía realizada, en este caso, también de forma *off-line*.

El aumento, predecible, del factor de calidad requerida ha de valorarse de forma adecuada, ya que indica que el sistema se encuentra preparado para una adquisición de mayor número de canales y más rápida, objetivos de estas pruebas en vista de la instalación final en el experimento ATLAS. Mediante este sistema se pueden tomar teóricamente unos 1200 sucesos por segundo, mientras que con el primer sistema el límite está en unos 100 sucesos por segundo. La realidad es que, debido a las limitaciones de los sistema de interconexión y a que la adquisición está limitada por la velocidad del sistema basado en el RAID, la tasa de datos tomada estaba cercana a 80 sucesos/s.

3.2.3 Resultados

El principal resultado de las pruebas fue la comprobación del correcto funcionamiento del sistema y de la corrección del diseño en vista a un prototipo final adecuado para el experimento ATLAS/LHC. El incremento en el factor de calidad requerido mediante la adición de nuevos elementos no es limitante si nos fijamos en los valores estimados para todo el experimento ATLAS/LHC. La implementación del módulo de preprocesado y agrupamiento de datos demuestra que es posible la realización de un sistema que mejore el rendimiento de la adquisición introduciendo preprocesado en la cadena.

Por otra parte, mediante el sistema descrito en el epígrafe anterior se realizó una serie de toma de datos en la que en total se almacenaron más de 2000 ficheros de datos para su posterior análisis. El objetivo de este análisis era, también, la comprobación del ajuste entre la estimación de la energía mediante la suma de las muestras digitalizadas y la interpolación parabólica realizada con cinco muestras únicamente. La figura 4.25 muestra un ejemplo de dicho ajuste.

4. BIBLIOGRAFÍA

- [1] TileCal Collaboration. *ATLAS Tile Calorimeter Technical Design Report*, CERN/LHCC 96-42, 1996.
- [2] Hamamatsu photonics. *Photomultiplier catalogue*, 1994.
- [3] Dell'Acqua et al. *A digital front-end and readout microsystem for calorimetry at LHC. The FERMI project*, IEEE Transactions on Nuclear Science, vol. 40, n° 4, 1993.
- [4] González, V. y Sanchis, E. *Application of a real-time data processing VME module using a custom high speed I/O multiport board to "ATLAS Tile-Cal Read Out Driver"*, ATLAS TILECAL-NO-056, 1995.
- [5] ATLAS Trigger-DAQ Steering Group. *Trigger & DAQ interfaces with front-end systems: requirement document (Draft 2.0)*, ATLAS/DAQ-NO-, 1995.
- [6] Taylor, B. *Timing, Trigger and Control signals distribution for LHC detectors*, CERN LHC/ Note , 1994.
- [7] Boyle, O., McLaren, R. y Van der Bij, E. *The S-Link Interface Specification*, CERN/ECP Division, 1996.
- [8] Motorola Corp. *OptoBus Technical Information*, BR1459/D.
- [9] McLaren, R. y Boyle, O. *ATLAS Read-Out link data format*, CERN/ECP, 1998.
- [10] Texas Instruments Inc. *TMS320C4x - User's Guide*.
- [11] Loughborough Sound Images Limited. *TIM-40 Module and Carrier Board*. VME Support Manual, version 1.00, 1994.
- [12] IEEE, *PCI Mezzanine Card*, IEEE 1386.
- [13] RD24 Collaboration. *RD24 Status Report, Application of the Scalable Coherent Interface to data acquisition at LHC*, CERN/LHCC 96-33, 1996.
- [14] Mora, F. *Aplicación de SCI en la adquisición de datos del detector ATLAS*. Tesis doctoral, Universidad Politécnica de Valencia, 1997.
- [15] Daura, F. *El ruido en los sistemas digitales: I a VIII*, Mundo Electrónico, n° 169 (págs.81-85), n° 171 (págs.126-131), n° 173 (págs.123-128), n° 175 (págs.95-102), n°135 (págs.135-144), n°180 (págs.97-105), n°181 (págs.53-61), 1986-1997
- [16] *Printed Circuit Board Layout for Improved Electromagnetic Compatibility*, Application Report, Texas Instrument, 1995.
- [17] Alvarez Santos, R. *Tecnología Microelectrónica 2. Circuitos impresos*, Ed. Ciencia.
- [18] Normas UNE 20-620-82, UNE 20-620-84, UNE 20-621-83, UNE 20-621-84, UNE 20-621-85, UNE 20-524-1 UNE 20-524-5.
- [19] *Manual de BetaSoft_R*, Dynamic Soft Analysis, Inc.
- [20] Holman, J.P. *Heat Transfer*, McGraw-Hill.
- [21] Microware User's Manual. *OS9 Advanced System Software*, 1991.
- [22] AMD, "Transparent Asynchronous transmitter-receiver Interface (TAXI)".
- [23] Hwang, K. *Advanced Computer Architecture*. McGraw-Hill, 1993.
- [24] IEEE. *VMEbus Specification 1014-1987*.
- [25] *NIM Standard, DOE/ER-0457*, 1964
- [26] Creative Electronic Systems, S. A. *VIC8251F User's Manual*.

- [27] CAEN, *FERMI Module V676 User's Manual*.
- [28] Creative Electronic Systems, S. A. *RCB 8047 CORBO User's Manual*.
- [29] Creative Electronic Systems, S. A. *RAID 8325 User's Manual*.
- [30] Lynx Real-Time Systems, *LynxOS User's Guide Version 2.5*, 1996.

CAPÍTULO

5

CONCLUSIONES Y TRABAJO FUTURO

1. CONCLUSIONES	1
2. TRABAJO FUTURO	2

1. CONCLUSIONES

Esta tesis comenzó tras la observación de que en muchos de los sistemas de procesamiento de sensores se venía empleando una técnica de integración jerárquica pero no se encontraba en la bibliografía consultada e incluso tras preguntar a las personas directamente involucradas en estos sistemas, una razón que justificara su empleo.

Ante este hecho, se planteó la tarea de intentar formalizar el problema en base a algún parámetro que permitiese la comparación de sistemas heterogéneos. La primera conclusión de esta tesis es que se ha definido este parámetro aquí llamado *factor de calidad requerida* de un sistema que depende del ancho de banda y la capacidad de procesamiento necesarias para resolver un determinado problema. Este factor se define de forma que a mayor valor más calidad requiere nuestro sistema y por lo tanto puede esperarse una mayor complejidad e incluso coste del mismo.

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 las expresiones para la obtención del factor de calidad requerida de la asociación serie y paralelo de procesadores.
- Se ha modelado el sistema de adquisición y procesamiento de datos para las redes de sensores distribuidos y, en general, se han obtenido las expresiones que permiten comparar dos arquitecturas en función de su factor de calidad requerida a partir de los resultados de los sistemas serie y paralelo. Mediante este factor se han comparado dos posibles soluciones a un problema de procesamiento y fusión de datos en redes de sensores distribuidos. Los resultados obtenidos apuntan a la solución jerárquica basada en sucesivos niveles de refinamiento como la mejor en aquellos casos en los que el problema venga caracterizado por un gran volumen de información a adquirir donde sólo una pequeña porción es de interés. Dicho estudio no se había encontrado en la bibliografía consultada.
- Se ha introducido una mejora en la arquitectura de procesamiento consistente en la aplicación de técnicas de localidad y agrupamiento de datos y se ha demostrado que reducen el factor de calidad requerida para el sistema.
- Se han aplicado los resultados del modelado anterior a un problema concreto donde el volumen de información es elevado y la proporción de datos válidos muy baja. Se trata de un experimento de física de altas energías. Para este caso, se ha evaluado el factor de calidad requerida si el problema se solucionase mediante un sistema paralelo o jerárquico. Tras esto se ha descrito cómo se podría modificar el sistema introduciendo las mejoras descritas anteriormente, se ha evaluado el factor de nuevo comprobándose que resulta inferior.
- Para la verificación experimental de estos resultados se ha construido un módulo de agrupamiento de datos para uno de los subdetectores que se construyen para el experimento de física de altas energías. Se han realizado las pruebas en los bancos de test disponibles actualmente, y se ha comprobado que la introducción de este módulo reduce el factor de calidad requerida, lo que valida el modelo desarrollado. Además, este módulo ha demostrado cumplir con la funcionalidad requerida para el subdetector para 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:

- Continuar en la dirección del modelado de los sistemas de adquisición introduciendo algunas variables ahora no considerados. Entre ellas están el efecto de la topología de interconexión del sistema, la evaluación de la fiabilidad de los sistemas y su alteración en función de la arquitectura empleada y el estudio de la complejidad de los sistemas y su relación con el factor de calidad de los mismos.
- Desarrollar un modelo para la simulación discreta de las diferentes opciones presentadas que permita un estudio más rápido de otros problemas a partir de sus requerimientos particulares.
- Profundizar en los modelos estadísticos de fusión óptima de datos para mejorar los sistemas de decisión y fusión empleados en ejemplos como el descrito en el trabajo.
- Estudiar y desarrollar un nuevo prototipo para el agrupamiento de datos, en vista de los resultados obtenidos con el actual, para su utilización en el sistema de adquisición de datos final del detector. Este trabajo ya ha comenzado y se ha pensado en una arquitectura paralela donde se incluirían cuatro procesadores dedicados en paralelo controlados por un procesador más general. La figura 5.1 muestra la solución prevista.

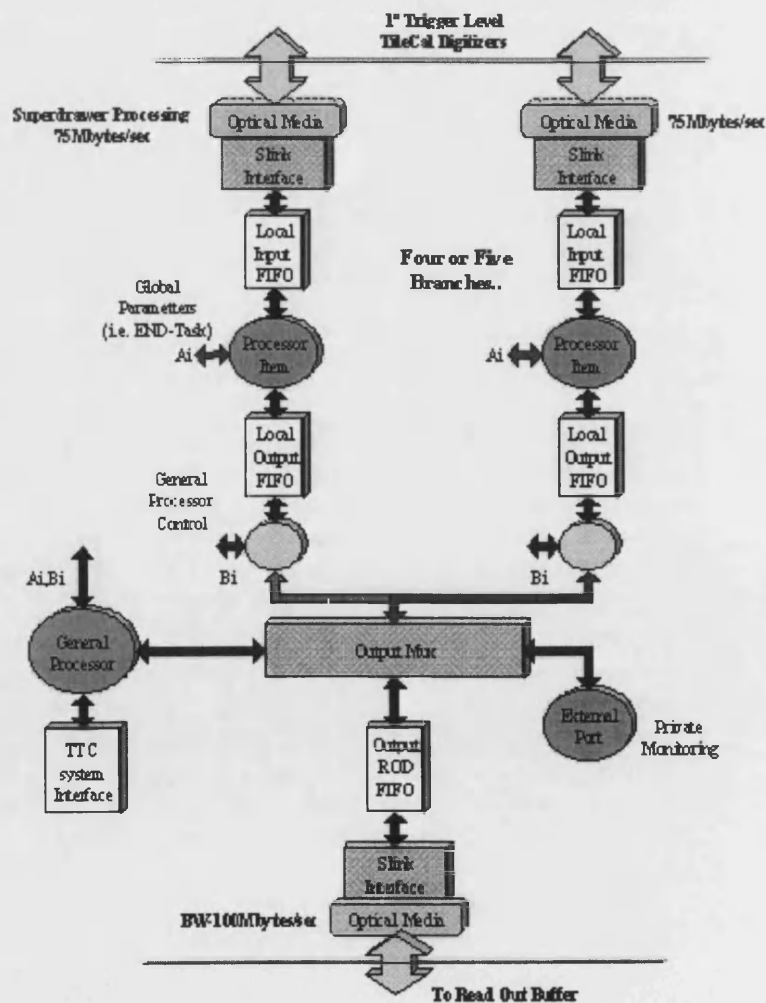


FIGURA 5.1 Diagrama de bloques del nuevo prototipo.

El procesado a realizar mediante esta tarjeta será el cálculo de la energía depositada en cada canal j del detector a partir de únicamente siete muestras de las 40 tomadas por el digitalizador. La evaluación de esta energía debe seguir la expresión de la ecuación (5.1):

$$E_j = C_j \times \sum_{i=1}^7 (S_j^i - P_j) \times H_j^i \quad (5.1)$$

donde C_j es una constante de calibración que corrige los errores de la no linealidad de los sensores y las atenuaciones en la transmisión, S_j^i es el valor de la muestra i , P_j es el valor del pedestal para ese canal, es decir, el valor del ruido presente y H_j^i es el coeficiente de filtrado óptimo para la muestra, obtenido mediante el estudio de la señal eléctrica proporcionada por los sensores y transductores.

Según esa expresión cada uno de los procesadores del prototipo debería realizar unos 450 productos y unas 800 sumas en los 10 μ s que dispone el sistema. Esto implica que la potencia de cálculo del sistema debe ser mayor o igual a 125 MFLOPs.

Las primeras pruebas realizadas se han dirigido a la evaluación de procesadores comerciales actualmente disponibles para poder conocer el estado actual de la tecnología antes que acudir a una solución hardware desarrollada específicamente para la aplicación. Estas pruebas se han encaminado hacia la medición de los tiempos de ejecución del algoritmo en los diferentes procesadores con los resultados de la tabla 5.1.

DSPs	Tiempo (μ s)
C40	32'5
C25	35
Procesador general	Tiempo (μ s)
PowerPC Motorola 100 MHz	176
AlphaStation 200 4/233	76
HP 9000/831	176

TABBLA 5.1 Tiempos de ejecución del algoritmo de estimación de energía para diferentes procesadores.

De estos primeros estudios parece evidente que los procesadores digitales de señal son los dispositivos adecuados en cuanto a potencia de cálculo puesto que con la tecnología de hoy en día ya se aproxima mucho al valor deseado.

Además de incluir DSPs, las soluciones a investigar deben pasar por la posibilidad de disponer de los mismos en tarjetas con formato PMC para aumentar la flexibilidad y poder hacer uso del bus PCI, único con el que, actualmente, parece viable la solución propuesta sin acudir a diseños específicos.

ANEXO

I

**ESQUEMAS DE LA TARJETA
INTERFACE S-LINK**

1. ESQUEMAS DE LA TARJETA INTERFACE S-LINK

A continuación se presentan los esquemas de disposición de componentes, rutado por la cara de pistas y por la de soldaduras del interface S-LINK diseñado para el prototipo de agrupamiento de datos.

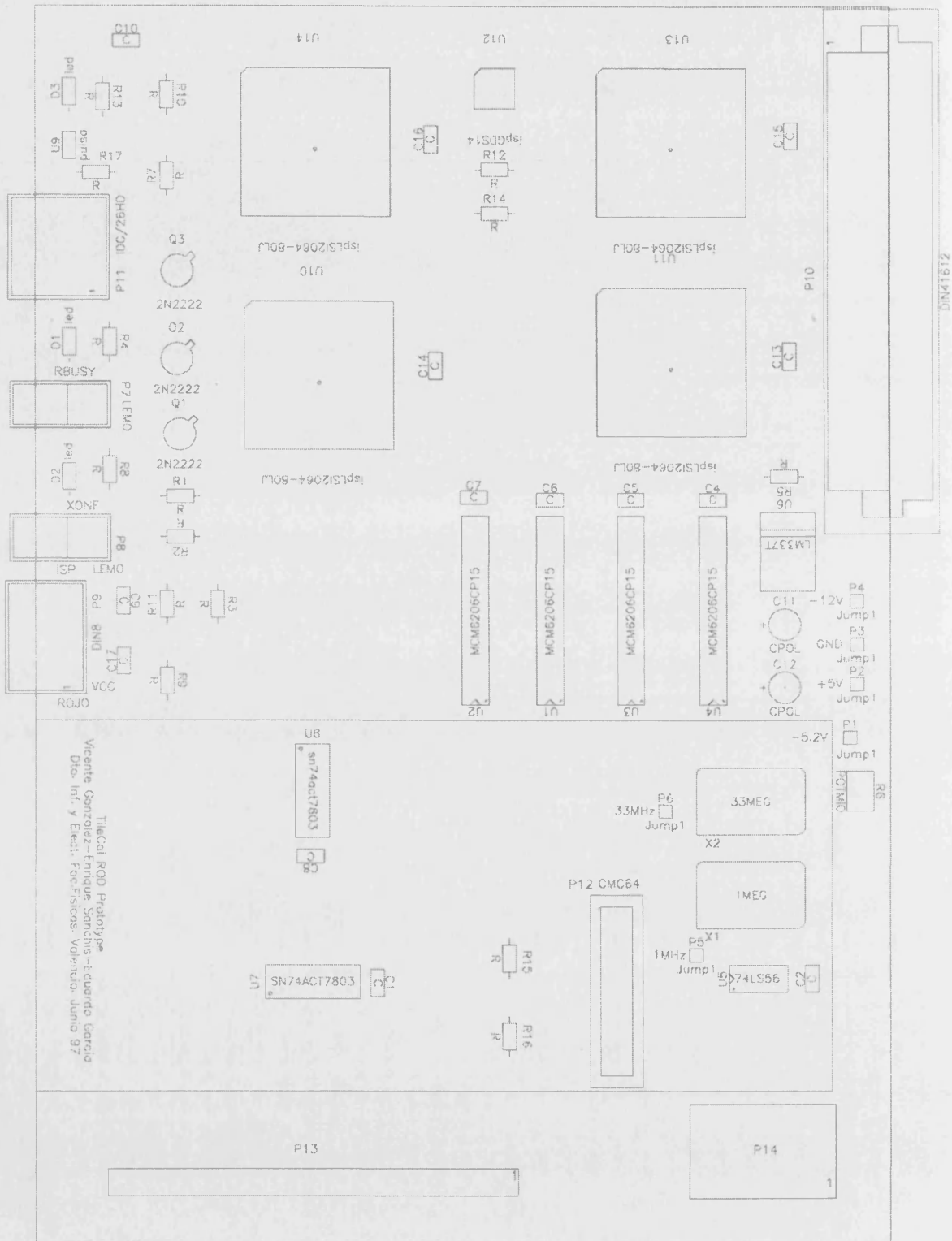
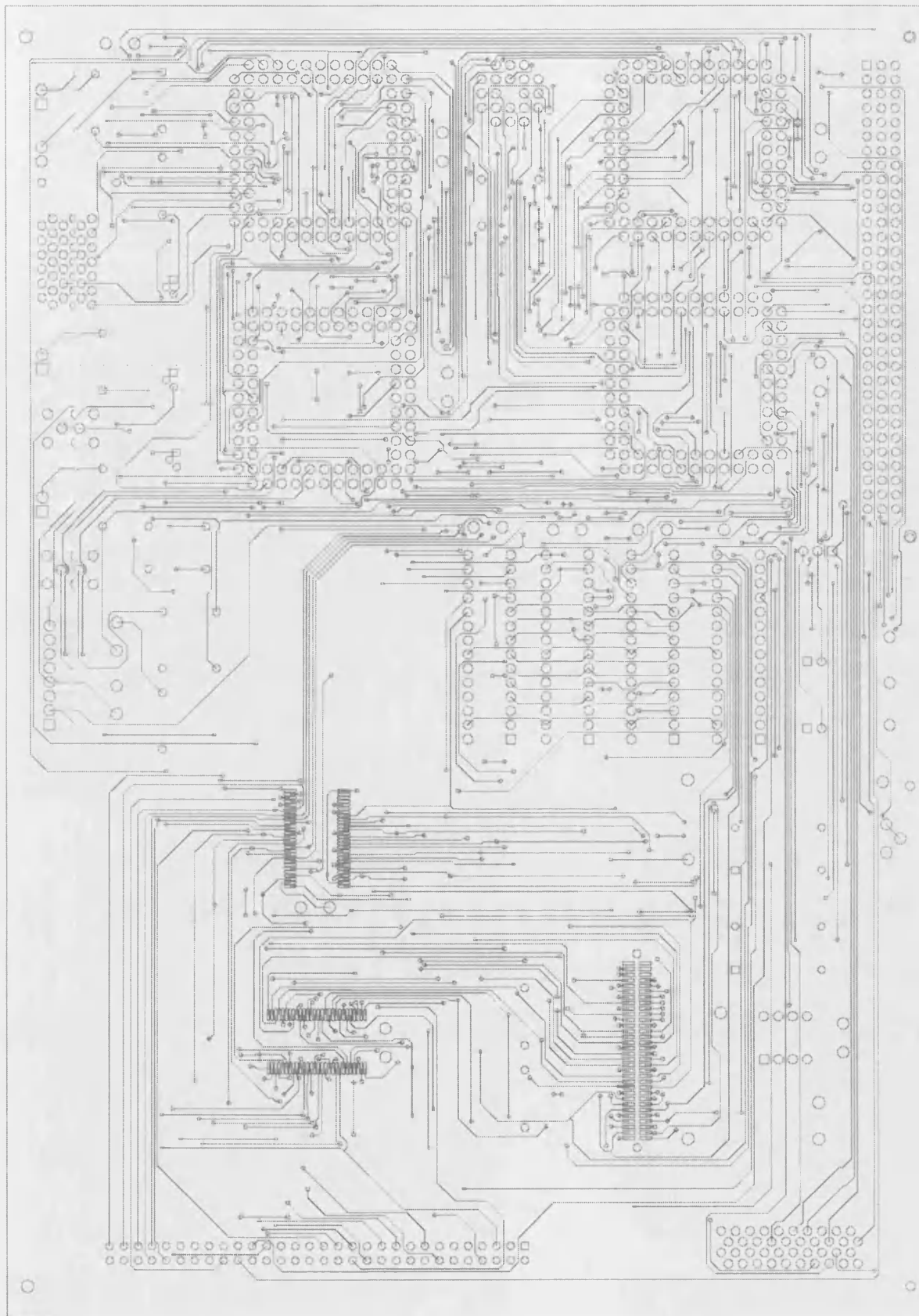


FIGURA I.1 Disposición de los componentes de la tarjeta interface S-LINK.



FIGGURA I.2 Aspecto del rutado de pistas por la cara de componentes de la tarjeta interface S-LINK.

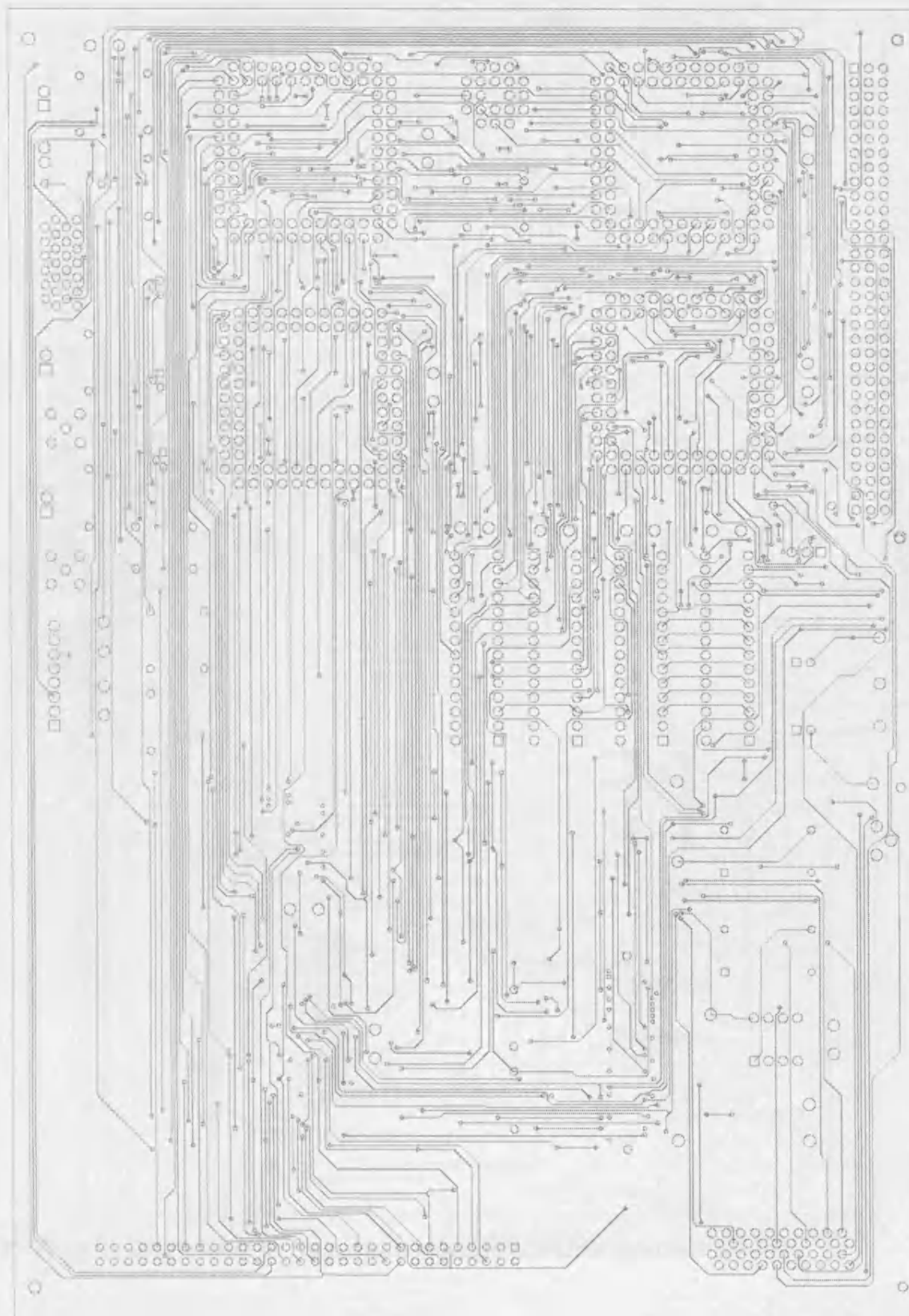


FIGURA I.3 Aspecto del rutado de pistas por la cara de soldaduras de la tarjeta interface S-LINK.