



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

PROYECTO FIN DE CARRERA

**Diseño de una plataforma de comunicaciones
bajo la norma IEC-61850**

AUTOR: Eduardo Alonso Rivas

MADRID, Septiembre 2009

ESTE PROYECTO CONTIENE LOS SIGUIENTES DOCUMENTOS

DOCUMENTO N° 1, MEMORIA

Parte I. Memoria	Páginas 1 a 99.	99 páginas.
Parte II. Estudio Económico	Páginas 100 a 102.	3 páginas.
Parte III. Código fuente	Páginas 103 a 104.	2 páginas.
Parte IV. Datasheets	Páginas 105 a 124.	20 páginas.

DOCUMENTO N° 2, PLIEGO DE CONDICIONES

Condiciones generales y económicas	Páginas 1 a 2.	2 páginas.
Condiciones técnicas y particulares	Páginas 3 a 4.	2 páginas.

DOCUMENTO N° 3, PRESUPUESTO

Mediciones	Páginas 1 a 2.	2 páginas.
Precios unitarios	Página 3.	1 página.
Sumas Parciales	Página 4.	1 página.
Presupuesto General	Página 5.	1 página.

Autorizada la entrega del proyecto

**Diseño de una plataforma de comunicaciones bajo la norma
IEC-61850**

Realizado por:

Eduardo Alonso Rivas

DIRECTORES DEL PROYECTO

Sadot Alexandres Fernández

Fdo.

Fecha:/...../.....

Carlos Rodríguez-Morcillo García

Fdo.

Fecha:/...../.....

Vº Bº DEL COORDINADOR DE PROYECTOS

Álvaro Sánchez Miralles

Fdo.

Fecha:/...../.....



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

PROYECTO FIN DE CARRERA

**Diseño de una plataforma de comunicaciones
bajo la norma IEC-61850**

AUTOR: Eduardo Alonso Rivas

MADRID, Septiembre 2009

DISEÑO DE UNA PLATAFORMA DE COMUNICACIÓN BAJO LA NORMA IEC-61850

Autor: Alonso Rivas, Eduardo.

Directores: Alexandres Fernández, Sadot.

Rodríguez-Morcillo García, Carlos.

Entidad colaboradora: IIT-Instituto de Investigación Tecnológica.

RESUMEN DEL PROYECTO

El objetivo principal de la norma IEC 61850 es conseguir la interoperabilidad entre los distintos equipos electrónicos inteligentes (IED, *Intelligent Electronic Device*) que componen un sistema de automatización de una subestación eléctrica (SAS, *Substation Automation System*), pudiendo ser dichos equipos de distintos fabricantes.

Concretamente, IEC 61850 desarrolla un modelo de datos que recoge toda la información que puede ser necesaria en un sistema de automatización de una instalación eléctrica, así como sus modos de intercambio entre los diferentes elementos de una manera coherente y estandarizada.

De entre la gran variedad de servicios de comunicación que ofrece esta norma, se encuentran algunos relacionados con mensajes especiales, tales como el aviso de la ocurrencia de un evento automático o el envío de valores instantáneos de medidas analógicas.

El desarrollo de este proyecto gira en torno a este último tipo de servicio, para cuya descripción la norma IEC-61850 reserva los apartados 9-1 y 9-2.

Surge de esta forma la motivación concreta de este trabajo, que consiste en el desarrollo de un dispositivo que sea capaz de capturar datos analógicos y transmitirlos dentro de un sistema de información acorde a la norma estudiada.

Las características deseables para la plataforma que se quiere obtener se describen a continuación:

- Debe tratarse de un elemento ligero y compacto, con el fin de permitir su empotramiento en otro dispositivo.
- El modelo de datos en el que se recojan los valores muestreados para su envío debe atenerse a lo especificado en la IEC-61850. Esto implica, de acuerdo con las especificaciones de la misma, que la transmisión de la información debe soportar el protocolo Ethernet.
- La configuración tanto de las propiedades de la transmisión de datos como de la información que se desea enviar debe poder ser fácilmente adaptada a cada caso en particular.

Todos estos requisitos dirigen las posibles implementaciones hacia la tecnología FPGA, la cual permite el encapsulado de distintos elementos hardware tales como memorias o controladores en una única pastilla.

De esta forma, la solución desarrollada consistió en un diseño lógico volcado sobre una tarjeta que contaba con un puerto Ethernet para la comunicación de datos con el exterior.

Dicho diseño requirió un profundo estudio de la norma IEC-61850 en sus apartados 9-1 y 9-2 y del protocolo de comunicación Ethernet (IEEE 802.3) a fin de estructurar los datos muestreados según un formato reconocible por cualquier dispositivo receptor que se atenga a los estándares mencionados.

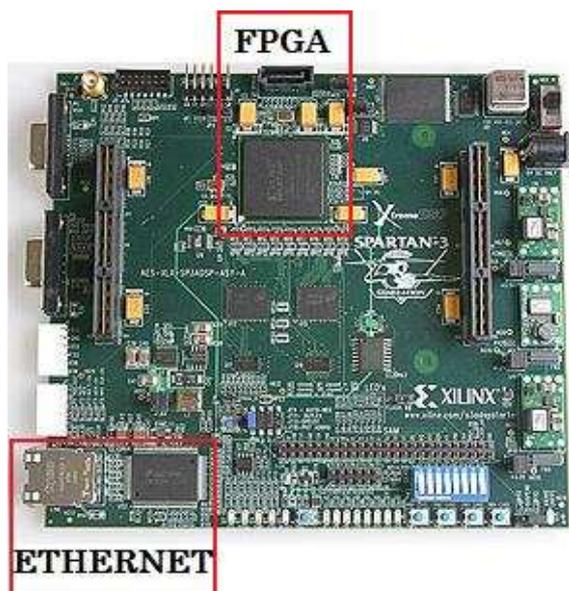


Figura 1. Placa de desarrollo empleada en el proyecto.

Resultados

La verificación tanto del protocolo de comunicación como de la correcta implementación para conseguir un intercambio satisfactorio de la información se realizó con la ayuda de un PC con un analizador de protocolos que captura y descompone las tramas Ethernet completas.

De esta forma, se obtuvieron los resultados esperados, monitorizando tramas IEC-61850 enviadas periódicamente con la estructura de datos recogida en los apartados correspondientes de la norma.

6	18.970757	Intel_62:c6:08	Dell_8d:14:e4	0x88ba	IEC 61850/SV
7	18.980762	Intel_62:c6:08	Dell_8d:14:e4	0x88ba	IEC 61850/SV
Ethernet II, Src: Intel_62:c6:08 (00:aa:00:62:c6:08), Dst: Dell_8d:14:e4 (00:12:3f:8d:14:e4)					
Destination: Dell_8d:14:e4 (00:12:3f:8d:14:e4)					
Source: Intel_62:c6:08 (00:aa:00:62:c6:08)					
Type: IEC 61850/SV (Sampled Value Transmission (0x88ba))					
Data (649 bytes)					
0000	00 12 3f 8d 14 e4 00 aa	00 62 c6 08 88 ba 40 00	..?.....	.b....@.	
0010	02 89 00 00 00 60 82	02 7d 80 01 05 a2 82 02}}	
0020	76 30 82 00 7a 80 35 11	22 33 44 55 66 77 88 99	v0..z.5.	"3DUfw..	
0030	aa 11 22 33 44 55 66 77	88 99 aa 11 22 33 44 55	.. "3DUfw "3DU	
0040	66 77 88 99 aa 66 77 88	99 aa 82 02 11 d5 83 04	fw...fw.	
0050	00 00 00 00 85 01 00 86	02 01 90 87 82 00 40 da@.@.	
0060	da 99 cd ef 11 22 33 44	55 66 77 88 99 01 23 45 "3D	Ufw... #E	
0070	67 89 cd ef 11 22 33 44	55 66 77 88 99 01 23 45	g.... "3D	Ufw... #E	
0080	67 89 cd ef 11 22 33 44	55 66 77 88 99 01 23 45	g.... "3D	Ufw... #E	
0090	67 89 cd ef 11 22 33 44	55 66 77 88 99 da da 30	g.... "3D	Ufw... 0	
00a0	82 00 7a 80 35 11 22 33	44 55 66 77 88 99 aa 11	..z.5."3	DUfw... .	
00b0	22 33 44 55 66 77 88 99	aa 11 22 33 44 55 66 77	"3DUfw..	.. "3DUfw	
00c0	88 99 aa 66 77 88 99 aa	82 02 11 d6 83 04 00 00	...fw...	
00d0	00 00 85 01 00 86 02 01	90 87 82 00 40 da da 99@.@.	
00e0	cd ef 11 22 33 44 55 66	77 88 99 01 23 45 67 89	... "3DUf	w... #Eg.	
00f0	cd ef 11 22 33 44 55 66	77 88 99 01 23 45 67 89	... "3DUf	w... #Eg.	
0100	cd ef 11 22 33 44 55 66	77 88 99 01 23 45 67 89	... "3DUf	w... #Eg.	
0110	cd ef 11 22 33 44 55 66	77 88 99 da da 30 82 00	... "3DUf	w... 0..	

Figura 2. Tramas IEC-61850 SV obtenidas como resultado.

Conclusiones

Como resultado del trabajo realizado, se ha obtenido de manera exitosa una plataforma que precisa únicamente de una FPGA y un controlador para el envío de datos vía Ethernet.

Estas características hardware, unidas al encapsulado de los datos analógicos muestreados según las especificaciones de la norma IEC-61850, cumplen perfectamente con los objetivos de ligereza, flexibilidad en la configuración y estandarización que se perseguían con el desarrollo de este proyecto.

Por ello se espera que sirva de base para realizar aplicaciones reales que faciliten la integración y la gestión eficiente de todos los niveles de los sistemas de generación, distribución y consumo de energía eléctrica.

DEVELOPMENT OF A COMMUNICATION DEVICE UNDER IEC-61850 STANDARD

Author: Alonso Rivas, Eduardo.

Supervisors: Alexandres Fernández, Sadot.

Rodríguez-Morcillo García, Carlos.

Partner: IIT-Institute for Research in Technology.

PROJECT SUMMARY

The main purpose of IEC-61850 standard is the achievement of a successful interaction among the different IEDs (Intelligent Electronic Devices) involved in a SAS (Substation Automation System) without any impact due to the specific features from each manufacturer.

Specifically, IEC 61850 propouses a specific data model that contains all the useful information needed in an automation system set in an electric substation. In addition, this standard suggests a consistent method to exchange those data from each device.

There is a lot of different communication services described in IEC-61850. One of them is related to special messages, such as the transmission of sampled analogue values, which is described in two sections of the standard: 9-1 and 9-2.

This type of service represents the starting point of the development of this project.

At this point, the goal of this project is the development of an electronic device capable of fetching analogue values and transmit them into an information system according to the standard mentioned above.

The expected features for the device are described as follows:

- It must consist of a light and compact appliance, in order to be able to embed it in any other device.
- The sampled analogue values and its data model must be agreed with IEC-61850 specifications. This involves the use of Ethernet protocol for the transmission of information.

- The Ethernet transmission features and the data settings must be configurable in order to be able to adapt the communication easily to a specific scenario.

All this requirements are be able to implement into FPGA technology, which permits the insertion of different hardware elements such as memories or drivers within a single chip.

In this way, one solution consists in a logical design embedded in an electronic platform with an Ethernet port for the data communication.

This development requires a deep study of IEC-61850 standard on its sections 9-1 and 9-2 and the Ethernet protocol, described in IEEE 802.3 specifications, so as the sampled data could be stored in a standardized structure.

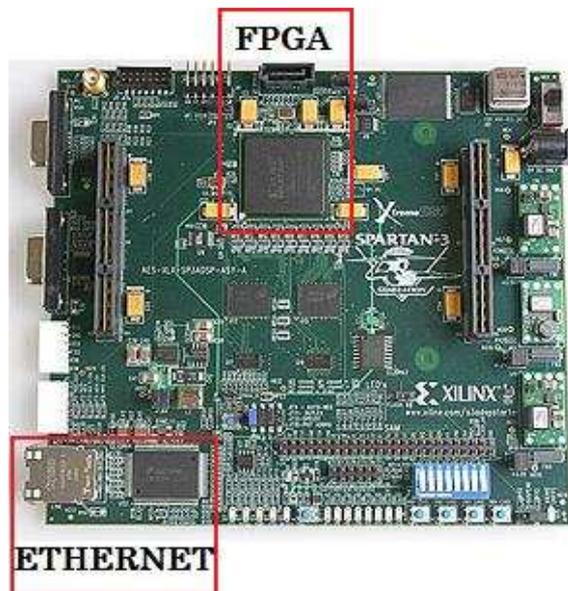


Figure 1. Development board used in the project.

Results

The test of the communication protocol as well as the physical implementation to get a successful trading of information required the use of a PC with a protocol analyzer tool installed, which captures and separates the whole Ethernet frames.

In this way, the expected results were obtained and IEC-61850 frames were recorded periodically with the structure suggested in the different sections of the standard.

6	18.970757	Intel_62:c6:08	Dell_8d:14:e4	0x88ba	IEC 61850/SV	
7	18.980762	Intel_62:c6:08	Dell_8d:14:e4	0x88ba	IEC 61850/SV	
Ethernet II, Src: Intel_62:c6:08 (00:aa:00:62:c6:08), Dst: Dell_8d:14:e4 (00:12:3f:8d:14:e4)						
Destination: Dell_8d:14:e4 (00:12:3f:8d:14:e4)						
Source: Intel_62:c6:08 (00:aa:00:62:c6:08)						
Type: IEC 61850/SV (Sampled Value Transmission (0x88ba))						
Data (649 bytes)						
0000	00 12 3f 8d 14 e4 00 aa 00 62 c6 08 88 ba 40 00					..?..... .b....@.
0010	02 89 00 00 00 00 60 82 02 7d 80 01 05 a2 82 02				 }.....
0020	76 30 82 00 7a 80 35 11 22 33 44 55 66 77 88 99					v0..z.5. "3Dufw..
0030	aa 11 22 33 44 55 66 77 88 99 aa 11 22 33 44 55					.. "3Dufw "3DU
0040	66 77 88 99 aa 66 77 88 99 aa 82 02 11 d5 83 04					fw...fw.
0050	00 00 00 00 85 01 00 86 02 01 90 87 82 00 40 da				 @...
0060	da 99 cd ef 11 22 33 44 55 66 77 88 99 01 23 45				 "3D Ufw... #E
0070	67 89 cd ef 11 22 33 44 55 66 77 88 99 01 23 45					g.... "3D Ufw... #E
0080	67 89 cd ef 11 22 33 44 55 66 77 88 99 01 23 45					g.... "3D Ufw... #E
0090	67 89 cd ef 11 22 33 44 55 66 77 88 99 da da 30					g.... "3D Ufw... 0
00a0	82 00 7a 80 35 11 22 33 44 55 66 77 88 99 aa 11					..z.5. "3 Dufw...
00b0	22 33 44 55 66 77 88 99 aa 11 22 33 44 55 66 77					"3Dufw.. .. "3Dufw
00c0	88 99 aa 66 77 88 99 aa 82 02 11 d6 83 04 00 00					...fw... ..
00d0	00 00 85 01 00 86 02 01 90 87 82 00 40 da da 99				 @...
00e0	cd ef 11 22 33 44 55 66 77 88 99 01 23 45 67 89				 "3DUF w... #Eg.
00f0	cd ef 11 22 33 44 55 66 77 88 99 01 23 45 67 89				 "3DUF w... #Eg.
0100	cd ef 11 22 33 44 55 66 77 88 99 01 23 45 67 89				 "3DUF w... #Eg.
0110	cd ef 11 22 33 44 55 66 77 88 99 da da 30 82 00				 "3DUF w... 0..

Figure 2. IEC-61850 SV frames obtained as results.

Conclusions

As a result of this project, a communication platform with a FPGA chip and an Ethernet driver supported has been obtained in a successfully way.

These hardware features, joint to the fact that the sampled analogue values are structured according to IEC-61850 standard specifications, fulfill the initial requirements of this project.

Because of this, it is desirable that this development will be useful for future real applications aimed to the integration of an efficient management concerning all the levels of a generation, distribution and consumption of electric power.

Documento 1

Memoria

ÍNDICE GENERAL

Parte I. Memoria	1
Capítulo 1. Introducción	2
1. Trabajos y tecnologías existentes.....	2
1.1 Estudios sobre la norma IEC-61850.....	2
1.2 Proyecto IELAS	3
1.3 ¿Por qué emplear tecnología FPGA?	4
2. Motivación del proyecto	7
3. Objetivos	9
4. Metodología / Solución desarrollada	9
5. Recursos y herramientas empleadas	10
5.1 Xilinx ISE Design Suite 10.1	10
5.2 Wireshark Network Protocol Analyzer Version 1.0.7	11
5.3 Protocolo Ethernet.....	11
5.4 Spartan 3A-DSP 1800A	14
5.5 Plataforma de conexión.....	16
Capítulo 2. Norma IEC-61850	17
1. Introducción.....	17
2. Conceptos básicos.....	17
2.1 Subestación eléctrica.....	18
2.2 Niveles del SAS e Interfaces Lógicas	18
2.3 Funciones y Nodos Lógicos (LN)	21
3. Modelo de información del SAS.....	22
3.1 Clases genéricas de datos (CDC, <i>Common Data Classes</i>)	23
3.2 Dispositivo Lógico (LD, <i>Logical Device</i>).....	23
4. Modelo de Servicios Abstractos de Comunicación	24
4.1 ACSI, <i>Abstract Communication Service Interface</i>	24
4.2 GSE, <i>Generic Substation Event</i>	26
4.3 TSV, <i>Transmission of Sampled Values</i>	27
5. Normas que componen la IEC-61850	28
6. Apartado 9-1.....	30
6.1 Capa física.....	31
6.2 Capa de enlace.....	32
6.3 Mapping del modelo de transmisión de valores muestreados usando multicast	34
6.4 Trama física y requisitos de envío.....	38
7. Apartado 9-2.....	41

7.1 <i>Stack</i> de comunicación	41
7.2 Capa física.....	43
7.3 Capa de enlace.....	43
7.4 Mapping del modelo de transmisión	45
7.5 Sintaxis ASN.1 y ejemplo de aplicación.....	50
8. Conclusiones.....	57
Capítulo 3. IEEE 802.3: Ethernet.....	58
1. Introducción.....	58
2. Capa física	58
3. Trama MAC.....	60
4. Trama Tagged MAC.....	64
5. Encapsulado de trama SAV según protocolo Ethernet.....	65
Capítulo 4. Arquitectura desarrollada.....	67
1. Esquema general de funcionamiento.....	67
2. Módulo <i>ConfigROM</i>	69
3. Módulo <i>DataROM</i>	72
4. Módulo <i>ConfigRAM</i>	73
5. Módulo <i>MVtoRAM</i>	74
6. Módulo <i>RAM</i>	76
7. Módulo <i>ethernetSend</i>	77
7.1 Composición de la trama.....	77
7.2 Cálculo del CRC.....	77
7.3 Transmisión física	80
7.4 Máquina de estados.....	81
8. Módulo <i>Ethernet_top</i>	82
8.1 Puertos	82
8.2 Máquina de estados.....	82
Capítulo 5. Experimentos y Resultados.....	84
1. Simulación del módulo.....	84
1.1 <i>Test bench</i>	84
1.2 Resultados de la simulación	86
2. Pruebas de transmisión	92
2.1 <i>ChipScope Pro Analyzer</i>	92
2.2 Resultado experimental	93
Capítulo 6. Conclusiones	95
Capítulo 7. Futuros desarrollos	96
1. Enlace de configuración del dispositivo.....	96

2. Etapa de captura de datos.....	97
3. Empleo de fibra óptica como medio de transmisión.....	97
Bibliografía.....	98
<i>Parte II. Estudio económico.....</i>	<i>100</i>
<i>Parte III. Código fuente.....</i>	<i>103</i>
Descripción VHDL.....	104
<i>Parte IV. Datasheets.....</i>	<i>105</i>
Documentación.....	106
Spartan 3A-DSP 1800-A.....	107
NS DP83865 Gig PHYTER.....	117

ÍNDICE DE FIGURAS

FIGURA 1-1. TARJETA DE COMUNICACIÓN DEL PROYECTO IELAS.....	3
FIGURA 1-2. CIRCUITO LÓGICO DE UN SUMADOR.	4
FIGURA 1-3. ARQUITECTURA DE LA TARJETA DEL PROYECTO IELAS.	6
FIGURA 1-4. ESQUEMA DEL PROYECTO DENISE.	7
FIGURA 1-5. WIRESHARK PROTOCOL ANALYZER	11
FIGURA 1-6. ESTRUCTURA GENERAL DEL PROYECTO.	12
FIGURA 1-7. ESTRUCTURA DE LOS TEST-BENCH.....	12
FIGURA 1-8. ESTRUCTURA DE LA PARTE DE ENVÍO DEL STACK IP.....	13
FIGURA 1-9. IMAGEN DE LA WEB DEL GENERADOR DE CRC.....	14
FIGURA 1-10. PLACA DE DESARROLLO EMPLEADA EN EL PROYECTO.....	15
FIGURA 1-11. ESTRUCTURA DEL ENLACE ETHERNET.....	15
FIGURA 1-12. DISPOSITIVO DE CONEXIÓN AL PC.	16
FIGURA 2-1. EJEMPLO DE DIVISIÓN DE UNA SUBESTACIÓN EN BLOQUES ELÉCTRICOS. (FUENTE:[4]).....	18
FIGURA 2-2. EJEMPLO DE TOPOLOGÍA DE LA AUTOMATIZACIÓN DE UNA SAS. (FUENTE:[5])	19
FIGURA 2-3. ESQUEMA UNIFILAR DE SUBESTACIÓN ELÉCTRICA DE DISTRIBUCIÓN CON DIVERSAS ALTERNATIVAS PARA LA DISPOSICIÓN DEL BUS DE PROCESO. (FUENTE:[3]).....	20
FIGURA 2-4. DISTINTAS POSIBILIDADES DE POSICIONAMIENTO DE LNS EN IEDS. (FUENTE:[5]).....	22
FIGURA 2-5. VIRTUALIZACIÓN DE UNA SUBESTACIÓN. (FUENTE:[5])	22
FIGURA 2-6. ESTRUCTURA DEL MODELO DE DATOS. (FUENTE:[1]).....	24
FIGURA 2-7. ESTRUCTURA DEL MODELO DE DATOS DE UN LD. (FUENTE:[5])	28
FIGURA 2-8. TRANSMISIÓN DE MENSAJES TSV DESDE UNA UNIDAD CONCENTRADORA A VARIOS IEDS. (FUENTE: [2]).....	30
FIGURA 2-9. ESTRUCTURA DE LA CABECERA PRIORIDAD Y VLAN. (FUENTE: [2]).....	32
FIGURA 2-10. ESTRUCTURA DEL ETHERTYPE PDU. (FUENTE: [2]).....	33
FIGURA 2-11. ESTRUCTURA Y CAMPOS DEL UNIVERSAL DATA-SET. (FUENTE: [2])	34
FIGURA 2-12. CONFIGURACIÓN DE ADQUISICIÓN DE DATOS DE ECT/EVT. (FUENTE: [2])	35
FIGURA 2-13. ESTRUCTURA Y CAMPOS DEL STATUS INDICATION DATA-SET. (FUENTE: [2])	36
FIGURA 2-14. CONCATENACIÓN DE ASDUS DENTRO DE UN APDU. (FUENTE: [2]).....	38

FIGURA 2-15. STACK DE COMUNICACIÓN REFERENCIA. (FUENTE:[3])	42
FIGURA 2-16. ESTRUCTURA DEL ETHERTYPE PDU. (FUENTE:[3]).....	44
FIGURA 2-17. FORMATO BER PARA EL ENCAPSULADO DE LA TRAMA. (FUENTE:[3]).....	51
FIGURA 2-18. FORMATO DEL BYTE TAG. (FUENTE:[3]).....	51
<i>FIGURA 2-19. FORMATO DE CODIFICACIÓN DEL CAMPO LENGTH.</i>	52
FIGURA 2-20. EJEMPLO DE APDU. (FUENTE:[3])	53
FIGURA 2-21. ESTRUCTURA COMPLETA DEL APDU. (FUENTE:[7])	54
FIGURA 2-22. CODIFICACIÓN DEL DATA-SET EMPLEADO. (FUENTE:[7])	55
FIGURA 2-23. MODELO PARA LA TRANSMISIÓN DE SAMPLED VALUES (FUENTE:[6])	57
FIGURA 3-1. FORMATO DE LA TRAMA MAC. (FUENTE:[8])	60
FIGURA 3-2. ESQUEMA DE UNA TRAMA CON CAMPO EXTENSION. (FUENTE:[8]).....	63
FIGURA 3-3. FORMATO DE LA TRAMA TAGGED-MAC. (FUENTE:[8])	64
FIGURA 3-4. TRAMA SAV ENCAPSULADA BAJO EL ESTÁNDAR 802.3 (FUENTE:[7])	66
FIGURA 4-1. ESTRUCTURA DE LA SOLUCIÓN IMPLEMENTADA.	67
FIGURA 4-2. CAPTURA DE PANTALLA DEL CORE GENERATOR.	70
FIGURA 4-3. ESQUEMA DEL APDU ENVIADO.	75
FIGURA 4-4. CÁLCULO INCORRECTO DEL CRC ANTE LOS DATOS ASCII 123456789.	79
FIGURA 4-5. CÁLCULO CORRECTO DEL CRC ANTE LOS DATOS ASCII 123456789.....	79
FIGURA 4-6. DIAGRAMA DE ESTADOS DEL MÓDULO ETHERNETSEND.	81
FIGURA 4-7. DIAGRAMA DE ESTADOS DEL MÓDULO ETHERNET_TOP	83
FIGURA 5-1. ESTRUCTURA DE UN PROCESO DE TEST.....	85
FIGURA 5-2. RESPUESTA DEL DISEÑO ANTE SIMULACIÓN. CAPTURA Y ENVÍO DE DATOS ..	87
FIGURA 5-3. RESPUESTA DEL DISEÑO ANTE SIMULACIÓN. ESCRITURA DE INFORMACIÓN DE CONFIGURACIÓN EN LA MEMORIA RAM.....	88
FIGURA 5-4. RESPUESTA DEL DISEÑO ANTE SIMULACIÓN. ESCRITURA DE DATOS MUESTREADOS EN LA MEMORIA RAM (I).....	89
FIGURA 5-5. RESPUESTA DEL DISEÑO ANTE SIMULACIÓN. ESCRITURA DE DATOS MUESTREADOS EN LA MEMORIA RAM (II)	90
FIGURA 5-6. RESPUESTA DEL DISEÑO ANTE SIMULACIÓN. ENVÍO DEL CRC	91
FIGURA 5-7. CAPTURA DE LA TRAMA ENVIADA UNA VEZ LEÍDA POR EL PC.....	94

ÍNDICE DE TABLAS

TABLA 2-1. DEFINICIÓN DE CDC MEASURED VALUES. (FUENTE:[3]).....	23
TABLA 2-2. CLASES DE SERVICIOS ACSI DEFINIDOS. (FUENTE:[6])	25
TABLA 2-3. DEFINICIÓN DE CLASE DATA-SET Y SERVICIOS ASOCIADOS. (FUENTE:[6])	26
TABLA 2-4. DEFINICIÓN DE CLASE MSVCB Y SERVICIOS ASOCIADOS. (FUENTE:[6])	27
TABLA 2-5. STACK DE COMUNICACIÓN. (FUENTE: [2])	31
TABLA 2-6. DEFINICIONES REFERENTES AL UNIVERSAL DATASET. (FUENTE: [2])	35
TABLA 2-7. CODIFICACIÓN DE LA CLASE DE DATOS SPS. (FUENTE: [2]).....	37
TABLA 2-8. CODIFICACIÓN DE LA CLASE DE DATOS MV. (FUENTE: [2])	37
TABLA 2-9. GUÍA DE SELECCIÓN PARA LA CAPA FÍSICA ETHERNET. (FUENTE: [2])	40
TABLA 2-10. A-PROFILE PARA LOS SERVICIOS SV. (FUENTE:[3])	42
TABLA 2-11. T-PROFILE PARA LOS SERVICIOS SV. (FUENTE:[3])	43
TABLA 2-12. RANGO DE DIRECCIONES MULTICAST RECOMENDADAS. (FUENTE:[3])	43
TABLA 2-13. VALORES DE LOS CAMPOS VID Y USER PRIORITY DE LA ETIQUETA DE PRIORITY DE LA TRAMA. (FUENTE: [3]).....	44
TABLA 2-14. VALORES DE LA CABECERA ETHERTYPE PDU. (FUENTE:[3])	44
TABLA 2-15. DEFINICIÓN MMS DE LA ESTRUCTURA DEL MSVCB. (FUENTE: [3])	46
TABLA 2-16. MAPPING DE LOS SERVICIOS MULTICAST. (FUENTE:[3])	46
TABLA 2-17. DEFINICIÓN MMS DE LA ESTRUCTURA DEL USVCB. (FUENTE: [3])	47
TABLA 2-18. MAPPING DE LOS SERVICIOS UNICAST. (FUENTE:[3])	47
TABLA 2-19. CODIFICACIÓN DEL BUFFER DE TRANSMISIÓN DE SV. (FUENTE:[3])	48
TABLA 2-20. CODIFICACIÓN DE LOS TIPOS BÁSICOS DE DATOS. (FUENTE:[3])	49
TABLA 2-21. DESGLOSE DEL CAMPO LENGTH DE LOS ATRIBUTOS.....	56
TABLA 2-22. DESGLOSE DEL CAMPO TAG DE LOS ATRIBUTOS.	57
TABLA 3-1. ALTERNATIVAS PARA EL MEDIO DE TRANSMISIÓN	59
TABLA 3-2. DESGLOSE DE LA INFORMACIÓN DE LA TRAMA.....	65

Parte I. MEMORIA

Capítulo 1. INTRODUCCIÓN

A lo largo de este Capítulo introductorio se puede encontrar un primer acercamiento a las especificaciones concretas del proyecto.

El primer apartado recoge un resumen de los estudios y desarrollos que han resultado de utilidad en este trabajo, cuya motivación y objetivos se detallan en los apartados 2 y 3. A continuación se especifican las tareas necesarias para la consecución de dichos objetivos en el apartado 4. Por último, se presenta la relación de recursos empleados para la realización del diseño, indicando la fuente de la que se obtuvo cada uno de ellos.

1. Trabajos y tecnologías existentes

1.1 Estudios sobre la norma IEC-61850

Como se describirá de manera más detallada en apartados posteriores, la norma IEC-61850 recoge las directrices para el desarrollo de un modelo de comunicación estándar sobre un sistema de energía eléctrica.

Existen numerosos documentos con análisis de diferentes materias al respecto, pudiendo encontrarse algunos en la Bibliografía de esta memoria.

Rafael Santodomingo Berry, en su trabajo *Modelos de información en sistemas eléctricos de potencia* (documento [1]) analiza el estándar IEC-61850 y su estructura de información, comparándola con la del modelo CIM a fin de obtener una armonización entre ambas.

Este escrito ha sido empleado como guía para la elaboración del Capítulo 2, donde se realiza una introducción a la norma.

La referencia [7] de la Bibliografía aporta una visión fundamental para el desarrollo del proyecto, ya que incluye un ejemplo detallado de

aplicación del modelo de encapsulamiento y envío de los datos acorde con el apartado 9-2 del estándar.

En este caso, el documento ha resultado de gran utilidad, debido a que las tramas enviadas por la plataforma de comunicación diseñada se han estructurado en base al ejemplo expuesto en el mismo.

1.2 Proyecto IELAS

Este proyecto, comprendido dentro del VI programa marco y desarrollado por Carlos Mateo Domingo y Juan Antonio Talavera Martín, supone un salto desde los estudios teóricos mencionados anteriormente hacia la aplicación práctica.

El proyecto consistió en la elaboración de una tarjeta capaz de adquirir muestras de un transformador para enviarlos a una protección dentro de un sistema automático de una subestación eléctrica. Esta transmisión debía seguir el modelo de datos y la descripción XML de los dispositivos cuyas pautas impone la IEC-61850. La Figura 1-1 muestra la plataforma diseñada a tal efecto.

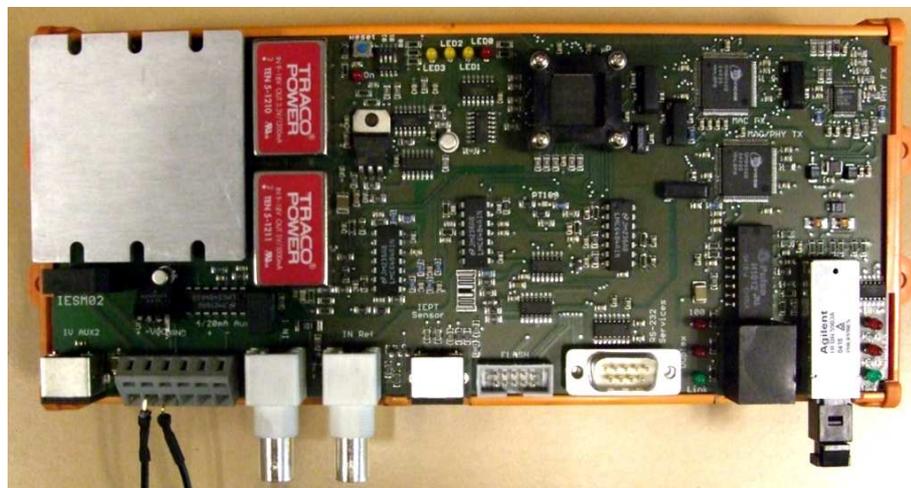


Figura 1-1. Tarjeta de comunicación del proyecto IELAS.

Como puede observarse en la Figura anterior, el desarrollo de la tarjeta requirió el empleo de un microcontrolador para la gestión de los datos, varias memorias físicas y diversas entradas analógicas y digitales, así como puerto Ethernet y de fibra óptica para el envío de la información.

Este trabajo puede considerarse un punto de partida para el proyecto que se presenta, ya que en ambos casos lo que se busca es realizar una plataforma de comunicación cuyo modelo de envío de datos se ajuste a las especificaciones impuestas por la IEC-61850.

Como diferencia fundamental, es necesario señalar que el proyecto IELAS centró la aplicación de dicha norma al caso de un transformador y su protección asociada, tal y como se describe en su apartado 9-1.

Sin embargo, en este caso el objetivo es conseguir una plataforma altamente configurable que permita el envío de valores muestreados desde cualquier tipo de dispositivo, lo que conlleva el estudio del apartado 9-2 de la norma y el desarrollo de un diseño más flexible.

1.3 ¿Por qué emplear tecnología FPGA?

Un dispositivo FPGA (*Field Programmable Gate Array*) es un elemento que contiene bloques lógicos básicos cuya interconexión y funcionalidad se pueden programar.

De esta forma, es posible conseguir por ejemplo un sumador como el que se muestra en la Figura 1-2 simplemente describiendo cada uno de los elementos que aparecen en el esquema y las conexiones entre ellos.

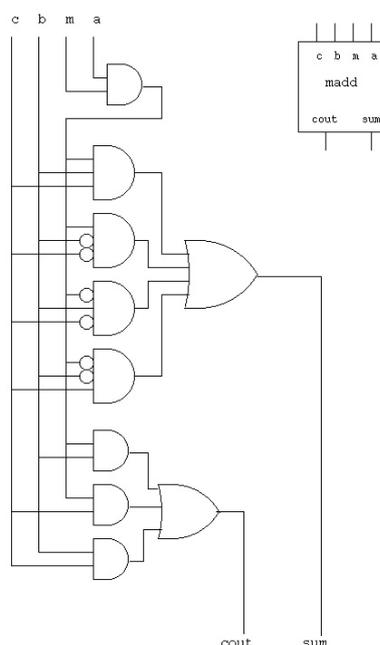


Figura 1-2. Circuito lógico de un sumador.

La representación textual del circuito se hace según un lenguaje del tipo HDL (*Hardware Description Language*), que permite la descripción de circuitos lógicos complejos, tal y como puede verse en el siguiente código:

```
entity madd is      -- multiplying full adder stage
  port(c      : in  std_logic;
        b      : in  std_logic;
        m      : in  std_logic;
        a      : in  std_logic;
        sum     : out std_logic;
        cout    : out std_logic);
end entity madd;
architecture circuits of madd is
  signal aa: std_logic;
begin
  aa <= a and m;
  sum <= (aa and b and c) or (aa and not b and not c) or
        (not aa and b and not c) or (not aa and not b and c) after 1 ns;
  cout <= (aa and b) or (aa and c) or (b and c) after 1 ns;
end architecture circuits;
```

Este tipo de tecnología presenta grandes ventajas a la hora de realizar un diseño como el que se pretende desarrollar:

- En primer lugar, se trata de una tecnología altamente configurable, ya que permite el embebimiento de cualquier tipo de elemento capaz de ser descrito en lógica digital, como puede ser el caso de una memoria RAM o incluso un microcontrolador que gestione todos los recursos de la tarjeta donde se implante.
- Es un recurso ligero y con gran capacidad de modularidad, lo que posibilita aislar partes del diseño para su análisis y depuración.
- La descripción lógica es de menor nivel que la programación software convencional, lo que permite un mayor control sobre los enlaces físicos y las estructuras de datos empleadas, así como la temporización y la monitorización de señales digitales críticas para, por ejemplo, establecer la comunicación entre dispositivos.

Veamos entonces qué puede aportar esta tecnología al diseño de una tarjeta similar a la desarrollada en el proyecto IELAS, cuya arquitectura se esquematiza en la siguiente Figura.

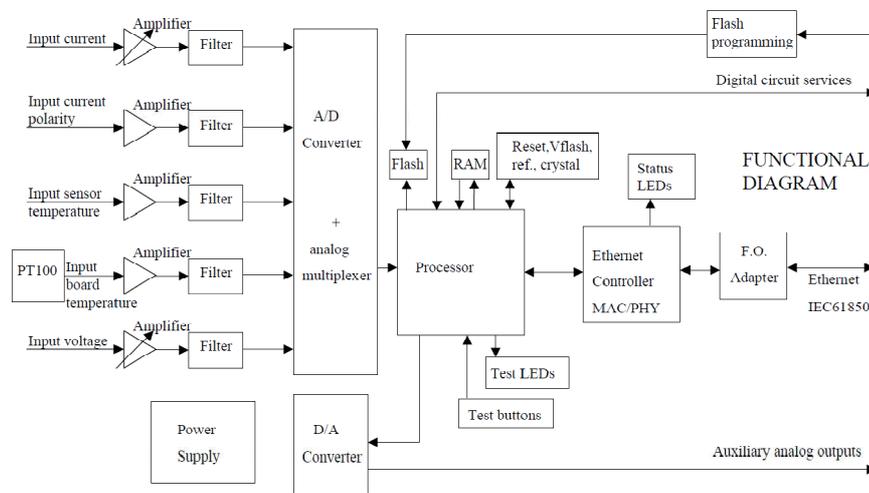


Figura 1-3. Arquitectura de la tarjeta del proyecto IELAS.

Analizando el recorrido que sigue la información en el diagrama, se puede observar cómo la adquisición de muestras procedentes de un transformador o sensor mediante un conversor A/D produce datos digitales que llegan a un procesador. Éste interactúa con una memoria Flash configurable desde el exterior y con una memoria RAM que almacena información del dispositivo.

Los datos, una vez tratados, se envían al controlador MAC/PHY, el cual los encapsula para su lanzamiento por el puerto Ethernet.

El empleo de una FPGA para el diseño de una plataforma de comunicación similar a la descrita permitirá embutir, dentro de una única pastilla, los bloques correspondientes al control del protocolo Ethernet, la etapa de procesamiento de los datos y ambas memorias; pudiendo además organizar todos los módulos dentro de un esquema jerárquico de control.

De esta manera, se obtendrá un dispositivo ligero y compacto al que sólo restarían añadir ciertas interfaces con el entorno, como puede ser la correspondiente a la adquisición de datos analógicos (conversor) o los puertos necesarios para el envío de las tramas (Ethernet) o la configuración y acceso a memoria (Ethernet o RS-232).

2. Motivación del proyecto

Este trabajo se desarrolla dentro del marco del proyecto DENISE, ajustándose a las condiciones establecidas en el consorcio correspondiente.

El proyecto DENISE (Distribución Energética Inteligente-Segura-Eficiente) consiste en dotar a la red de distribución eléctrica convencional de una estructura de integración inteligente tanto en la generación como en distribución y consumo.

En la siguiente Figura se esquematizan los diferentes objetivos de este proyecto.



Figura 1-4. Esquema del Proyecto DENISE.

Una de las parcelas de trabajo es conseguir la interoperabilidad entre los distintos equipos electrónicos inteligentes (IED, *Intelligent Electronic Device*) que componen un sistema de automatización de una subestación eléctrica (SAS, *Substation Automation System*), pudiendo ser dichos equipos de distintos fabricantes.

Con este propósito surge la norma IEC 61850, la cual presenta un modelo de datos que recoge toda la información que puede ser necesaria en un sistema de automatización de una instalación eléctrica y su intercambio entre varios dispositivos.

De esta forma, la norma define la organización de dispositivos, su información y estructura de datos, sin restringir la funcionalidad de los elementos. El Capítulo 2 recoge un análisis en profundidad de este modelo de comunicación.

Uno de los servicios que ofrece la estructura de datos de la IEC-61850 es el envío de mensajes desde un dispositivo hacia otro de nivel superior. Concretamente, este proyecto se centra el envío de mensajes de transmisión de valores analógicos muestreados (*SendMSVMessage*, *SendUSVMessage*).

¿En qué tipo de aplicaciones resulta útil este desarrollo?

De manera inmediata se puede pensar en un dispositivo que soporte este servicio dentro de una subestación eléctrica. Por ejemplo, un transformador de tensión puede monitorizar el estado de una línea hacia un equipo con una jerarquía superior, de forma que ante cualquier alteración se dispare una protección controlada remotamente.

Como se puede observar en la Figura 1-4, el proyecto DENISE persigue una integración horizontal que abarque desde la generación hasta el consumo de la energía eléctrica.

Pensemos ahora por un momento en el otro extremo de una línea eléctrica. Un dispositivo empotrable que permita el envío de valores analógicos encapsulados según un modelo estándar puede resultar útil para, por ejemplo, controlar el consumo de los electrodomésticos de una vivienda en tiempo real. Así, cada uno de ellos, con una configuración determinada, podría enviar por un puerto información sobre la potencia empleada para llevar a cabo su función a una unidad concentradora presente en cada hogar.

Surge aquí la motivación fundamental de este proyecto, que consiste en la implementación de la capa física del modelo de comunicación propuesto por la norma IEC-61850, de manera que se obtenga un dispositivo configurable para su uso en diferentes aplicaciones.

Como se expuso anteriormente, el proyecto IELAS desarrollado por Carlos Mateo y Juan Antonio Talavera plantea una arquitectura similar (Figura 1-3) que se empleará de base. El objetivo por tanto es sustituir la plataforma existente por una implantación hardware soportada en tecnología FPGA.

3. Objetivos

En el punto anterior se ha ofrecido un punto de partida para conocer cuál es la intención fundamental del proyecto.

Como se vio, el objetivo es desarrollar una plataforma hardware de comunicaciones configurable bajo la norma IEC-61850 capaz de transmitir datos que se ajusten al servicio de envío de valores analógicos muestreados.

La transmisión de datos, siguiendo el modelo de la norma, debe realizarse empleando el protocolo Ethernet como medio físico.

Por lo tanto, este proyecto contempla el empleo de dos estándares:

- IEC-61850 para el modelo de datos y encapsulado de los valores enviados.
- IEEE 802.3 para la transmisión física de las tramas.

La adaptación de la información según estos dos modelos debe llevar en paralelo una implementación abierta para la configuración tanto del dispositivo como de los datos enviados.

Un último requisito para la plataforma que se pretende diseñar es que permita su empotramiento dentro de gran variedad de dispositivos, para lo cual se requiere una solución ligera y compacta.

4. Metodología / Solución desarrollada

La plataforma configurable que se quiere realizar debe estar embebida dentro de un dispositivo hardware que permita tanto la captura como la transmisión de datos a gran velocidad.

La tecnología FPGA encaja perfectamente en estas características, ya que ofrece la posibilidad de realizar un diseño lógico altamente

configurable, en el cual se pueden incluir máquinas de estado, memorias físicas e incluso microcontroladores.

El uso de esta tecnología requiere el empleo de un lenguaje de descripción de circuitos lógicos HDL. En concreto el proyecto será programado en VHDL debido a su estudio previo durante la carrera.

Se presenta ya una idea clara y concisa de qué es lo que se pretende obtener con este proyecto y cómo se quiere conseguir. Para completar su desarrollo se llevarán a cabo las siguientes tareas.

- Estudio de la norma IEC 61850 y en concreto de las especificaciones en referencia a los servicios de comunicación y mapping a un sistema de comunicación concreto sobre Ethernet, que se describe en los apartados 9-1 y 9-2 (documentos [2] y [3]).
- Desarrollo hardware basado en un sistema empotrable con tecnología FPGA siguiendo el protocolo IEEE 802.3.
- Adaptación a los modelos de datos y servicios especificados por la norma IEC-61850.
- Comprobación del formato de encapsulado de los datos y del protocolo de comunicación. Para ello se utilizará una tarjeta Ethernet convencional de un PC para la captura de las tramas mediante un software a tal efecto.

5. Recursos y herramientas empleadas

El desarrollo de este proyecto requirió el uso tanto de medios físicos como de programas informáticos, los cuales se exponen a continuación.

5.1 Xilinx ISE Design Suite 10.1

La implantación del dispositivo de comunicación deseado dentro de una FPGA requiere el empleo de un software de diseño digital basado en lenguaje del tipo HDL.

Dentro de las herramientas del mercado, se empleó el software de *Xilinx ISE Design Suite*, versión 10.1, ya que el DEA de la Universidad contaba con una licencia industrial completa.

Este paquete software incluye tanto el programa de diseño (*Project Navigator*) como las herramientas de simulación (*ISim*), conexión a la placa (*iMPACT*) y depuración sobre el chip (*Chip Scope Pro Analyzer*).

5.2 Wireshark Network Protocol Analyzer Version 1.0.7

Este producto es una herramienta gratuita que permite la captura de tramas desde las tarjetas de adquisición del PC, así como su análisis en función del protocolo con el que se han codificado.

En este caso se empleó para monitorizar los datos recibidos por el módulo Ethernet del ordenador y su descomposición siguiendo el formato de encapsulado IEEE 802.3.

La Figura 1-5 muestra una captura del programa.

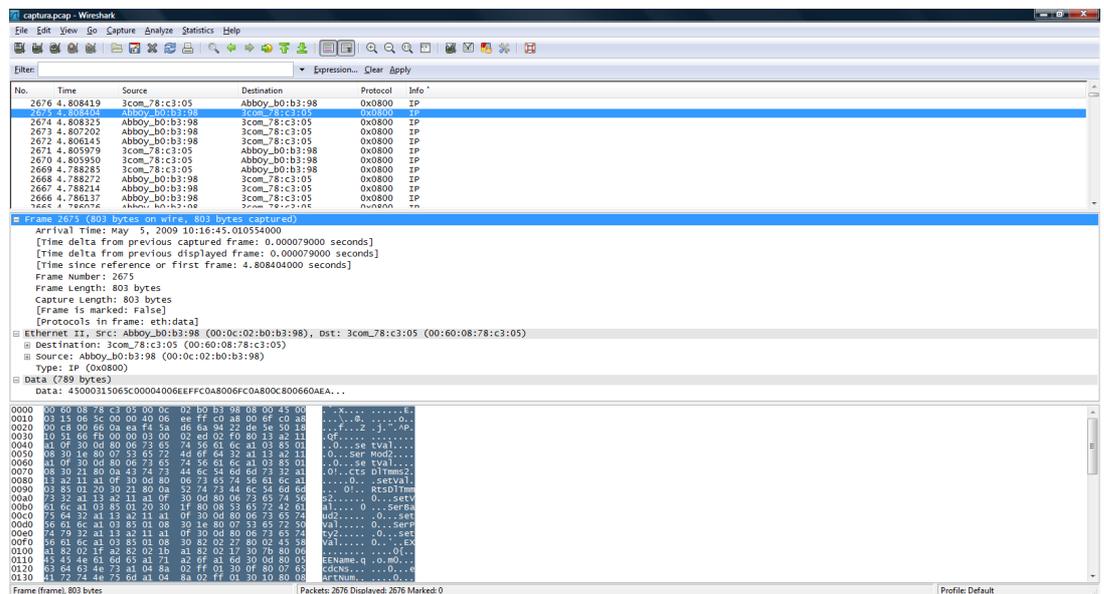


Figura 1-5. Wireshark Protocol Analyzer

5.3 Protocolo Ethernet

En el Capítulo 3 se puede encontrar un análisis sobre la trama Ethernet requerida para la transmisión. La generalización de este estándar supone una ventaja a la hora de plantear el proyecto, dado que es posible obtener un módulo implementado de varias fuentes.

Por una parte, el software de diseño empleado proporciona la posibilidad de trabajar con un bloque Ethernet IP (*Intellectual Property*) configurable. Sin embargo, la disponibilidad del mismo era

limitada ya que sólo permitía realizar simulaciones de un modo gratuito, mientras que la sintetización requería un pago previo.

La otra opción posible era recurrir a Internet.

De entre las webs que ofrecían módulos sintetizables de transmisión Ethernet, hubo tres que resultaron, al menos en principio, útiles para el proyecto:

5.3.1 Módulo Verilog

A través de la web <http://www.opencores.org/project.ethmac> se puede descargar un proyecto sintetizable sobre Ethernet MAC 10/100Mbps, desarrollado por Igor Mohor en 2002.

Las siguientes figuras muestran tanto la arquitectura completa del proyecto como la estructura de los test realizados sobre el mismo.

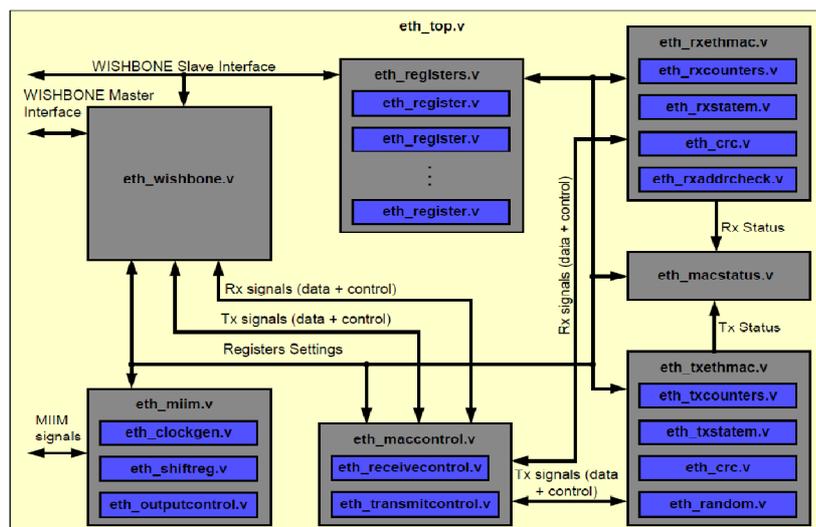


Figura 1-6. Estructura general del proyecto.

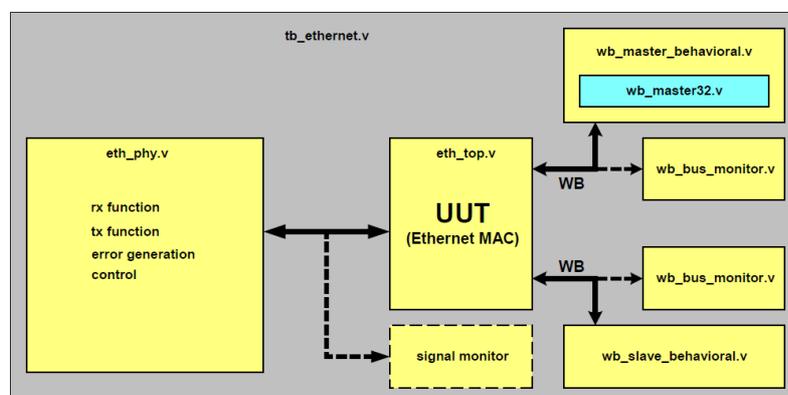


Figura 1-7. Estructura de los test-bench.

Este proyecto, que en principio parecía adaptarse a los requerimientos de la plataforma buscada, se descartó debido a dos razones.

En primer lugar, se trataba de un módulo muy voluminoso con un código muy extenso realizado además en un lenguaje diferente al esperado, ya que estaba descrito en Verilog. Este lenguaje, aunque también sintetizable, resultaba más oscuro debido al desconocimiento del mismo, lo que dificultaba enormemente la adaptación a las especificaciones deseadas.

La otra razón del descarte de este proyecto fue su aparente incompatibilidad con la herramienta software de simulación, ya que no se consiguió completar con éxito una prueba de la arquitectura.

Parecía por tanto necesario recurrir a otras fuentes para lograr una implementación que resultara útil para el proyecto.

5.3.2 Módulo VHDL

En <http://www.itee.uq.edu.au/~peters/xsvboard/stack/stack.htm> se puede encontrar un módulo con protocolo IP completo implementado en código VHDL desarrollado por la *School of Computer Science and Electrical Engineering University of Queensland* (Brisbane, Australia).

La Figura 1-8 recoge la organización de la parte emisora del proyecto.

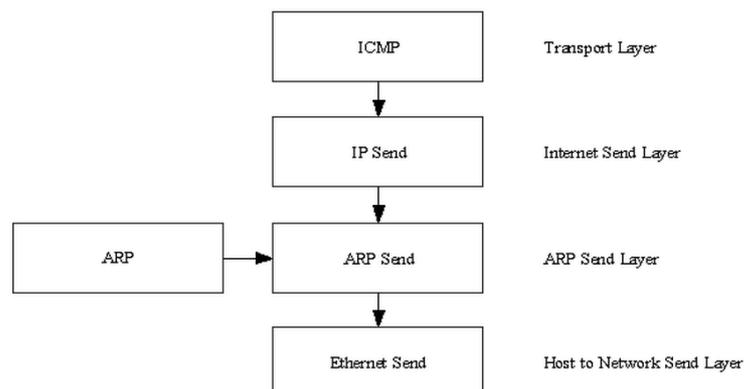


Figura 1-8. Estructura de la parte de envío del stack IP

Para la implementación requerida no era necesario añadir protocolo IP, ya que la conexión no se realiza vía Internet, sino mediante un concentrador físico. Por esta razón, únicamente resulta útil el módulo *Ethernet Send*.

Este módulo incluía una función para el cálculo del CRC (ver Capítulo 3, apartado 3). Sin embargo, por motivos de temporización, resultó necesaria la búsqueda de un diseño puramente combinacional.

5.3.3 CRC

Para el cálculo de este algoritmo se pueden encontrar multitud de opciones en la web. En este proyecto se ha optado por el generador disponible en el enlace <http://www.easics.com/webtools/crctool>.

La Figura 1-9 muestra la captura de pantalla de la página mencionada.

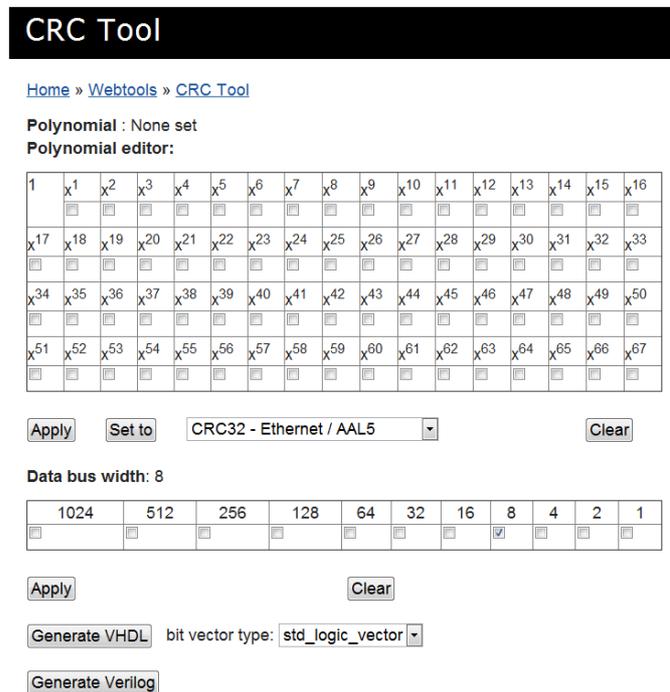


Figura 1-9. Imagen de la web del generador de CRC.

5.4 Spartan 3A-DSP 1800A

La plataforma de comunicaciones, una vez descrita, debe ser sintetizada sobre una tarjeta adaptada a tal efecto.

Debido a las especificaciones del proyecto, resultaba necesario contar con un dispositivo que ofreciera una FPGA y un medio de conexión Ethernet hacia el exterior.

En la línea de la opción elegida para el software se adquirió una tarjeta de desarrollo fabricada por Xilinx, que se muestra a continuación.

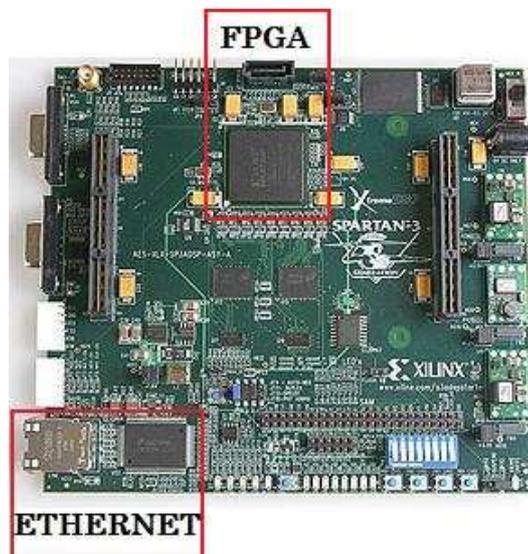


Figura 1-10. Placa de desarrollo empleada en el proyecto.

En la Figura 1-10 se han señalado los elementos de interés para la realización del proyecto. En la Parte IV de esta memoria pueden encontrarse las hojas de características de estos dispositivos.

5.4.1 FPGA

La tarjeta de desarrollo contiene una FPGA Xilinx *3SD1800A-FG676* de gran tamaño.

5.4.2 Puerto Ethernet

En cuanto a la conexión Ethernet, la placa presenta un conector *Tyco-AMP RJ-45* así como un controlador PHY de National Semiconductor con la referencia *National DP83865DVH Gig PHYTER* V, que permite una conexión de 10, 100 y 1000 Mbps.

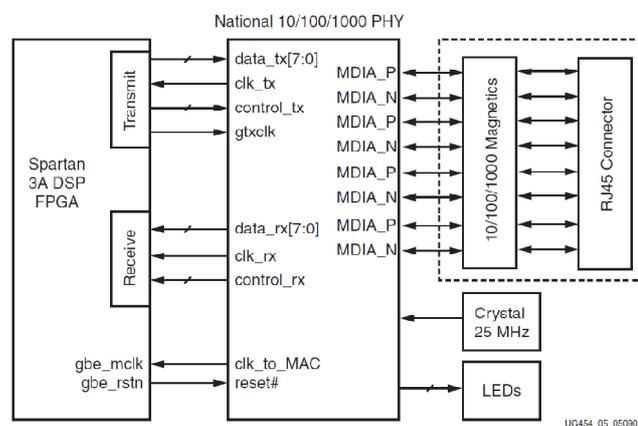


Figura 1-11. Estructura del enlace Ethernet.

5.5 Plataforma de conexión

El uso de la tarjeta presentada requiere de un dispositivo de conexión USB a un PC para su configuración.

Para cumplir esta función se adquirió un cable con la referencia *Xilinx Platform Cable USB II*, cuya apariencia se muestra a continuación.



Figura 1-12. Dispositivo de conexión al PC.

Capítulo 2. NORMA IEC-61850

El estudio de esta norma supone uno de los pilares fundamentales del proyecto, por lo que se reserva un Capítulo introductorio a la misma antes de exponer la solución desarrollada.

Los dos primeros apartados de este Capítulo recogen una serie de definiciones que se emplearán a lo largo de toda la exposición. Los apartados 3, 4 y 5 presentan los conceptos básicos relacionados con el modelo de información de la norma. Los tres últimos apartados resultan imprescindibles para el proyecto, ya que detallan las especificaciones concretas respecto al servicio de comunicación que se desea implantar.

1. Introducción

La norma IEC-61850 surge con el objetivo de garantizar la interoperabilidad entre distintos equipos electrónicos inteligentes (IED, Intelligent Electronic Device) que componen un sistema de automatización de una subestación eléctrica.

Para lograr este objetivo, la norma desarrolla un modelo de datos que recoge toda la información que puede ser necesaria en un sistema de automatización de una instalación eléctrica, de modo que todos los IEDs que cumplen con la norma organicen su información según el mismo modelo de datos.

La interoperabilidad, sin embargo, no garantiza la intercambiabilidad, lo que viene a significar que las funcionalidades para las que está preparado cada dispositivo no están estandarizadas.

2. Conceptos básicos

Tomando [1] como referencia, se presentarán diversos conceptos básicos necesarios para una comprensión global de la norma, tanto de elementos como de modelo de datos.

2.1 Subestación eléctrica.

Los sistemas que se tratan en la IEC 61850 son sistemas de automatización de las subestaciones eléctricas (SAS). Por ello, los diferentes elementos que se deben modelar para su control y supervisión son los componentes típicos que se pueden encontrar dentro de una subestación, como pueden ser seccionadores, interruptores, transformadores de intensidad y tensión y transformador de potencia.

Una vez modelados estos elementos, es necesaria su separación en distintos bloques eléctricos (bahías o *bays*), teniendo en cuenta que por motivos de funcionalidad o fabricación, los dispositivos pueden ser controlados de manera conjunta. La Figura 2-1 muestra una de las posibilidades de división de una subestación eléctrica: por módulos prefabricados.

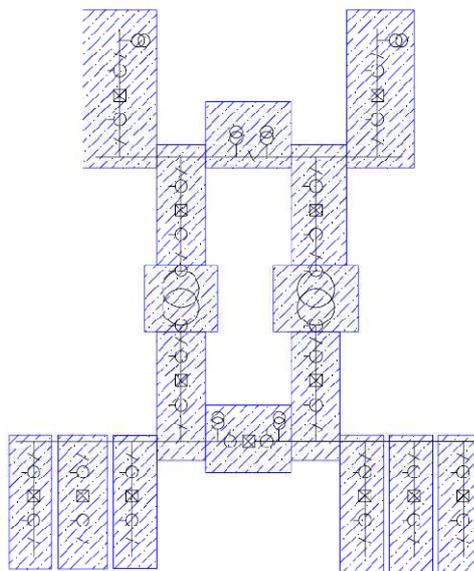


Figura 2-1. Ejemplo de división de una subestación en bloques eléctricos. (Fuente:[4])

Por supuesto, la norma no impone cómo se debe realizar la separación de la subestación en distintos bloques eléctricos, sino que en cada caso se debe analizar cuál es la mejor manera de dividirla.

2.2 Niveles del SAS e Interfaces Lógicas

Como se ha indicado en el Capítulo 1 de esta memoria, una de las deficiencias actuales en los sistemas de automatización de una

subestación eléctrica es que no permiten la interoperabilidad entre dispositivos de distintos fabricantes. A este inconveniente hay que unir el cableado excesivo que presentan los SAS.

La IEC-61850 permitirá a los sistemas de automatización superar ambas deficiencias, partiendo del modelo abstracto de información que se repite de manera genérica en todas las subestaciones eléctricas. La norma separa este sistema en tres niveles jerárquicos, representados en la Figura 2-2 que se muestra a continuación:

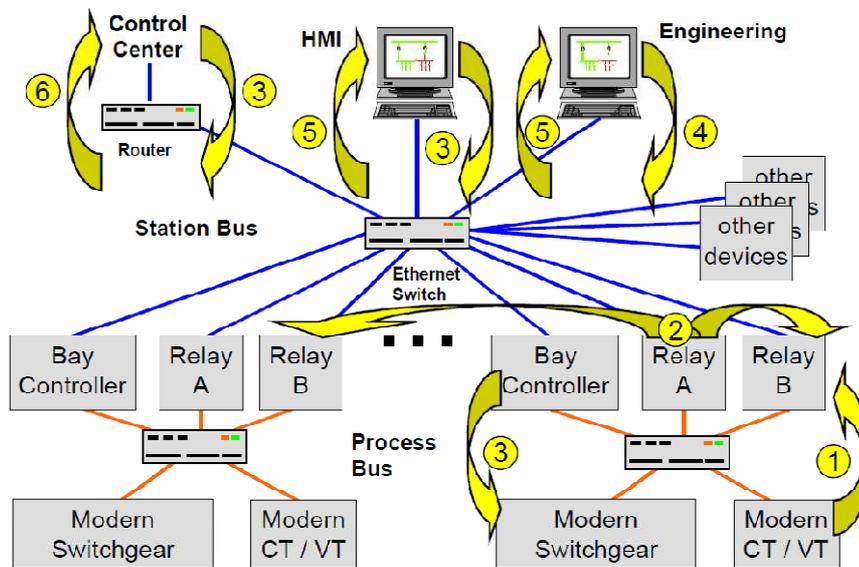


Figura 2-2. Ejemplo de topología de la automatización de una SAS. (Fuente:[5])

- Nivel de proceso: Es el nivel correspondiente a los dispositivos electrónicos (IEDs) que permiten el acceso a los equipos desde niveles superiores. Un ejemplo podría ser un PLC que controla y gestiona la información de un interruptor.
- Nivel de bahía o bloque eléctrico: Este nivel se corresponde con los IEDs que se encargan de controlar y proteger a los elementos de un determinado bloque eléctrico. Un relé es un ejemplo de este tipo de dispositivos.
- Nivel de subestación: Corresponde con el puesto de operación local de la subestación, desde el que el operador puede supervisar y gobernar los distintos aparatos. Como se puede ver en la Figura 2-2, puede tratarse o bien de una interfaz hombre-máquina (HMI) o de un puesto de trabajo remoto.

A la vez que los distintos niveles, la Figura 2-2 muestra los números que etiquetan las interfaces lógicas entre los dispositivos del SAS. Varias de estas interfaces pueden estar implementadas en un único conector físico y en una única LAN (bus). De hecho, la norma propone una configuración con dos buses que se repartirían las interfaces lógicas presentes en el sistema de comunicación:

- Bus de subestación. Alberga los intercambios de información que comunican los niveles de bahía y subestación o nivel de subestación con un puesto de control remoto.
- Bus de proceso. Alberga las interfaces lógicas correspondientes a la relación entre niveles de proceso y de bahía.

La disposición física de los buses depende de la división que se haya realizado previamente de los elementos de la subestación en distintos bloque eléctricos.

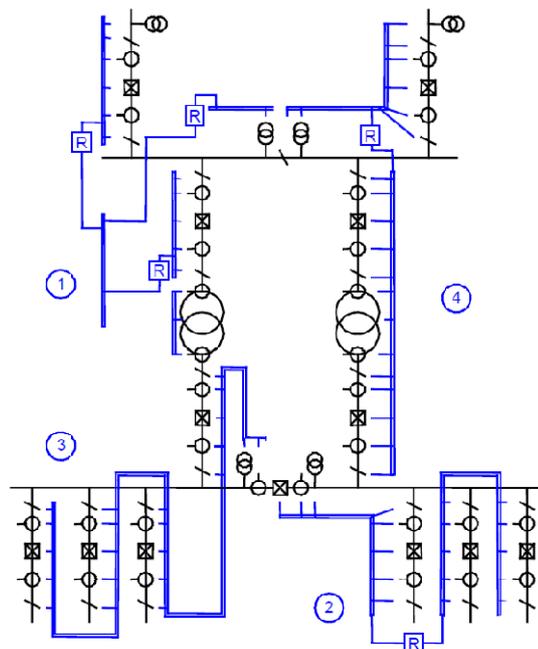


Figura 2-3. Esquema unifilar de subestación eléctrica de distribución con diversas alternativas para la disposición del bus de proceso. (Fuente:[3])

La Figura 2-3 muestra diversas alternativas para la disposición del bus de proceso, atendiendo a los requerimientos de flujo de datos, de robustez o bien simplemente a la facilidad de la instalación.

2.3 Funciones y Nodos Lógicos (LN)

Se definen aquí dos conceptos básicos en la norma, y su relación:

- **Funciones del SAS.** El sistema de automatización de la subestación eléctrica debe llevar a cabo una serie de funciones: de protección, control o supervisión. La norma estandariza cuál es la estructura que deben seguir las diferentes funciones de un SAS, las cuales deben estar compuestas por Nodos Lógicos.
- **Nodos Lógicos (LNs).** Son las unidades lógicas básicas que componen una función que debe realizar el sistema, y se alojan en los IEDs. La norma estandariza las distintas clases de LNs.

Se tomará el ejemplo presentado en [1] para ilustrar de una forma más detallada cuál es la relación que une a estos elementos:

Se propone un bloque eléctrico dentro de una subestación, en el que el SAS debe realizar tres **funciones**: protección de distancia, protección de sobreintensidad y función de comprobación de sincronismo. Según la norma, estas tres funciones deben estar formadas por distintas unidades básicas o **Nodos Lógicos** especificados perfectamente en ella. Por ejemplo, se define el LN *XCBB*, que contiene información necesaria para el control y supervisión de un interruptor; o el LN *TCTR*, que contiene la información de un transformador de intensidad

En general, toda función del SAS va a estar compuesta como mínimo por tres LN: uno correspondiente a la interfaz hombre máquina (IHM) para el control de la función desde el puesto de operación, un LN central de la función, que da nombre a la misma, y por último un LN correspondiente al nivel de proceso, responsable de la adquisición de medidas y accionamiento, si procede, de un elemento de protección.

¿Cuál es la disposición de estos Nodos Lógicos dentro de los IEDs? En el caso del ejemplo, el LN correspondiente al IHM se aloja en el Puesto Local de Operación de la subestación. Los correspondientes a las medidas de intensidad y tensión se alojan en IEDs de proceso que contienen información de los transformadores. Por su parte, el LN encargado del accionamiento del interruptor se aloja en un IED de

proceso que gestiona la información del bloque eléctrico. Los cálculos de las condiciones de sincronismo se realizan en un LN alojado en un relé.

La disposición de los Nodos Lógicos definidos en la norma dentro de los dispositivos físicos IEDs que componen un SAS concreto es muy flexible, ya que la norma no hace restricciones al respecto.

La Figura 2-4 recoge varias posibilidades de implantación de LN (PTOC, PDIS, PTRC, XCBR), dentro de uno o varios IEDs.

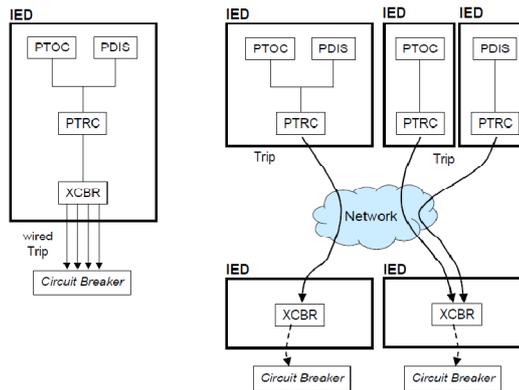


Figura 2-4. Distintas posibilidades de posicionamiento de LNs en IEDs. (Fuente:[5])

3. Modelo de información del SAS

La norma IEC-61850 presenta un modelo de información jerarquizado para la representación de un SAS. En él, cada Nodo Lógico contiene una serie de datos, los cuales se componen a su vez de atributos. De esta forma, se plantea una *virtualización* de la subestación eléctrica; es decir, que todo lo que hay dentro de la subestación se modelará mediante Nodos Lógicos.

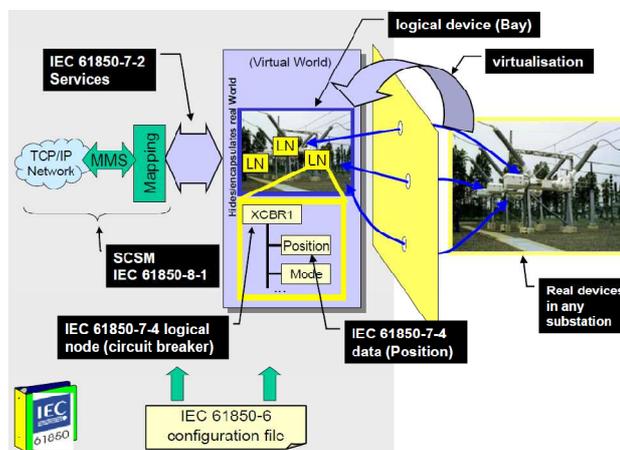


Figura 2-5. Virtualización de una subestación. (Fuente:[5])

La Figura 2-5 representa este proceso de virtualización, en el que se muestra el ejemplo del LN XCBR1, correspondiente a un elemento de control de un interruptor, que a su vez contiene una serie de **datos**. Los tipos de datos que contiene cada una de las clases genéricas de Nodos Lógicos están definidos en el apartado 7-4 de la norma. En el ejemplo ilustrado en la figura, se muestran los datos *Pos* y *Mode* correspondientes al LN XCBR1.

3.1 Clases genéricas de datos (CDC, *Common Data Classes*)

Como se indicó anteriormente, los datos que componen los LN tiene, a su vez, distintos atributos asociados. El apartado 7-3 de la norma define las **clases genéricas de datos** como estructuras para tipos de datos que comparten la organización y tipos de atributos, aunque tengan distinto significado.

La Tabla 2-1 presenta la codificación de clase de datos de Valores Medidos (*Measured Values*), que como se verá en los siguientes apartados resulta fundamental en este estudio. En la tabla se pueden observar los distintos atributos que componen la CDC MV.

Common data class MV (IEC 61850-7-3)		Coding in IEC 61850-9-1	Comment
Attribute name	Attribute type		
instMag	AnalogueValue		Not mapped
mag	AnalogueValue		
i f	INT32 FLOATING POINT32	UI16 -	Sampled analogue values of the universal data set according to IEC 60044-8.
range	ENUMERATED	-	Not mapped, see Note 1
q	Quality		
validity	CODED ENUM	BOOLEAN <0> = valid <1> = questionable, invalid	
detail-qual	PACKED LIST	-	Not mapped
source	CODED ENUM	-	Not mapped
test	BOOLEAN	-	Not mapped
operatorBlocked	BOOLEAN	-	Not mapped
t	TimeStamp	-	Not mapped, see Note 2

NOTE 1 According to IEC 61850-7-3, range is an optional attribute and is not required in the sampled value buffer format defined in IEC 61850-7-2.

NOTE 2 According to IEC 61850-7-3, t is a mandatory attribute. However, in the specification of the sampled value buffer format as defined in IEC 61850-7-2, t is not included with the data object values; there is only one sample counter attached that indicates the refresh of the universal data set sampled values as specified in IEC 60044-8.

Tabla 2-1. Definición de CDC Measured Values. (Fuente:[3])

3.2 Dispositivo Lógico (LD, *Logical Device*)

El Dispositivo Lógico es un concepto necesario para completar el modelo de información del SAS, y se refiere a un elemento de jerarquía inmediatamente superior al LN. Es decir, que un IED albergará distintos LDs, que a su vez contendrán una serie de LNs.

Todos los LDs contienen un mínimo de dos LNs: LLN0, que es el que contiene la información común a todo el LD; y LPHD, que contiene la información acerca del equipo físico original del LD.

¿Cuál es la finalidad de la introducción de este concepto? La necesidad de definir los LD tiene su origen en el modo de organización de la información según el modelo IEC-61850. Si bien en todo SAS existirán dispositivos físicos (IEDs) conectados directamente al bus IEC-61850, también es probable la presencia de otros dispositivos cuya información viaje hasta el bus a través de un concentrador, lo que obliga a crear un dispositivo lógico por cada dispositivo físico, con el fin de distinguir a qué equipo pertenece una determinada información.

La Figura 2-6 representa este modelo de información jerarquizado de manera esquemática.

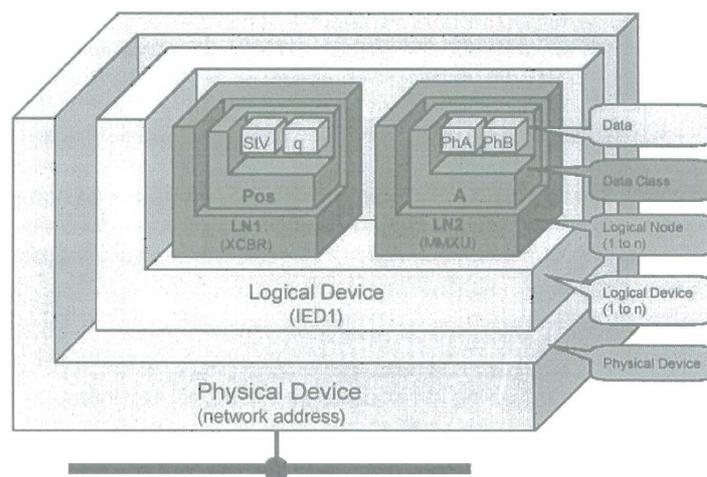


Figura 2-6. Estructura del modelo de datos. (Fuente:[1])

4. Modelo de Servicios Abstractos de Comunicación

4.1 ACSI, *Abstract Communication Service Interface*

La norma IEC-61850 en su apartado 7-2 define los Servicios Abstractos de Comunicación como los servicios que permiten que se intercambie información dentro del modelo del SAS descrito anteriormente.

Estos servicios son abstractos; es decir, a alto nivel. La obtención de tramas binarias o **PDU** (*Protocol Data Unit*) precisa la realización de un mapping específico en un sistema de comunicaciones determinado.

Los apartados 8-1, 9-1 y 9-2 contienen las especificaciones para llevarlo a cabo.

La Tabla 2-2 recoge los diferentes servicios abstractos para la comunicación de mensajes del modelo del SAS tal como se definen en [5].

<p><u>SERVER model (Clause 6)</u> GetServerDirectory</p> <p><u>ASSOCIATION model (Clause 7)</u> Associate Abort Release</p> <p><u>LOGICAL-DEVICE model (Clause 8)</u> GetLogicalDeviceDirectory</p> <p><u>LOGICAL-NODE model (Clause 9)</u> GetLogicalNodeDirectory GetAllDataValues</p> <p><u>DATA model (Clause 10)</u> GetDataValue SetDataValue GetDataDirectory GetDataDefinition</p> <p><u>DATA-SET model (Clause 11)</u> GetDataSetValues SetDataSetValues CreateDataSet DeleteDataSet GetDataSetDirectory</p> <p><u>Substitution model (Clause 12)</u> SetDataValue GetDataValue</p> <p><u>SETTING-GROUP-CONTROL-BLOCK model (Clause 13)</u> SelectActiveSG SelectEditSG SetSGValues ConfirmEditSGValues GetSGValues GetSGCBValues</p> <p><u>REPORT-CONTROL-BLOCK and LOG-CONTROL-BLOCK model (Clause 14)</u> BUFFERED-REPORT-CONTROL-BLOCK: Report GetBRCBValues SetBRCBValues UNBUFFERED-REPORT-CONTROL-BLOCK: Report GetURCBValues SetURCBValues</p>	<p><u>LOG-CONTROL-BLOCK model⁴₅</u> GetLCBValues SetLCBValues QueryLogByTime QueryLogAfter GetLogStatusValues</p> <p><u>Generic substation event model – GSE (Clause 15)</u> GOOSE SendGOOSEMessage GetGoReference GetGOOSEElementNumber GetGoCBValues SetGoCBValues GSSE SendGSSEMessage GetGsReference GetGSSEDataOffset GetGsCBValues SetGsCBValues</p> <p><u>Transmission of sampled values model (Clause 16)</u> MULTICAST-SAMPLE-VALUE-CONTROL-BLOCK: SendMSVMessage GetMSVCBValues SetMSVCBValues UNICAST-SAMPLE-VALUE-CONTROL-BLOCK: SendUSVMMessage GetUSVCBValues SetUSVCBValues</p> <p><u>Control model (Clause 17)</u> Select SelectWithValue Cancel Operate CommandTermination TimeActivatedOperate</p> <p><u>Time and time synchronization (Clause 18)</u> TimeSynchronization</p> <p><u>FILE transfer model (Clause 20)</u> GetFile SetFile DeleteFile GetFileAttributeValues</p>
--	--

Tabla 2-2. Clases de servicios ACSI definidos. (Fuente:[6])

De la lista se puede observar que existen servicios asociados a LD (*GetLogicalDeviceDirectory*), LN (*GetLogicalNodeDirectory*), Datos (*GetDataValue*) o Data-Set (*CreateDataSet*). Se definen los **Data-Set** como conjuntos de datos que se agrupan con fines relacionados con la generación de mensajes informando sobre algún evento. Por ejemplo, si se produce un cambio en un atributo de un dato y éste pertenece a un Data-Set, se generará un mensaje de evento incluyendo todos los

atributos de ese dato que sean de tipo *status*, como pueden ser el nuevo valor, la calidad de la medida o la etiqueta del tiempo del cambio.

Dentro de los tipos de ACSI, se encuentran algunos relacionados con mensajes especiales, que se describen a continuación. Su existencia responde a la necesidad que tiene el sistema de automatización de enviar determinados mensajes entre distintos IEDs de manera rápida y segura; y requieren un mapping concreto.

Antes de detallar las definiciones de estos ACSI particulares, es necesario destacar el concepto de **clases**, descrito en la norma IEC-61850-7-2 y necesario para la composición del modelo de información del SAS. Sirva la Tabla 2-3 como ejemplo ilustrativo, donde se puede apreciar que cada clase va a tener asociados una serie de atributos y de servicios.

DATA-SET class		
Attribute name	Attribute type	Value/value range/explanation
DSName	ObjectName	Instance name of an instance of DATA-SET
DSRef	ObjectReference	Path-name of an instance of DATA-SET
DSMemberRef [1..n]	(*)	(*) Functional constrained data (FCD) or functional constrained data attribute (FCDA)
Services GetDataSetValue SetDataSetValue CreateDataSet DeleteDataSet GetDataSetDirectory		

Tabla 2-3. Definición de clase Data-Set y servicios asociados. (Fuente:[6])

4.2 GSE, *Generic Substation Event*

El primero de estos servicios especiales es el GSE, que viene a ser el servicio que permite la comunicación de eventos genéricos de la subestación a varios dispositivos IEDs dentro del sistema de manera simultánea, rápida y segura. Está relacionado con todo tipo de acciones automáticas que se deben llevar a cabo dentro del SAS y que requieren del intercambio de información entre distintos dispositivos de la subestación con una importante limitación temporal.

Un ejemplo de GSE que se transmite podría ser el cambio en el valor de posición de un determinado interruptor cuyo estado no sólo interesa al IED que se encarga de controlarlo, sino que afecta a otros IEDs del

sistema, los cuales necesitarán conocer los cambios que se produzcan en dicho elemento en el menor tiempo posible para el correcto funcionamiento de un sistema de automatización de una subestación eléctrica.

Para poder proporcionar este servicio, la norma en su apartado 7-2 define dos clases que representan dos bloques de control: GoCB, que permite el envío de mensajes GOOSE (*Generic Object Oriented Substation Event*), referidos a cualquier tipo de información accesible desde un Data-Set asociado; y GsCB, para el envío de mensajes de tipo GSSE (*Generic State Substation Event*), que simplemente indican un cambio de alguno de los estados que se incluyen en una lista GsCB.

4.3 TSV, *Transmission of Sampled Values*

Este servicio permite la transmisión de valores instantáneos de medidas analógicas (SAV), minimizando el tiempo que transcurre desde el muestreo hasta la recepción del mensaje.

Para proporcionar este servicio, se definen en [6] dos clases que representan bloques de control: MSVCB (*Multicast Sample Values Control Block*, Tabla 2-4) y USVCB (*Unicast Sample Values Control Block*).

MSVCB class				
Attribute name	Attribute type	FC	TrgOp	Value/value range/explanation
MsvCBNam	ObjectName	-	-	Instance name of an instance of MSVCB
MsvCBRef	ObjectReference	-	-	Path-name of an instance of MSVCB
SvEna	BOOLEAN	MS	dchg	Enabled (TRUE) disabled (FALSE), DEFAULT FALSE
MsvID	VISIBLE STRING65	MS	-	
DatSet	ObjectReference	MS	dchg	
ConfRev	INT32U	MS	dchg	
SmpRate	INT16U	MS	-	(0..MAX)
OptFlds	PACKED LIST	MS	dchg	
refresh-time	BOOLEAN			
sample-synchronized	BOOLEAN			
sample-rate	BOOLEAN			
data-set-name	BOOLEAN			
Services				
SendMSVMessage				
GetMSVCBValues				
SetMSVCBValues				

Tabla 2-4. Definición de clase MSVCB y servicios asociados. (Fuente:[6])

La diferencia entre ambos bloques de control estriba en que si bien MSVCB permite el envío de mensajes TSV a varios IEDs, el bloque USVCB sólo permite el envío a un IED concreto.

A lo largo de este Capítulo se ha procurado realizar un resumen suficientemente amplio para conocer los conceptos básicos definidos en la IEC-61850 que se emplearán posteriormente en esta memoria.

La Figura 2-7 muestra, de manera esquemática, la estructura del modelo de datos presentado en la norma así como de los tipos de datos, servicios y mensajes descritos con anterioridad.

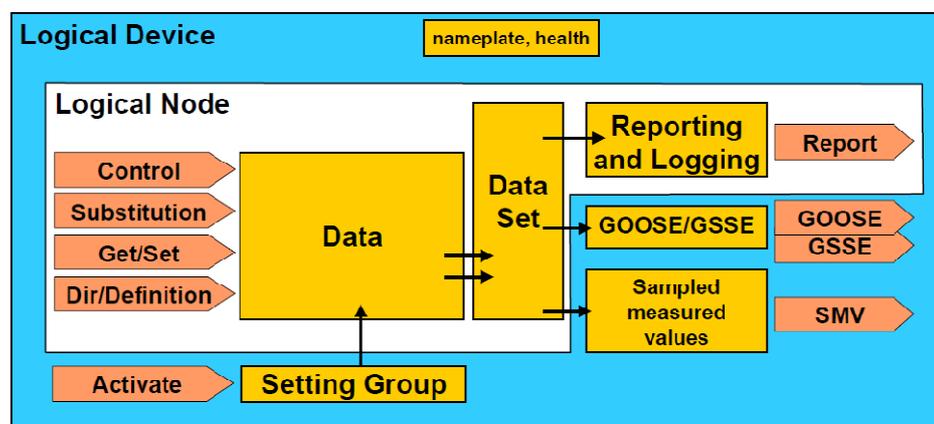


Figura 2-7. Estructura del modelo de datos de un LD. (Fuente:[5])

En concreto, el objetivo de este proyecto es el desarrollo de una plataforma capaz de realizar el envío de mensajes SV (*Sampled Value*), cuyo mapping se detallará más adelante.

5. Normas que componen la IEC-61850

Una vez introducidos los conceptos más importantes a los que se refiere la norma, se enumeran a continuación los diferentes apartados que contiene:

IEC 61850-1: Introduction and Overview. Introduce conceptos básicos.

IEC 61850-2: Glossary. Glosario de términos.

IEC 61850-3: General requirements. Requisitos de calidad, condiciones ambientales y servicios auxiliares.

IEC 61850-4: System and Project management. Gestión de sistemas y proyectos.

IEC 61850-5: Communication requirements for functions and device models. Requisitos del sistema de comunicación y de los equipos.

IEC 61850-6: Substation Configuration Language (SCL). Explica el lenguaje empleado para la configuración de los IEDs

IEC 61850-7: Basic communication structure. Explica el modelo de información y servicios de comunicación

IEC 61850-7-1: Principles and models. Visión general de 61850-7

IEC 61850-7-2: Abstract Communication Service Interface (ACSI).

Define los servicios de comunicación necesarios para el intercambio de datos en el modelo de información

IEC 61850-7-3: Common Data Classes (CDC). Define el modelo de información de la subestación eléctrica.

IEC 61850-7-4: Compatible Logical Node classes and data classes.

Define el modelo de información de la subestación eléctrica.

IEC 61850-8-1: Specific Communication Service Mapping (SCSM) – Mapping to MMS. Mapping del modelo de información y servicios de comunicación abstractos a un sistema de comunicaciones concreto. Mapping en MMS.

IEC 61850-9-1: Specific Communication Service Mapping (SCSM) – Serial unidirectional multidrop point link. Mapping del modelo de información y servicios de comunicación abstractos a un sistema de comunicaciones concreto.

IEC 61850-9-2: Specific Communication Service Mapping (SCSM) – Mapping to IEEE 802.3 based process bus. Mapping del modelo de información y servicios de comunicación abstractos a un sistema de comunicaciones concreto.

IEC 61850-10: Conformance Testing. Pruebas de validez.

Como se comentó en el punto 4, este proyecto requiere una mayor profundización en los apartados referentes al mapping del modelo en un sistema de comunicación físico, por lo que se reservan los siguientes puntos para dicho análisis.

6. Apartado 9-1

Los apartados IEC-61850-9-x recogen la explicación para la realización del mapping de los servicios TSV.

En concreto, la parte IEC-61850-9-1 trata el mapping de los servicios relacionados con el envío de mensajes TSV a varios dispositivos de manera simultánea; es decir, *SendMSVMessage*.

La Figura 2-8 recoge un ejemplo ilustrativo del sistema de comunicación unidireccional para este tipo de servicio. Como se puede observar, una unidad concentradora (*Merging unit*) recibe la información tanto de los transformadores de la línea como de sus entradas binarias y de sincronización y monitorización. Los datos recogidos son enviados por el dispositivo a los distintos IEDs del nivel de bahía, como podrían ser un relé de protección y una unidad controladora PLC.

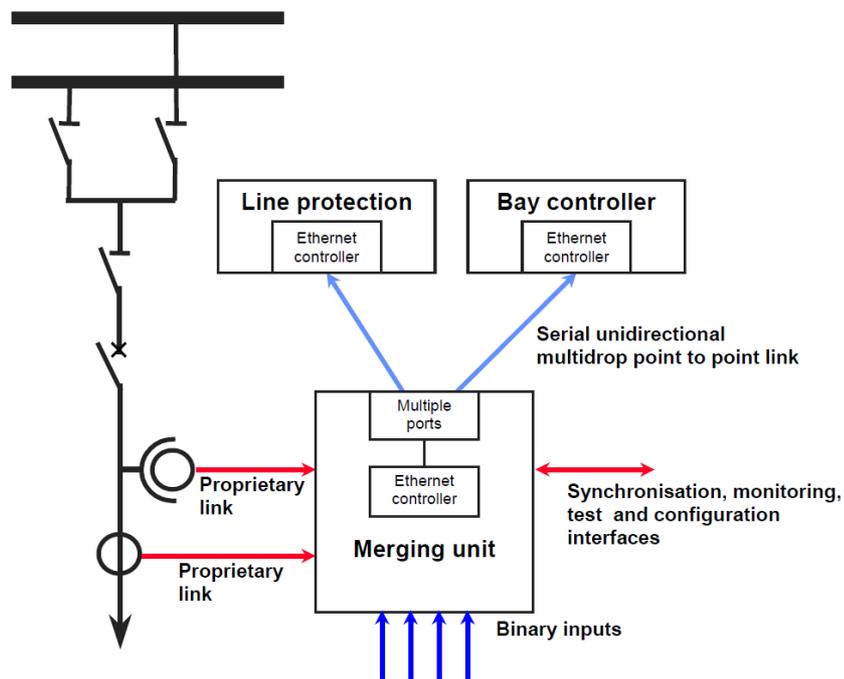


Figura 2-8. Transmisión de mensajes TSV desde una unidad concentradora a varios IEDs. (Fuente: [2])

De esta forma, la norma en su apartado 9-1 puntualiza que únicamente se restringe a la descripción de la comunicación entre transformadores ECT/EVT (*Electronic Current/Voltage Transformer*) y los dispositivos

situados en el nivel de bahía a través de una unidad concentradora, basándose en las normas IEC-60044-7 e IEC-60044-8 de descripción de transformadores cuyo estudio no tiene cabida en este proyecto.

La Tabla 2-5 muestra de manera esquemática el mapping de estos servicios es un sistema de comunicaciones tomando como referencia los niveles OSI de una capa de comunicación. El proyecto presentado se desarrolla dentro de las capas física y de enlace del mapping mostrado, que como se puede apreciar se basan en los estándares IEEE 802.3, IEEE 802.1Q, e ISO/IEC 8802-3; es decir, en las directrices del modelo de comunicación Ethernet.

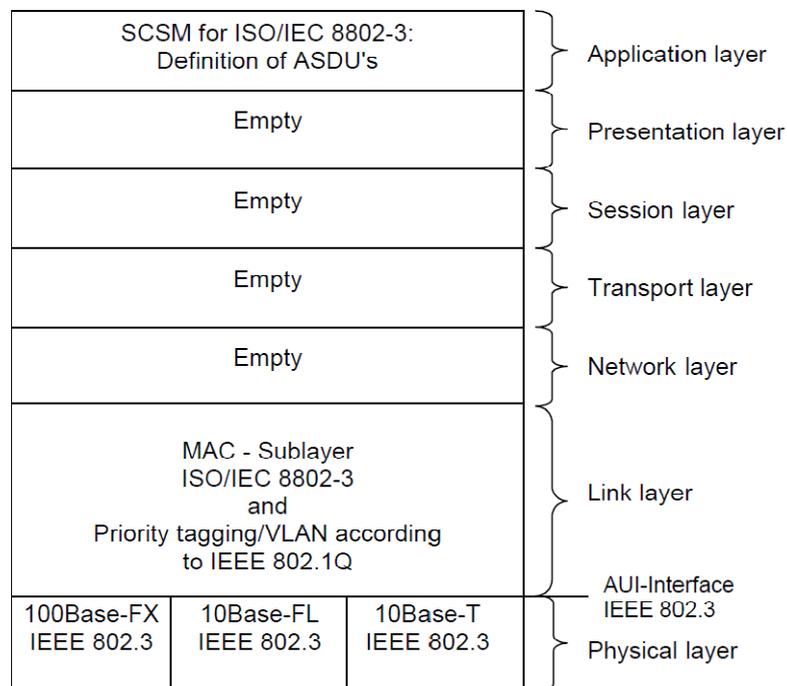


Tabla 2-5. Stack de comunicación. (Fuente: [2])

6.1 Capa física

La parte 9-1 de la norma recoge como preferencia para la transmisión de datos el uso de la fibra óptica (100Base-FX), si bien deja abierta la posibilidad del empleo de una capa basada en transmisión por par trenzado siempre y cuando se tengan en cuenta las restricciones de acoplamientos magnéticos y por tanto los aislamientos necesarios.

6.2 Capa de enlace

En cuanto a la capa de enlace, se presentan en [1] las siguientes especificaciones:

- La dirección de destino del enlace debe ser por defecto la dirección *broadcast*, que consiste en una secuencia de unos en el campo *Destination Adress*, aunque se deja abierta la posibilidad de que sea configurable para, por ejemplo la conexión de una unidad concentradora vía *switch* a los niveles superiores. Se especifica sin embargo que una única dirección Ethernet debe ser usada como dirección de origen (*Source Address*).
- La especificación respecto al uso de los campos de prioridad y LAN Virtual se hace en base al estándar recogido en IEEE 802.1Q, de manera que el mapping del modelo permita la separación de los valores analógicos y de los mensajes GOOSE de protección, cuya temporización es crítica y por tanto de mayor prioridad. La Figura 2-9 detalla esta estructura:

Octets		8	7	6	5	4	3	2	1
1	TPID	0x8100							
2									
3	TCI	User priority			CFI		VID		
4		VID							

Figura 2-9. Estructura de la cabecera prioridad y VLAN. (Fuente: [2])

- TPID. Con valor 0x8100.
- *User priority*. Permitirá la diferenciación entre mensajes críticos y no críticos.
- CFI (*Canonical Formal Indicator*). Indica si al campo *Length/Type* le sigue otro denominado RIF (*Resource Identification Field*).
- VID (*Virtual LAN Identifier*). La norma presenta este campo como configurable, siempre y cuando el protocolo VLAN sea soportado por el sistema. En caso contrario, debe estar por defecto a cero.

- El valor del campo *Ether-type* se corresponde con el reservado por la *IEEE Authority Registration* para el envío de valores analógicos muestreados (SAV, *Sampled Analogue Value*).

Asimismo, como se puede observar en la Figura 2-10, la norma ofrece una estructura para el campo completo *Ether-type PDU* (*Protocol Data Unit*), cuya división se detalla a continuación.

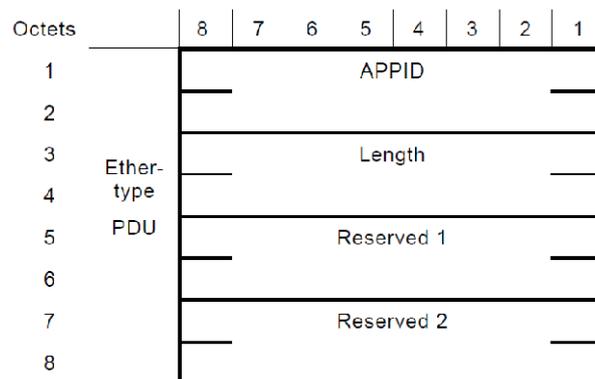


Figura 2-10. Estructura del *Ether-type PDU*. (Fuente: [2])

- *APPID* (*Application Identifier*). Se emplea para el envío de tramas Ethernet que contienen SAV y para distinguir el tipo de aplicación. Los valores reservados para este campo van del 0x4000 al 0x7FFF.
 - *Length*. Indica el número de octetos dentro del PDU.
 - *Reserved*. Estos campos se rellenan con ceros ya que están reservados a futuras aplicaciones aún no definidas.
- Por último, respecto a la capa de enlace, la norma presenta una restricción referente a la direccionalidad de la información, ya que se especifica que la transmisión de SAV se efectúa únicamente desde la unidad concentradora hacia el nivel de bahía; es decir, que la bidireccionalidad sólo se plantea en el caso de que sea necesaria para un correcto funcionamiento de la capa física del medio, sin que por ello deba influir en la correcta transmisión unidireccional.

6.3 Mapping del modelo de transmisión de valores muestreados usando multicast

El documento [1] recoge las definiciones de dos tipos de Data-Set en consonancia con la norma IEC-60044 (apartados 7 y 8), que se pueden encontrar en los anexos A y C de la referencia indicada, y se presentan a continuación:

6.3.1 Universal Data Set

La Figura 2-11 corresponde con la definición del Data-Set que, como se observa, alberga la información relativa a los valores analógicos muestreados de los diferentes transformadores, así como los parámetros referentes a la identificación del conjunto de datos (*DataSetName*) dentro del equipo (*LNNName*, *LDName*) y a las características de la transmisión (*Sampling Rate*, *Sample Counter*).

		2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
Byte 1	ASDU Header	msb Length of ASDU (- 44)							
Byte 2		lsb							
Byte 3	ASDU (universal data set)	msb LNNName (=02)							
Byte 4		lsb DataSetName (=01)							
Byte 5	ASDU (universal data set)	msb LDName							
Byte 6		lsb							
Byte 7	ASDU (universal data set)	msb Rated Phase Current							
Byte 8		lsb							
Byte 9	ASDU (universal data set)	msb Rated Neutral Current							
Byte 10		lsb							
Byte 11	ASDU (universal data set)	msb Rated Phase Voltage							
Byte 12		lsb							
Byte 13	ASDU (universal data set)	msb Rated Delay Time							
Byte 14		lsb							
Byte 15	ASDU (universal data set)	msb Current Phase A, prot.							
Byte 16		lsb							
Byte 17	ASDU (universal data set)	msb Current Phase B, prot.							
Byte 18		lsb							
Byte 19	ASDU (universal data set)	msb Current Phase C, prot.							
Byte 20		lsb							
Byte 21	ASDU (universal data set)	msb Current Neutral							
Byte 22		lsb							
Byte 23	ASDU (universal data set)	msb Current Phase A, mes.							
Byte 24		lsb							
Byte 25	ASDU (universal data set)	msb Current Phase B, mes.							
Byte 26		lsb							
Byte 27	ASDU (universal data set)	msb Current Phase C, mes.							
Byte 28		lsb							
Byte 29	ASDU (universal data set)	msb Voltage Phase A							
Byte 30		lsb							
Byte 31	ASDU (universal data set)	msb Voltage Phase B							
Byte 32		lsb							
Byte 33	ASDU (universal data set)	msb Voltage Phase C							
Byte 34		lsb							
Byte 35	ASDU (universal data set)	msb Voltage Neutral							
Byte 36		lsb							
Byte 37	ASDU (universal data set)	msb Busbar Voltage							
Byte 38		lsb							
Byte 39	ASDU (universal data set)	msb StatusWord#1							
Byte 40		lsb							
Byte 41	ASDU (universal data set)	msb StatusWord#2							
Byte 42		lsb							
Byte 43	ASDU (universal data set)	msb Sample Counter							
Byte 44		lsb							
Byte 45	ASDU (universal data set)	msb Sampling rate							
Byte 46		lsb Configuration revision no.							

Figura 2-11. Estructura y campos del Universal Data-Set. (Fuente: [2])

La siguiente tabla recoge la información relevante a los campos de la figura anterior, y su definición:

Attribute	Type	Definition
DataSetName	INTEGER	For the universal data set according to IEC 60044-8 the integer value is set to 1
Data-Reference	See next table	

Logical node instance name	Data name	Common data class	Definition according to IEC 60044-8
phsaTCTR	ARtg	ASG	Rated phase current Defines the rated current in Ampere r.m.s.
neutTCTR	ARtg	ASG	Rated neutral current Defines the rated neutral current in Ampere r.m.s.
phsaTVTR	VRtg	ASG	Rated phase voltage Defines the rated voltage in 1/10 kV r.m.s.
	Tdr	SAV	Rated delay time Defines the rated delay time in μ s. The rated delay time indicates the time between the instant a certain current/voltage is present at the primary terminals and the instant the transmission of the belonging digital data set starts. According to this standard, synchronisation pulses are used to synchronise several merging units. Therefore the rated delay time is not relevant for the ECT/EVT accuracy. The rated delay time value shall be high enough to allow reasonable antialiasing filters in the merging unit, but it shall not be so high that it significantly affects protection device performance. Therefore the rated delay time for this standard should be 3 000 μ s (tolerance band -100 % to +10 %) for all sampled rates. Tdr is not defined in IEC 61850-7-4.
phsaTCTR	Amp	SAV	Current phase A, used for protection
phsbTCTR	Amp	SAV	Current phase B, used for protection
phscTCTR	Amp	SAV	Current phase C, used for protection
neutTCTR	Amp	SAV	Current neutral
phsaTCTR1	Amp	SAV	Current phase A; different scaling; used for measurement
phsbTCTR1	Amp	SAV	Current phase B; different scaling; used for measurement
phscTCTR1	Amp	SAV	Current phase C; different scaling; used for measurement
phsaTVTR	Vol	SAV	Voltage phase A
phsbTVTR	Vol	SAV	Voltage phase B
phscTVTR	Vol	SAV </td <td>Voltage phase C</td>	Voltage phase C
neutTVTR	Vol	SAV	Voltage neutral
bbTVTR	Vol	SAV	Busbar Voltage

Tabla 2-6. Definiciones referentes al Universal DataSet. (Fuente: [2])

Como se puede observar en la Figura 2-12, los diferentes campos corresponden a las entradas analógicas de la unidad concentradora, referidas a los secundarios de protección y medida de los transformadores ECT/EVT.

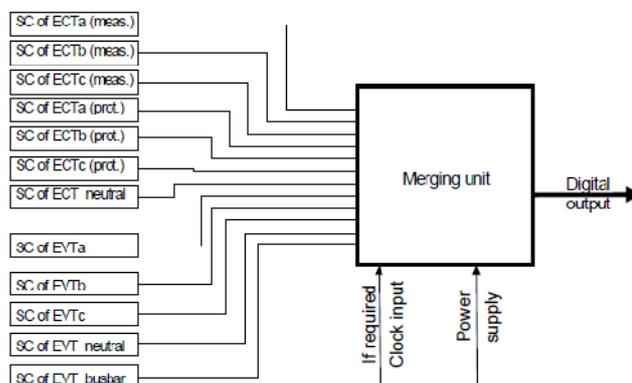


Figura 2-12. Configuración de adquisición de datos de ECT/EVT. (Fuente: [2])

6.3.2 Status Indication Data Set

La Figura 2-13 muestra la definición del Data-Set de Indicación de Estado que acompaña a la trama Universal.

Octets		8	7	6	5	4	3	2	1	
1	Length	msb								
2		Length = 21								
3	Data set reference	msb								
4		LNName = 2								
5		DataSetName = 2								
6		msb								
		LDName								
7	SIG	S16	S15	S14	S13	S12	S11	S10	S9	
8		S8	S7	S6	S5	S4	S3	S2	S1	
9		Q16	Q15	Q14	Q13	Q12	Q11	Q10	Q9	
10		Q8	Q7	Q6	Q5	Q4	Q3	Q2	Q1	
11		RE	RE	RE	IC	OD	EE	OS	IV	
12			2^{31}						2^{24}	
13			2^{23}						2^{16}	
14			2^{15}						2^8	
15			2^7						2^0	
16			RE	LK	Reserved			NS	CF	
17			RE			2^{16}				
18		2^{15}						2^8		
19		2^7						2^0		
20	SC	msb								
21		Counter								
22	SR	msb								
		Sample Rate								
23	CR	msb								
		Configuration Revision								
		lsb								

Figura 2-13. Estructura y campos del Status Indication Data-Set. (Fuente: [2])

Al contrario que el Data-Set descrito anteriormente, éste recoge información referente a las entradas binarias de la unidad concentradora (Figura 2-8) así como de la calidad de las medidas.

La intención de la existencia de este tipo de trama es ofrecer a los dispositivos la posibilidad de transmitir información sobre el estado de las medidas de manera que se evite otro tipo de implementación independiente.

De manera meramente ilustrativa se presenta en la Tabla 2-7 la codificación de la clase de datos SPS (*Single Point Status*), empleada para la definición de los diferentes campos que aparecen en la trama del *Status Indication Data-Set*.

Common data class SPS (IEC 61850-7-3)		Coding in IEC 61850-9-1	Comment
Attribute name	Attribute type		
stVal	BOOLEAN	-	Not mapped
grpVal	BIT STRING	BS16 <0> = FALSE, OFF <1> = TRUE, ON	16 individual status values. See Annex A
q	Quality	BS16 <0> = INVALID <1> = VALID	16 individual quality indications related to the status values. See Annex A
validity (IV)	CODED ENUM	BS1 [0] <0> = INVALID <1> = VALID	Further elements of detail-qual not mapped
detailQual	PACKED LIST	BS7 [1] = oscillatory (OS) [2] = failure, external error (EE) [3] = oldData (OD) [4] = inconsistent (IC) [5..7] = reserved (RE)	
source	CODED ENUM	-	
test operatorBlocked	BOOLEAN BOOLEAN	- -	
t	TimeStamp		Not mapped Not mapped Not mapped
SecondsSinceEpoch	INT32	UI32	
FractionOfSecond	INT24	UI21	
Reserved	BIT STRING	BS3 [21 .. 23]	
TimeQuality	TimeQuality		
ClockFailure (CF)	BOOLEAN	BS1 [0] <0> = FALSE, <1> = TRUE.	Time function is unreliable
ClockNotSynchronized (NS)	BOOLEAN	BS1 [1] <0> = FALSE, <1> = TRUE.	Clock is not synchronized to the reference source
TimeAccuracy	CODED ENUM	UI3	Reserved
LeapSecondsKnown (LK)	BOOLEAN	BS1 [5] <0> = FALSE <1> = TRUE	
Reserved (RE)	BIT STRING	BS2 [6..7]	

Tabla 2-7. Codificación de la clase de datos SPS. (Fuente: [2])

Sin embargo, huelga para la realización de este proyecto un análisis más detallado, debido a que no es de interés el estudio de un Data-Set concreto definido en base a la norma IEC-60044.

Resulta más interesante sin duda detenerse en la definición de la clase de datos MV (*Measured Values*), recogida en la Tabla 2-8:

Common data class MV (IEC 61850-7-3)		Coding in IEC 61850-9-1	Comment
Attribute name	Attribute type		
instMag	AnalogueValue		Not mapped
mag	AnalogueValue		
i f	INT32 FLOATING POINT32	UI16 -	Sampled analogue values of the universal data set according to IEC 60044-8.
range	ENUMERATED	-	Not mapped, see Note 1
q	Quality		
validity	CODED ENUM	BOOLEAN <0> = valid <1> = questionable, invalid	Not mapped Not mapped Not mapped Not mapped
detail-qual	PACKED LIST	-	
source	CODED ENUM	-	
test operatorBlocked	BOOLEAN BOOLEAN	- -	
t	TimeStamp	-	Not mapped, see Note 2
NOTE 1 According to IEC 61850-7-3, range is an optional attribute and is not required in the sampled value buffer format defined in IEC 61850-7-2.			
NOTE 2 According to IEC 61850-7-3, t is a mandatory attribute. However, in the specification of the sampled value buffer format as defined in IEC 61850-7-2, t is not included with the data object values: there is only one sample counter attached that indicates the refresh of the universal data set sampled values as specified in IEC 60044-8.			

Tabla 2-8. Codificación de la clase de datos MV. (Fuente: [2])

Del análisis de la tabla y el tipo de codificación se pueden extraer dos conclusiones:

- En cuanto a los valores analógicos, hay dos campos que son de interés para cada medida, y que son los que realmente se mapean según este apartado de la norma: La magnitud de la medida (*mag*), y la calidad de ésta (*q*).
- A la hora de codificar estos dos campos físicamente, se define la magnitud como entero sin signo de 16 bits (UI16); y la calidad, en concreto su dato validez (*validity*) como un dato binario (BOOLEAN), para indicar si la medida es válida o cuestionable.

Posteriormente a lo largo de esta memoria se detallará cómo se realiza la implementación de la trama física de valores analógicos, aunque como adelanto se puede puntualizar que ambos tipos de datos (magnitud y calidad de cada medida) se comportan como entradas de la plataforma de comunicación.

6.4 Trama física y requisitos de envío

Una vez definidos los tipos de datos que se van a enviar y su estructura dentro de los Data-Set, queda por determinar cómo será el encapsulado de los mismos en la trama física que se transmitirá vía Ethernet.

6.4.1 Estructura de la trama

Como se pudo observar en la Tabla 2-5, el mapping bajo la norma IEC-61850-9-1 pasa directamente de la capa de enlace a la capa de aplicación dentro del *stack* de comunicación. Es precisamente en este nivel donde se define el encapsulado que será necesario enviar, y que toma la forma de **APDU** (*Application-Protocol Data Unit*). Esta estructura puede observarse en la Figura 2-14.

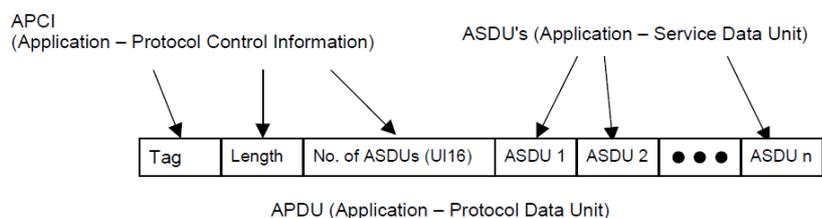


Figura 2-14. Concatenación de ASDUs dentro de un APDU. (Fuente: [2])

La organización mostrada presenta diferentes campos que se describen a continuación:

- *ASDU (Application-Service Data Unit)*. Constituyen cada uno de los Data-Set necesarios para la transmisión. La norma permite la concatenación de más de un ASDU dentro de un APDU.
- *No. of ASDUs*. Este campo se define como entero de 16 bits sin signo, y recoge el número de Data-Set que se incorporan a la trama. Como se puede suponer, limita el número de ASDUs que se pueden concatenar, aunque en un número elevado (65535).
- *Tag-Length*. Estos dos campos se añaden como cabecera de la trama del buffer de transmisión de valores analógicos, y se codifican según las reglas BER (*Basic Encoding Rules*) del estándar ASN.1, que se detallará más adelante en este Capítulo.

Se ha descrito por lo tanto la estructura del buffer de transmisión de los valores analógicos muestreados, según la cual varias unidades de datos (Data-Set) se encadenan para formar una trama con cabecera propia (APDU) que debe ser enviada.

Pero ¿cuáles son las directrices para realizar dicha concatenación? Evidentemente la norma restringe el envío de ASDUs dentro de un APDU, teniendo en cuenta un factor que hasta el momento no se ha comentado, aunque resulta fundamental a la hora de realizar un dispositivo de comunicación a nivel físico: La frecuencia de muestreo.

6.4.2 Frecuencia de muestreo

La frecuencia de muestreo influye tanto en la toma como en la transmisión de los datos analógicos. El apartado 9-1 de la norma ofrece en uno de sus anexos una guía para seleccionar la velocidad de la transmisión física Ethernet necesaria en función de la frecuencia de muestreo (Tabla 2-9).

Se observa de la tabla que la frecuencia de muestreo está en función de la *rated frequency*, que se corresponde con los 50 o 60Hz del sistema eléctrico sobre el que se realizan medidas.

Sampling rate	1	
$10 \times f_r$	10 Mbps	
$12 \times f_r$	10 Mbps	
$16 \times f_r$	10 Mbps	
$20 \times f_r$	10 Mbps	Rated value according to IEC 60044-8
$40 \times f_r$	10 Mbps	
$48 \times f_r$	10 Mbps	Rated value according to IEC 60044-8
$80 \times f_r$	10 Mbps	Rated value according to IEC 60044-8
$200 \times f_r$	100 Mbps	
$400 \times f_r$	100 Mbps	
f_r : Rated frequency (Hz).		

Tabla 2-9. Guía de selección para la capa física Ethernet. (Fuente: [2])

Por lo tanto, los extremos contemplados por la norma van desde un muestreo de 500 muestras por segundo a uno de 20000 muestras por segundo, si tomamos como referencia los 50Hz.

Una vez conocidas las posibles frecuencias de muestreo del SAS, la norma IEC-61850-9-1 establece las siguientes indicaciones en referencia al envío de los datos:

- El refresco del buffer de transmisión debe ser igual que la frecuencia de actualización de la comunicación del sistema.
- Solamente se puede almacenar un APDU en el buffer de transmisión, lo que implica una sobreescritura de información.
- Para evitar que de esta manera se pierdan datos, el envío debe ser inicializado inmediatamente después de que la actualización del buffer de transmisión se haya completado.
- Dicha actualización se realiza en función del periodo de muestreo siguiendo dos posibilidades:
 - *Refresh rate=Sampled rate*. Implica que el APDU se sitúe en el buffer de transmisión tras la toma de datos.
 - *Refresh rate=Sampled rate/n*. Implica que más de un ASDU forme parte del APDU de manera que éste se sitúe en el buffer de transmisión únicamente cuando se hayan tomado n muestras.

En resumen, la relación entre las ASDUs y el APDU viene dado por la frecuencia de muestreo, y por un parámetro n configurable, que equivale al número de muestras que contiene cada APDU. Si por cada

muestra que se toma se necesita un único ASDU, el valor de n se corresponde con el del número de ASDUs contenidas en el APDU.

7. Apartado 9-2

En el punto anterior se ha analizado cuál es el mapping propuesto en el apartado 9-1 de la IEC-61850, en el que se trata el caso de una unidad concentradora (*merging unit*) que envía información obtenida de unos transformadores ECT/EVT conectados a la línea eléctrica hacia un nivel superior de dispositivos IEDs de bahía, cuya representación se pudo observar en la Figura 2-8.

Como ya se ha visto, este modelo es limitado, debido a que se ajusta a la norma IEC-60044 para modelar los dispositivos, y deja sin resolver ciertas cuestiones necesarias para realizar un mapping que permita una configuración suficientemente robusta y configurable para que pueda ser adaptada a cualquier tipo de dispositivo, que es uno de los objetivos iniciales del proyecto.

Surge de esta manera la IEC-61850-9-2, que define el SCSM (*Specific Communication Service Mapping*) para la transmisión de valores analógicos de acuerdo con las especificaciones de niveles abstractos definidos en el apartado 7-2. Dicho mapping se realiza directamente sobre el nivel de enlace siguiendo las directrices del estándar ISO/IEC 8802-3; es decir: Ethernet.

Tomando como base lo ya expuesto en el punto 6 de este capítulo, se describirán a continuación únicamente las especificaciones contenidas en el apartado 9-2 que o bien no aparezcan en el apartado 9-1, o bien complementen la información contenida en el mismo.

7.1 *Stack* de comunicación

La definición del *stack* de comunicación es el primer paso para la obtención de un SCSM. En este caso, al igual que en la 9-1, se emplea como referencia el modelo OSI para situar el mapping dentro de la capa de comunicación correspondiente. La Figura 2-15 muestra el modelo.

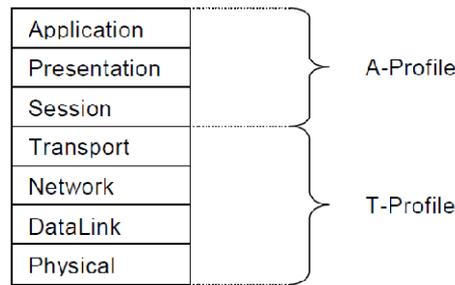


Figura 2-15. Stack de comunicación referencia. (Fuente:[3])

Se definen dentro de la norma dos grupos de especificaciones (*Profiles*):

- *A-Profile*. Referente a los niveles de comunicación de Aplicación, Presentación y Sesión.
- *T-Profile*. Referente a los niveles Físico, Enlace, Red y Transporte.

Estos dos tipos de especificaciones son empleados en la definición de las dos clases de servicios ACSI contemplados en este apartado de la IEC-61850:

Servicios Cliente/Servidor basados en MMS (*Manufacturing Message Specification*). Estos servicios están descritos en el apartado 8-1 de la norma, y se refieren a las especificaciones del SAS a alto nivel. Un análisis en profundidad de este tipo de servicios no tiene cabida en este documento.

Servicios SV basados en la capa de enlace. Concretamente, se refiere a los servicios *SendMSVMessage* y *SendUSVMessage* de la Tabla 2-2.

Se detallará el mapping necesario para el envío de este tipo de mensajes, debido a que es el objetivo fundamental del proyecto.

Las siguientes tablas (Tabla 2-10 y Tabla 2-11) recogen las especificaciones de cada nivel del *stack* de comunicación para llevar a cabo los servicios SV descritos.

OSI model layer	Specification			m/o
	Name	Service specification	Protocol specification	
Application	SV service			m
Presentation	Abstract syntax	ISO/IEC 8824-1:1999	ISO/IEC 8825	m
Session				

Tabla 2-10. A-Profile para los servicios SV. (Fuente:[3])

OSI model layer	Specification			m/o
	Name	Service specification	Protocol specification	
Transport				
Network				
DataLink	Priority tagging/VLAN	IEEE 802.1Q		m
	Carrier Sense Multiple Access with Collision Detection (CSMA/CD)	ISO/IEC 8802-3:2001		m
Physical	Fibre optic transmission system 100Base-FX	ISO/IEC 8802-3:2001		c1
	Basic optical fibre connector NOTE This is the specification for the ST connector.	IEC 60874-10-1, IEC 60874-10-2 and IEC 60874-10-3		c1

c1 – Recommended, but future technology could be used.

Tabla 2-11. T-Profile para los servicios SV. (Fuente:[3])

7.2 Capa física

La especificación en este apartado es ligeramente distinta a la recogida en 6.1, ya que aunque aquí se muestra también preferencia por el uso de un enlace de fibra óptica 100BASE-FX, se deja abierta la posibilidad de uso de otro tipo de tecnologías, aunque sin determinar cuáles.

7.3 Capa de enlace

Siguiendo con la misma estructura que en el punto 6.2 se describen a continuación las especificaciones complementarias a las ya expuestas:

- En cuanto a la dirección de origen se indica de nuevo que una única dirección Ethernet debe ser usada para tal propósito. Respecto a la dirección de destino de la trama, se propone una dirección *multicast/unicast* configurable, en base al rango recomendado que se presenta en la Tabla 2-12.

Service	Recommended address range assignments	
	Starting address (hexadecimal)	Ending address (hexadecimal)
GOOSE	01-0C-CD-01-00-00	01-0C-CD-01-01-FF
GSSE	01-0C-CD-02-00-00	01-0C-CD-02-01-FF
Multicast sampled values	01-0C-CD-04-00-00	01-0C-CD-04-01-FF

Tabla 2-12. Rango de direcciones Multicast recomendadas. (Fuente:[3])

- Las especificaciones respecto a los campos de Prioridad y VLAN (véase la Figura 2-9) complementan a las expuestas anteriormente en los siguientes aspectos:

- El valor de CFI debe ser FALSE (por ejemplo, cero).
- Los valores de los campos VID y User priority se definen en la siguiente tabla:

Service	Default VID	Default priority
Sampled Values	0	4

Tabla 2-13. Valores de los campos VID y User priority de la etiqueta de prioridad de la trama. (Fuente: [3])

- La definición de la cabecera *Ethertype PDU* es más completa en este apartado de la norma que en el 9-1, como se puede ver en la Figura 2-16:

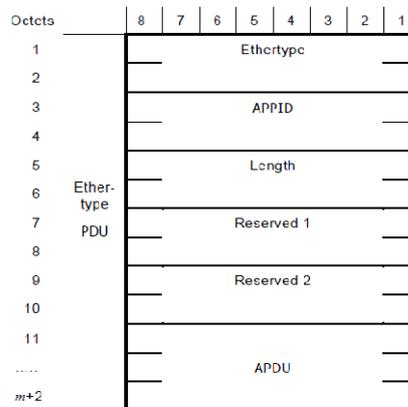


Figura 2-16. Estructura del Ether-type PDU. (Fuente:[3])

Asimismo, recoge las especificaciones complementarias que se incluyen a continuación:

- Valor del campo *Ether-type*. Según la Tabla 2-14.
- Valor del campo APPID. Combinación del tipo de APPID (Tabla 2-14), definido como los dos bits más significativos, y el valor ID, configurable y relacionado con los valores muestreados. Se tomará el valor por defecto 0x4000, que indica que no existe configuración del APPID.

Use	Ether-type value (hexadecimal)	APPID type
IEC 61850-8-1 GOOSE	88-B8	0 0
IEC 61850-8-1 GSE Management	88-B9	0 0
IEC 61850-9-1 Sampled Values	88-BA	0 1
IEC 61850-9-2 Sampled Values	88-BA	0 1

Tabla 2-14. Valores de la cabecera Ether-type PDU. (Fuente:[3])

- Valor del campo *Length*. Se define como el número de bytes incluyendo el *Ethertype PDU* y comenzando en el campo APPID; es decir, que tomando como referencia la Figura 2-16 el valor sería de *m*.

Conviene puntualizar que de las Tablas Tabla 2-12 y Tabla 2-14 sólo contiene información útil para este proyecto los campos correspondientes a los servicios referentes a SV.

7.4 Mapping del modelo de transmisión

7.4.1 Data-Set

Si bien el apartado 9-1 de la norma define dos tipos de Data-Set (*Universal* y *Status Indication*), el documento [3] puntualiza lo siguiente:

- El mapping especifica la transmisión de Data-Sets relacionados con el modelo de valores analógicos, en adición al tipo *Universal* definido en la 9-1.
- El tipo *Status Indication* no se contempla en este mapping, debido a que existen otros métodos de comunicación de los indicadores de estado binarios.
- Por lo tanto, los Data-Set empleados son configurables y se definen empleando el lenguaje XML, siguiendo las directrices de la norma IEC-61850-6. Esta definición se aloja en el LN LLN0, a través de la descripción de configuración del IED (*ICD*, *IED Configuration Description*), que es un fichero de alto nivel del tipo SCL (*Substation Configuration Language*) que recoge la información configurable del dispositivo.
- La definición del Data-Set debe ser coherente con el estándar BER del ASN.1 en cuanto a la notación de etiquetas y cabeceras. El empleo de este tipo de encapsulado se detallará posteriormente.

7.4.2 Bloques de control y servicios

Los bloques de control de valores muestreados, tal y como se describen en [6], deben estar o bien predefinidos o bien mapeados en un MSVCB (*Multicast Sampled Value Control Block*) o USVCB (*Unicast Sampled Value Control Block*) en MMS, que se alojará en el Nodo Lógico.

Las tablas que se muestran a continuación contienen la descripción correspondiente a la estructura MMS de los diferentes bloques de control y servicios mapeados.

MMS component name	MMS TypeDescription	r/w	m/o	Condition	Comments
MsvCBNam	ObjectName	r	m		MMS object name: the value of this component shall be of the format of ObjectReference and shall be limited to VMD or Domain scoped NamedVariableLists.
MsvCBRef	ObjectReference	r	m		MMS object name: the value of this component shall be of the format of ObjectReference and shall be limited to VMD or domain scoped NamedVariableLists.
SvEna	Boolean	r/w	m		TRUE = transmission of sampled value buffer is activated. FALSE = transmission of sampled value buffer is deactivated.
MsvID	Visible-string	r	m		System wide unique identification.
DatSet	ObjectReference	r	m		MMS object name: the value of this component shall be of the format of ObjectReference and shall be limited to VMD or Domain scoped NamedVariableLists.
ConfRev	Integer	r	m		Count of configuration changes regard to MSVCB.
SmpRate	Integer	r	m		Amount of samples per period.
OptFlds					
refresh-time	Boolean				TRUE = SV buffer contains the attribute "RefrTm" FALSE = attribute "RefrTm" is not available in the SV buffer.
sample-synchronised	Boolean	r	m		TRUE = SV buffer contains the attribute "SmpSynch". FALSE = attribute "SmpSynch" is not available in the SV buffer.
sample-rate	Boolean	r	m		TRUE = SV buffer contains the attribute "SmpRate". FALSE = attribute "SmpRate" is not available in the SV buffer.

Tabla 2-15. Definición MMS de la estructura del MSVCB. (Fuente: [3])

Services of MSVCB Class	Service
SendMSVMessage	Transmission of MSV messages is mapped directly on data link layer as defined in 8.4 and 8.5
GetMSVCBValue	Mapped to MMS read service
SetMSVCBValue	Mapped to MMS write service

Tabla 2-16. Mapeo de los servicios multicast. (Fuente:[3])

MMS component name	MMS type description	r/w	m/o	Condition	Comments
UsvCBNam	ObjectName	r	m		MMS object name: the value of this component shall be of the format of ObjectReference and shall be limited to VMD or Domain scoped NamedVariableLists.
UsvCBRef	ObjectReference	r	m		MMS object name: the value of this component shall be of the format of ObjectReference and shall be limited to VMD or Domain scoped NamedVariableLists.
SvEna	Boolean	r/w	m		TRUE = transmission of sampled value buffer is activated. FALSE = transmission of sampled value buffer is deactivated.
Resv	Boolean	r/w	m		TRUE = USVCB is exclusively reserved for the client that has set this value to TRUE.
UsvID	Visible-string	r	m		System-wide unique identification.
DatSet	ObjectReference	r	m		MMS object name: the value of this component shall be of the format of ObjectReference and shall be limited to VMD or domain scoped NamedVariableLists.
ConfRev	Integer	r	m		Count of configuration changes regard to USVCB.
SmpRate	Integer	r	m		Amount of samples per nominal periods.
OptFlds					
refresh-time	Boolean				TRUE = SV buffer contains the attribute "RefrTm" FALSE = attribute "RefrTm" is not available in the SV buffer.
sample-synchronised	Boolean	r	m		TRUE = SV buffer contains the attribute "SmpSynch". FALSE = attribute "SmpSynch" is not available in the SV buffer.
sample-rate	Boolean	r	m		TRUE = SV buffer contains the attribute "SmpRate". FALSE = attribute "SmpRate" is not available in the SV buffer.

Tabla 2-17. Definición MMS de la estructura del USVCB. (Fuente: [3])

Services of USVCB class	Service
SendUSVMessage	Transmission of USV messages is mapped directly on data link layer as defined in 8.4 and 8.5
GetUSVCBValue	Mapped to MMS read service
SetUSVCBValue	Mapped to MMS write service

Tabla 2-18. Mapping de los servicios unicast. (Fuente:[3])

Debido a que las tablas contienen la definición de cada campo, y a que la descripción XML se corresponde con una capa superior a la que se desarrolla en el proyecto, no se profundizará en la explicación de este apartado.

7.4.3 Capa de aplicación

Retomando la información presentada en la Tabla 2-5, se plantea el mapping de la trama directamente en el nivel de aplicación del *stack*.

De nuevo la norma en el apartado 9-2 encapsula la información en una APDU formada por varias ASDUs, tal y como se mostró en la Figura 2-14. Las especificaciones son las mismas que las explicadas en el punto 6.4.1, a las que se añade las siguientes restricciones:

- La concatenación de ASDUs dentro del APDU no puede variar de manera dinámica, sino que debe ser configurable únicamente antes del inicio de la transmisión.
- Dentro de una trama, el ASDU con los valores muestreados con mayor anterioridad debe situarse en primer lugar.

7.4.4 Capa de presentación

El mapping de los valores analógicos relacionado con la capa de presentación requiere el empleo de la gramática descrita en ASN.1-BER, lo que permite la codificación de los valores muestreados en una trama Ethernet. La Tabla 2-19 recoge los parámetros de esta codificación.

```
IEC61850 DEFINITIONS ::= BEGIN
IMPORTS Data FROM ISO-IEC-9506-2
IEC 61850-9-2 Specific Protocol ::= CHOICE {
  9-1-Pdu [0] IMPLICIT OCTET STRING, -- Reserved for 9-1 APDU
  savPdu [APPLICATION 0] IMPLICIT SavPdu,
```

Abstract Buffer Format according to IEC 61850-7-2		Coding in IEC 61850-9-2	Comments
Attribute name	Attribute type	ASN.1 Basic Encoding Rules (BER) SavPdu ::= SEQUENCE {	
		noASDU [0] IMPLICIT INTEGER (1..65535),	Number of ASDUs, which will be concatenated into one APDU.
Security		security [1] ANY OPTIONAL,	Reserved for digital signature.
		asdu [2] IMPLICIT SEQUENCE OF ASDU }	1 to <i>n</i> number of ASDUs as specified before.
		ASDU ::= SEQUENCE {	
MsvID or UsvID	VISIBLE STRING	svID [0] IMPLICIT VisibleString,	Should be a system-wide unique identification.
DatSet	ObjectReference	datset [1] IMPLICIT VisibleString OPTIONAL,	Value from the MSVCB or USVCB
SmpCnt	INT16U	smpCnt [2] IMPLICIT OCTET STRING (SIZE(2)),	Will be incremented each time a new sampling value is taken. The counter shall be set to zero if the sampling is synchronised by clock signal (SmpSynch = TRUE) and the synchronising signal occurs. See NOTE 2. The OCTET STRING is interpreted as INT16U as defined in Table 14
ConfRev	INT32U	confRev [3] IMPLICIT OCTET STRING (SIZE(4)),	Value from the MSVCB or USVCB. The OCTET STRING is interpreted as INT32U as defined in Table 14
RefrTm	EntryTime	refrTm [4] IMPLICIT UtcTime OPTIONAL,	RefrTm contains the refresh time of the SV buffer.
SmpSynch	BOOLEAN	smpSynch [5] IMPLICIT BOOLEAN DEFAULT FALSE,	TRUE = SV are synchronised by a clock signal. FALSE = SV are not synchronised.
SmpRate	INT16U	smpRate [6] IMPLICIT OCTET STRING (SIZE(2)),	Value from the MSVCB or USVCB. The OCTET STRING is interpreted as INT16U as defined in Table 14
Sample [1..n]	Type depends on the CDC defined in IEC 61850-7-3.	sample [7] IMPLICIT SEQUENCE OF Data }	List of data values related to the data set definition. See NOTE 1.
NOTE 1 For the encoding of the Data, the rules for the encoding of the basic data types shall apply as defined in Table 14.			
NOTE 2 When sync pulses are used to synchronise merging units, the counter shall be set to zero with every sync pulse. The value 0 shall be given to the data set where the sampling of the primary current coincides with the sync pulse.			
NOTE 3 The usage of the OptFlds attribute according to IEC 61850-7-2 is not necessary, because the relating attributes RefrTm, SmpSynch and SmpRate will be signed as optional via the ASN.1 attribute directly.			

```
... }
END
```

Tabla 2-19. Codificación del buffer de transmisión de SV. (Fuente:[3])

Encontramos en esta tabla la definición de los campos que deben ir alojados en la trama física y que por tanto resultan de especial interés para el desarrollo del proyecto. Debido a ello, merece la pena realizar una serie de puntualizaciones que clarifiquen las características mostradas:

- La etiqueta inicial de la trama se ajusta a las especificaciones ASN.1, pudiendo ser de dos tipos:
 - 9-1 Pdu. Reservado para el Universal Data-Set, definido en el apartado anterior de la norma. En este caso el valor reservado para este campo es 0x80.
 - savPdu. Correspondiente al APDU que contenga valores analógicos.
- La codificación de cada atributo se define tanto en la segunda columna de la tabla (*Attribute type*) como en la cuarta columna que contiene comentarios sobre la definición del campo y de su tipo. La Tabla 2-20, mostrada a continuación, recoge la descripción física de cada tipo de datos.

Data types according to IEC 61850-7-2	Encoding in data set	Comments
BOOLEAN	8 Bit set to 0 FALSE; anything else = TRUE	
INT8	8 Bit Big Endian	signed
INT16	16 Bit Big Endian	signed
INT32	32 Bit Big Endian	signed
INT128	128 Bit Big Endian	signed
INT8U	8 Bit Big Endian	unsigned
INT16U	16 Bit Big Endian	unsigned
INT24U	24 Bit Big Endian	unsigned
INT32U	32 Bit Big Endian	unsigned
FLOAT32	32 Bit IEEE Floating Point (IEEE 754)	
FLOAT64	64 Bit IEEE Floating Point (IEEE 754)	
ENUMERATED	32 Bit Big Endian	
CODED ENUM	32 Bit Big Endian	
OCTET STRING	20 Bytes ASCII Text, Null terminated	
VISIBLE STRING	35 Bytes ASCII Text, Null terminated	
UNICODE STRING	20 Bytes ASCII Text, Null terminated	
ObjectName	20 Bytes ASCII Text, Null terminated	
ObjectReference	20 Bytes ASCII Text, Null terminated	
TimeStamp	64 Bit Timestamp as defined in IEC 61850-8-1	
EntryTime	48 Bit Timestamp as defined in IEC 61850-8-1	
Data types according to IEC 61850-8-1	Encoding in data set	Comments
BITSTRING	32 Bit Big Endian	

Tabla 2-20. Codificación de los tipos básicos de datos. (Fuente:[3])

- Respecto a la información presente en la tercera columna de la Tabla 2-19, conviene realizar una aclaración necesaria para comprender el ejemplo que se presentará en el punto siguiente de este apartado: Los números que se encuentran entre corchetes en cada uno de los atributos serán de utilidad para la asignación de la etiqueta de cada campo según la sintaxis ASN.1.
- Para reducir la complejidad de la implementación, los atributos que se definen como opcionales (*OPTIONAL*), que son *Security*, *DatSet* y *RefrTm* no se mapearán en la trama física que se empleará en este proyecto, aunque teóricamente es el bloque de control del servicio el que debe indicar si los atributos aparecen o no en el mensaje SV (véanse Tablas Tabla 2-15 y Tabla 2-17).

A lo largo de este apartado se ha presentado la información necesaria para realizar el mapping del modelo de información para la transmisión de valores analógicos. A continuación, se presenta un ejemplo de trama APDU que servirá además para explicar la sintaxis ASN.1 empleada a la hora de configurar una trama física con formato IEC-61850.

7.5 Sintaxis ASN.1 y ejemplo de aplicación

7.5.1 ASN.1 Basic Encoding Rules

La notación ASN.1 es un estándar recogido en la norma ISO/IEC 8825-1, y fue desarrollado como parte de la capa de presentación del *stack* de comunicación OSI.

Esta norma proporciona un nivel de abstracción similar a los lenguajes de alto nivel, y permite que dispositivos de distintos fabricantes o con representaciones de datos diferentes puedan disponer de una sintaxis de representación estándar.

ASN.1 define varias clases de datos, llamadas **tipos**. De entre ellos, son dos los que resultan interesantes a la hora de construir el encapsulado de la trama IEC-61850, y se definen a continuación:

- Tipos primitivos (*Primitive*). Representan las clases de datos básicas. Ejemplos de este tipo son INTEGER, BOOLEAN, OCTECT STRING, que se corresponden con algunos de los definidos en la Tabla 2-20.
- Tipos constructores o estructurados (*Constructed*). Esta clase de datos permite la formación de una tabla o lista a partir de los tipos primitivos. A modo de ejemplo, la Tabla 2-19 presenta en la columna correspondiente a la codificación según el apartado 9-2 los tipos SEQUENCE y SEQUENCE OF, ambos constructores que como se puede apreciar según la notación engloban una serie de tipos básicos.

Una vez definidos los tipos de datos que en este caso alberga la trama física, se ilustra en la Figura 2-17 la estructura que debe tener un mensaje encapsulado según este estándar de interpretación de datos.

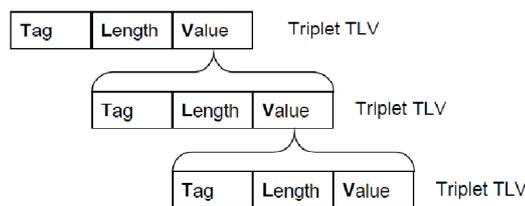


Figura 2-17. Formato BER para el encapsulamiento de la trama. (Fuente:[3])

Como se puede observar, las reglas de codificación BER definen un formato con tres componentes llamado TLV (*Tag, Length, Value*). Cada uno de estos campos está formado por series de octetos, y cumplen las especificaciones siguientes:

- Campo Etiqueta (*Tag*). Identifica mediante codificación cuál es el tipo de estructura que sigue. Está formado por un octeto con tres campos que se ajusta a lo descrito en la siguiente Figura:

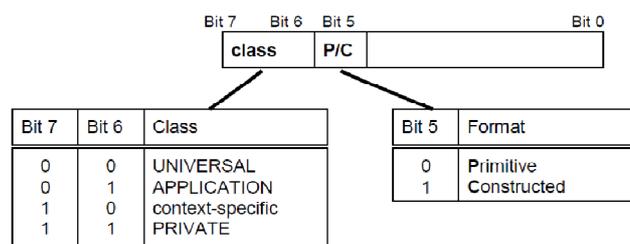


Figura 2-18. Formato del byte Tag. (Fuente:[3])

El único campo que requiere una explicación ulterior es el que se corresponde con los bits 4 al 0, que se rellenará con los números entre corchetes que aparecen en la Tabla 2-19, a cuya finalidad ya se hizo referencia en el punto 7.4.4.

- Campo Longitud (*Length*). Sirve para determinar el número de bytes que contendrá el siguiente campo. Puede estructurarse según dos formatos, como se puede observar en la Figura 2-19.

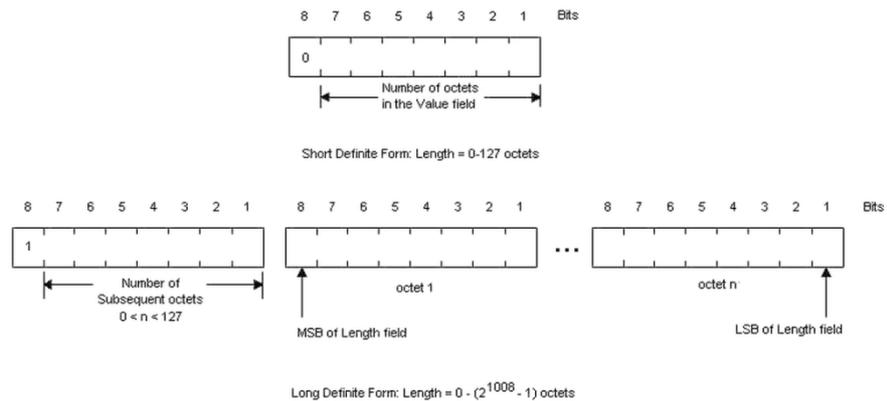


Figura 2-19. Formato de codificación del campo Length.

Por un lado, la **forma definida corta** ocupa un único byte, y representa una longitud de Valor de hasta 127 octetos, ya que para su identificación el bit 8 del campo longitud debe valer cero. Para la representación de datos que ocupen más de 127 bytes, se emplea la **forma definida larga**. El primer byte de esta estructura contiene el número de octetos que le siguen necesarios para representar toda la longitud del campo Valor, y está identificado por un uno en su bit más significativo. A partir del segundo octeto se encuentra el valor del campo Longitud, que puede representarse hasta con 127 bytes. *Esta estructura puede representar longitudes más cortas que los 128 bytes*, lo que será de utilidad para realizar la implementación física, ya que permite reservar tres bytes para el campo de longitud aunque el tamaño pueda ser variable y no restringido a 127 bytes.

- Campo Valor. Una vez identificado el tipo de dato y la longitud en bytes que ocupa, sólo queda agregar el valor representado a las cabeceras anteriores. Téngase en cuenta que este campo puede constituir una nueva estructura TLV.

Para completar la definición de la sintaxis, se ha de establecer el orden de envío de los datos. Así, ASN.1 presenta un formato *Big Endian*, de tal forma que cada byte transmite primero el bit más significativo (MSB), el cual se sitúa a la izquierda del octeto en la representación estructurada.

7.5.2 Ejemplo de APDU

El anexo A.2 del documento [3] presenta un ejemplo de APDU codificada según la sintaxis BER ASN.1, tal y como se muestra en la Figura 2-20.

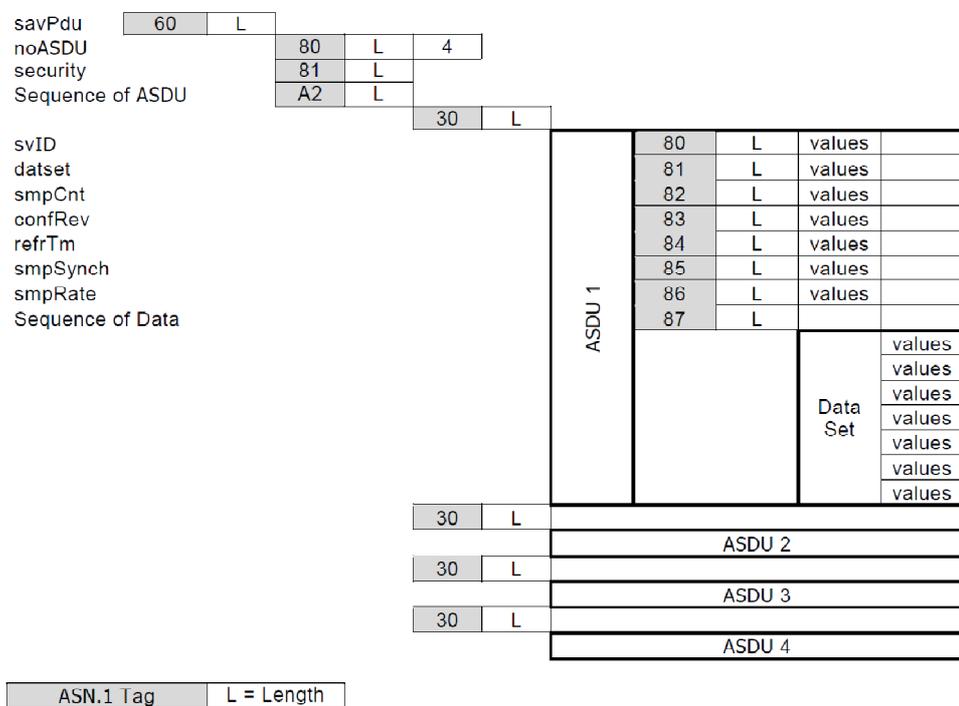


Figura 2-20. Ejemplo de APDU. (Fuente:[3])

La estructura jerárquica está claramente reflejada en el esquema, donde se aprecian cuatro ASDUs concatenadas que contienen los campos de información ya expuestos en la Tabla 2-19. Asimismo, saltan a la vista las cabeceras *Tag* y *Length* situadas en primer lugar de cada dato o estructura para organizar el encapsulado.

Sin embargo, el ejemplo expuesto en la IEC-61850-9-2 resulta algo incompleto, ya que ni define el Data-Set empleado ni supone una guía para los campos de longitud de las cabeceras.

Afortunadamente, la bibliografía consultada aporta los datos necesarios para una descripción completa de un mensaje SAV. El documento [7] (*Implementation guideline for digital interface to instrument transformers using IEC 61850-9-2*) incluye la Figura 2-21, mostrada a continuación.

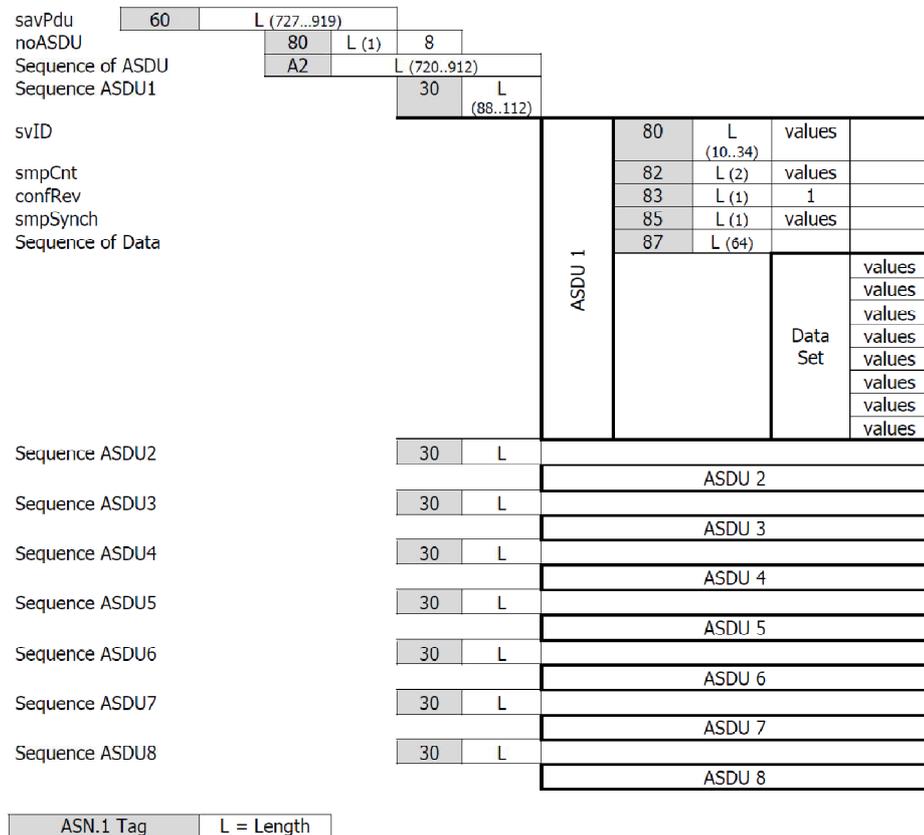


Figura 2-21. Estructura completa del APDU. (Fuente:[7])

En base a este ejemplo se puede realizar una explicación más precisa para clarificar el mapping utilizado para la implementación física de las tramas SAV.

En primer lugar, es necesario detallar cuál es el Data-Set que se envía en cada ASDU. Como se comentó en el punto 7.4.1 de este Capítulo, el Data-Set empleado en la transmisión es totalmente configurable siempre y cuando su descripción se detalle en el fichero SCL de configuración.

A este respecto, se encuentra en [7] la Figura 2-22 expuesta aquí.

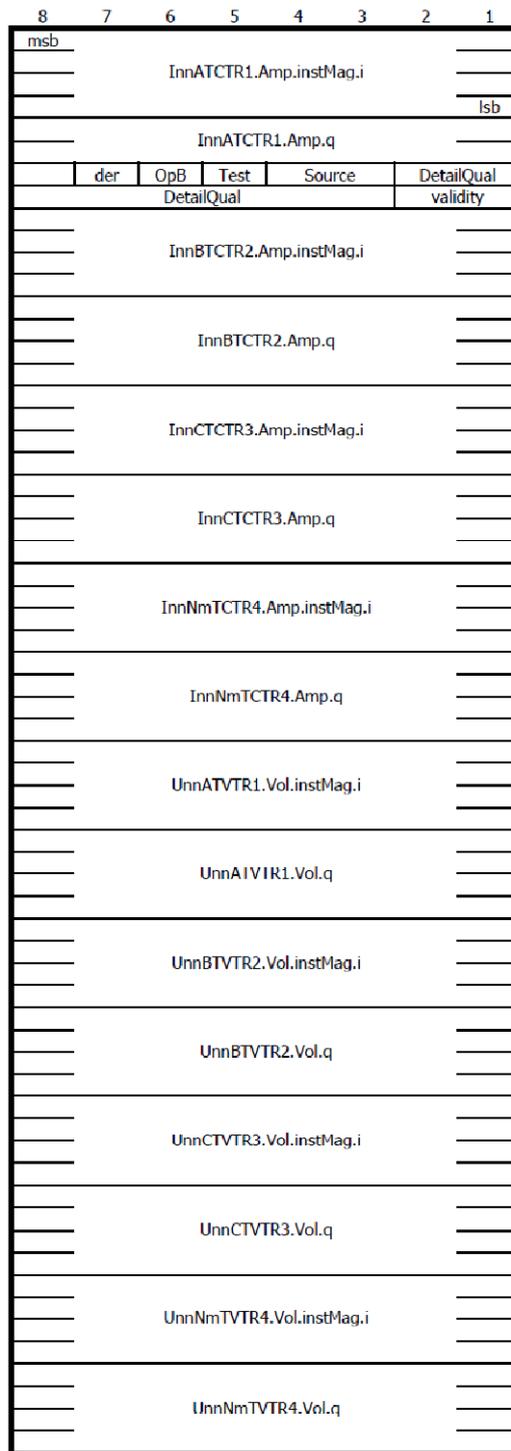


Figura 2-22. Codificación del Data-Set empleado. (Fuente:[7])

Tal y como se aprecia, el ASDU recoge los atributos Magnitud (*Mag*) y Calidad (*q*) de los datos Tensión (*Vol*) y Corriente (*Amp*) de los Nodos Lógicos correspondientes a los transformadores de tensión (TVTR) y de corriente (TCTR).

Compárese con los parámetros de los distintos atributos recogidos en la Tabla 2-8. La definición del atributo *Mag* indica que la codificación se hace en UI16 (entero de 16 bits sin signo), al igual que en la trama mostrada. De igual forma, los parámetros englobados como Calidad coinciden con los especificados en el apartado 9-1 de la norma, aunque se reservan 32 bits para el campo completo, quizá para simplificar la implementación a la hora de trabajar con tramas del mismo tamaño. Por lo tanto, se tienen varios ASDUs cada uno de ellos conteniendo un Data-Set de 64 bytes.

En aras de simplificación, se recoge en las siguientes tablas la información necesaria para la completa comprensión de los campos *Tag* y *Length*. Se recomienda comparar los contenidos de las Tablas Tabla 2-21, Tabla 2-22 y Tabla 2-19 con la Figura 2-21 para obtener una visión concreta del mapping realizado sobre el estándar.

Atributos	<i>Valor dec.</i>	<i>Valor Hex.</i>	<i>Comentario</i>
savPdu	919	0x82-03-97	Forma definida larga Incluye: -Misma longitud que <i>Sequence of ASDU</i> -7 bytes de longitud
noASDU	1	0x01	Forma definida corta
Sequence of ASDU	912	0x82-03-90	Forma definida larga Incluye: -8 ASDUs -48 bytes de longitud
Sequence ASDUn	112	0x70	Forma definida corta Incluye: -Campos inferiores -5 bytes de longitud
svID	34	0x22	Forma definida corta
smpCnt	2	0x02	Forma definida corta
confRev	1	0x01	Forma definida corta
smpSynch	1	0x01	Forma definida corta
Sequence of Data	64	0x40	Forma definida corta

Tabla 2-21. Desglose del campo Length de los atributos.

<i>Atributos</i>	<i>Valor Hex.</i>	<i>Class</i>	<i>P/C</i>	<i>[Número]</i>
savPdu	0x60	01 (APPLICATION)	1 (Constructed)	0
noASDU	0x80	10 (context-specific)	0 (Primitive)	0
Sequence of ASDU	0xA2	10 (context-specific)	1 (Constructed)	2
Sequence ASDUn	0x30	00 (UNIVERSAL)	1 (Constructed)	0
svID	0x80	10 (context-specific)	0 (Primitive)	0
smpCnt	0x82	10 (context-specific)	0 (Primitive)	2
confRev	0x83	10 (context-specific)	0 (Primitive)	3
smpSynch	0x85	10 (context-specific)	0 (Primitive)	5
Sequence of Data	0x87	10 (context-specific)	0 (Primitive)	7

Tabla 2-22. Desglose del campo Tag de los atributos.

8. Conclusiones

Este Capítulo ha servido para exponer el mapping que se implementará posteriormente a nivel físico. Los puntos 6 y 7 recogen toda la información necesaria para construir las tramas IEC-61850 que serán enviadas vía Ethernet.

Por último, se muestra la Figura 2-23, que contiene un modelo de la transmisión que será implementada.

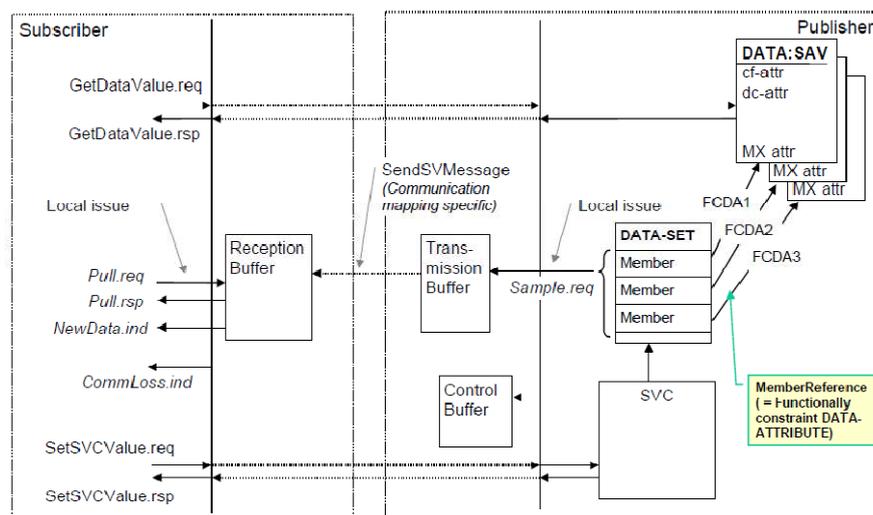


Figura 2-23. Modelo para la transmisión de sampled values (Fuente:[6])

Según esta estructura, el servicio *SendSVMessage* requiere el empleo de un buffer de transmisión que recoge los datos de un Data-Set, cuyo funcionamiento es regulado por un bloque de control.

Capítulo 3. IEEE 802.3: ETHERNET

Como indican las especificaciones de este proyecto, el enlace físico para la transmisión de datos debe estar soportado por el protocolo de comunicación Ethernet.

Tras un apartado introductorio, el apartado 2 presenta las distintas posibilidades para la configuración física del enlace. Seguidamente, los apartados 3 y 4 detallan la descripción del encapsulado de la información y los valores correspondientes a cada campo de la trama según el estándar. Por último, en el apartado 5 se presenta un ejemplo obtenido de la bibliografía que permite tener una visión clara de cuál es la estructura de datos que se desea aplicar.

1. Introducción

Como se especificó en el Capítulo 2, la norma IEC-61850 impone para el mapping del modelo de comunicación el empleo del protocolo Ethernet en las capas física y de enlace definidas según el sistema de referencia de *stack* OSI.

Ethernet es un estándar de comunicación ampliamente extendido que fue desarrollado por el comité de estándares IEEE 802.3 para redes LAN (*Local Access Network*).

2. Capa física

Se ha explicado que una de las especificaciones de la norma para realizar el envío de mensajes era la configuración física necesaria para la transmisión.

A este respecto, el estándar Ethernet ofrece un gran abanico de opciones, cuya normalización es lo suficientemente abierta como para permitir la incorporación de desarrollos futuros.

De esta forma, se define una notación concisa para distinguir las diferentes implementaciones posibles:

<velocidad de transmisión (Mbps)><método de señalización><longitud máxima del segmento (hm)>

Según esta nomenclatura, se presentan varios formatos de conexión física, algunas de las cuales se muestran en la siguiente tabla obtenida de la información de [9] junto con sus características.

	<i>10BASE5</i>	<i>10BASE-T</i>	<i>100BASE-TX</i>	<i>100BASE-FX</i>
Medio de transmisión	Coaxial	Par trenzado	2 pares	2 fibra óptica
Técnica de señalización	Banda Base (Manchester)	Banda Base (Manchester)	MLT-3	4B5B NRZI
Long. máxima de segmento	500	100	100	100

Tabla 3-1. Alternativas para el medio de transmisión

Obsérvese que BASE-T y BASE-F no siguen la notación anterior: T se emplea para expresar el uso de par trenzado y F para la fibra óptica.

A la vista de la Tabla 3-1 se puede deducir la existencia de multitud de alternativas tanto para *Ethernet* (10Mbps) como para *Fast Ethernet* (100Mbps), ambas soportadas por el estándar de envío de datos de la norma 61850, como se verá en los puntos 6.1 y 7.2 del Capítulo 2.

Existen también estándares definidos para el Gigabit Ethernet o incluso el Ethernet de 10Gbps.

¿Cuál es la utilidad de conexiones tan rápidas? Aunque a nivel de usuario una conexión de 10Gbps puede parecer exagerada, lo cierto es que la necesidad de enlaces cada vez más veloces surge, por una parte, debido a la constante migración de usuarios a conexiones más altas (100Mbps a 1Gb, 10Mbps a 100Mbps...) y al aumento de aplicaciones demandantes de ancho de banda, como la reproducción de vídeo de alta calidad. La concentración geográfica de conexiones es otro de los motivos del aumento progresivo de la velocidad, debido a que de esta manera se permite la construcción de redes de área metropolitana (MAN, *Metropolitan Area Network*) o redes de área amplia (WAN, *Wide Area Network*) para establecer un entramado que comunique redes LAN dispersas entre distintos puntos de presencia.

3. Trama MAC

Como ya se recogió en el punto 6.2 del Capítulo anterior, el empleo de Ethernet como capa de enlace del *stack* de comunicación para el envío de datos con formato IEC-61850 únicamente exige la transmisión unidireccional, por lo que en la implementación del proyecto se optó por diseñar un módulo sencillo que simplemente enviara datos a un dispositivo de recepción Ethernet, como puede ser el caso de un PC.

Esta especificación implica el encapsulado de los datos SAV en paquetes reconocibles para un receptor Ethernet. El documento [8] ofrece la estructura estandarizada necesaria para conseguirlo, que no es otra que la trama MAC, que se muestra en la Figura 3-1.

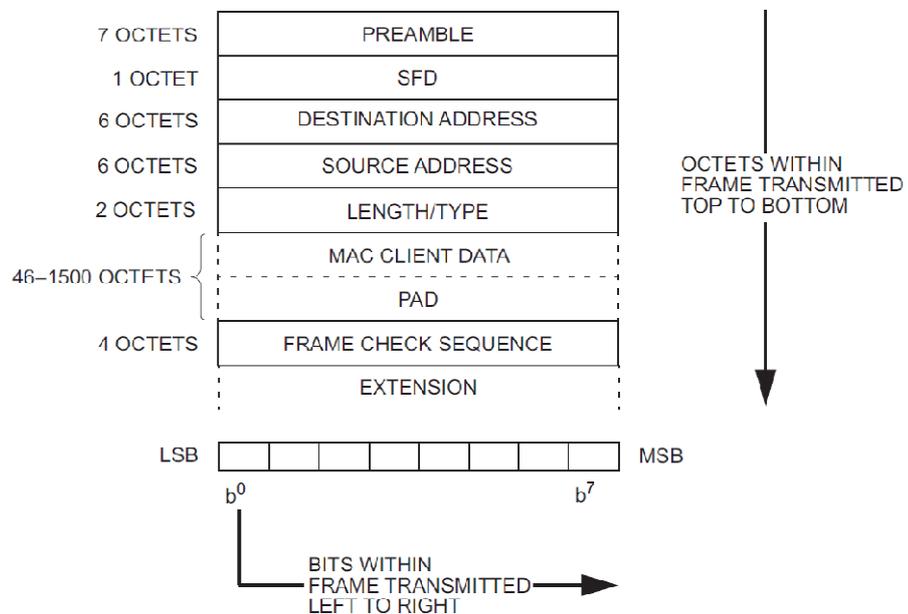


Figura 3-1. Formato de la trama MAC. (Fuente:[8])

La Figura recoge cada uno de los campos necesarios para una correcta transmisión de datos así como el tamaño en bytes de cada uno de ellos. A continuación se realiza una descripción según el orden en el que deben ser enviados:

- Preámbulo (*Preamble*). Sirve para establecer la sincronización inicial entre el emisor y el receptor. Su valor es de 7 bytes de bits ceros y unos alternados: 10101010.

- Delimitador del comienzo de la trama (SFD, *Start Frame Delimiter*). Indica al receptor el comienzo real de la trama, y posibilita que se pueda localizar el primer bit del resto de la información. Consiste en la secuencia de bits 10101011.
- Dirección de destino (*Destination Address*). Especifica la estación o estaciones a las que va dirigida la trama. Puede tratarse de una única dirección física, una dirección de grupo o una dirección global.
- Dirección de origen (*Source Address*): especifica la estación que envió la trama.

Los requerimientos de la IEC-61850 referentes a los campos de dirección se han analizado en el punto 7.3 de esta memoria.

- Longitud/Tipo (*Length/Type*). Este campo puede tener dos significados: o bien contiene la longitud del campo de datos que serán enviados expresado en octetos, o bien representa al campo Tipo de Ethernet, dependiendo de que la trama siga la norma IEEE 802.3 o la especificación primitiva de Ethernet. En cualquier caso, el tamaño máximo de la trama, excluyendo los campos Preámbulo y SFD es de 1518 octetos.

De nuevo aparece aquí una restricción en cuanto al envío de los datos, aunque en este caso es mucho más fuerte que la comentada en 6.4.1, ya que limita el número de Data-Set que se pueden encapsular en cada APDU. Por ejemplo, para un Data-Set de 64 bytes, el número máximo aproximado de ASDUs que se puede transmitir por cada trama Ethernet sería de 20, teniendo en cuenta que es necesaria la adición de las diferentes cabeceras como se analizó en el ejemplo de [7] recogido al final del Capítulo anterior.

Como se especificó en el punto 7.3 del Capítulo anterior, la norma IEC-61850 indica que la interpretación debe ser de Tipo, e indica cuál es el valor que debe contener el campo en función del mensaje (Tabla 2-14).

- Datos MAC (MAC *client data*). Este campo se corresponde con los datos que se quieren enviar. En lo concerniente al desarrollo de este proyecto, se trata de una estructura APDU completa.
- Relleno (*Pad*). El estándar 802.3 especifica que el tamaño mínimo del campo Datos debe ser de 46 octetos para un correcto funcionamiento de la técnica de detección de colisiones (CD). Para que esto se cumpla en cualquier circunstancia y la trama sea lo suficientemente larga se añaden bytes de relleno en este campo.
- Secuencia de comprobación de trama (FCS, *Frame Check Sequence*). Uno de los mecanismos que hace de Ethernet uno de los protocolos de comunicación más empleados es un eficiente sistema de detección de colisiones mediante un algoritmo de comprobación de redundancia cíclica (CRC, *Cyclic Redundancy Check*). En este caso, el FCS contiene el valor del CRC de 32 bits calculado sobre la trama enviada.

Este mecanismo se codifica según un polinomio matemático que cumple la siguiente fórmula:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

El procedimiento para el cálculo del CRC de una trama dada se realiza según las siguientes pautas:

- Los primeros 32 bits de la trama se complementan.
- Los bits n de la trama son interpretados como coeficientes de un polinomio $M(x)$ de grado $n-1$, de tal forma que el primer bit del campo *Destination Address* se corresponde con el término x^{n-1} y el último bit del campo Datos con el término x^0 .
- $M(x)$ se multiplica entonces por x^{32} y se divide por $G(x)$, produciendo un residuo $R(x)$ de grado menor o igual a 31.
- Los coeficientes de $R(x)$ se consideran entonces una secuencia de 32 bits cuyo complemento da como resultado el valor del CRC.

Los 32 bits del CRC se sitúan en el campo FCS de modo que el término x^{31} se corresponde con el LSB y el término x^0 con el MSB; es decir, que el orden de transmisión es $x^{31}, x^{30}, \dots, x^1, x^0$.

Durante este proceso se tienen en cuenta todos los campos de la trama Ethernet excepto el preámbulo, el SFD y el FCS.

Este cálculo es realizado tanto por el emisor como por el receptor de la trama, de forma que si este campo no coincide en ambos encapsulados, los datos se consideran corruptos y por tanto no se tienen en cuenta.

El algoritmo para la obtención del CRC es fácil de conseguir debido a la gran extensión de uso del protocolo IEEE 802.3, y a través de Internet se puede acceder a un vasto abanico de opciones para su implantación, como se verá más adelante en esta memoria.

- Extensión (*Extension*). [8] sólo recoge el uso de este campo en el caso de que el tamaño de la trama sea inferior al valor de un parámetro definido como *slotTime* según el estándar. Esta circunstancia sólo se produce cuando se trabaja sobre una conexión de 1Gbps, ya que en este caso el *slotTime* tiene un valor de 4096 *bit times* mientras que el tamaño mínimo de la trama es para cualquier velocidad de 64 bytes (512 bits). La Figura 3-2 esquematiza el empleo de este campo.

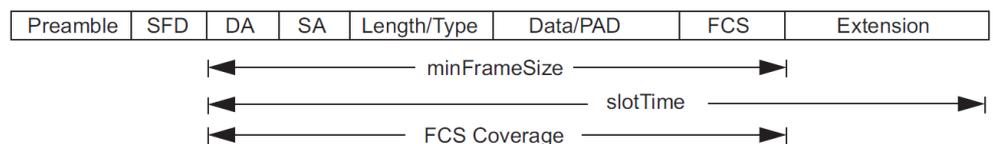


Figura 3-2. Esquema de una trama con campo *Extension*. (Fuente:[8])

En este punto se ha realizado una descripción de todos los campos necesarios para la composición de una trama reconocible bajo el formato Ethernet.

Sin embargo, el mismo estándar estudiado presenta una extensión a esta trama para la incorporación de etiquetas que aporten más información para la transmisión, el cual se trata a continuación.

4. Trama Tagged MAC

Como se puede extraer de la información contenida en la Tabla 2-5, las especificaciones de la norma IEC-61850 para el envío de valores analógicos muestreados exigen el empleo en la capa de enlace de etiquetas de prioridad y de identificación de LAN virtual (*Priority tagging/VLAN*) según indica el protocolo IEEE 802.1Q.

Es precisamente la trama *Tagged MAC* la que permite la adición de este tipo de información en un campo añadido a la estructura básica, cuya composición se puede apreciar en la Figura 3-3.

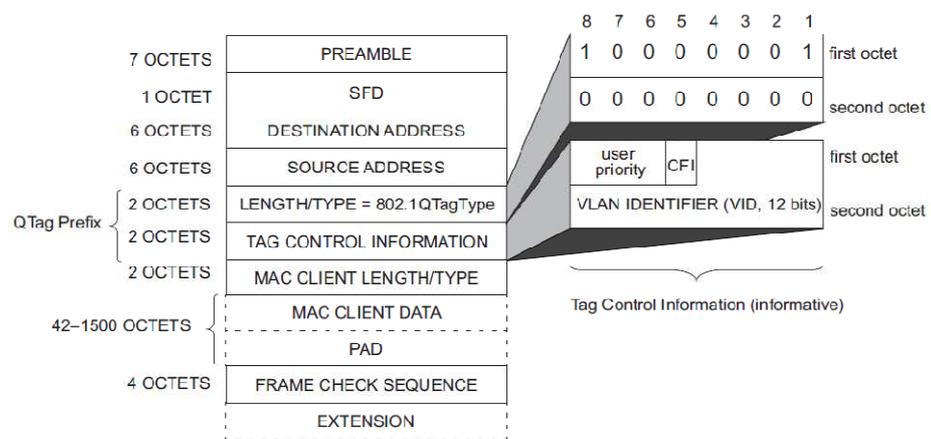


Figura 3-3. Formato de la trama Tagged-MAC. (Fuente:[8])

Se observa la adición de una etiqueta de 4 bytes con formato 802.1Q (*QTag Prefix*) que se sitúa entre el campo *Source Address* y la indicación de longitud/tipo, aportando la siguiente información:

- En primer lugar se sitúa un campo de Longitud/Tipo similar al presente en la trama básica. Según [8] en esta ocasión se aplica siempre la interpretación de Tipo, y contiene la constante 0x81-00 reservada para una estructura *Tag Protocol Type*.
- El segundo campo se define como de información de control de las etiquetas (*Tag Control Information*). Los tres apartados de los que consta se diferencian en la Figura, y su sintaxis está recogida en el estándar IEEE P802.1Q. Para el desarrollo de este proyecto no es necesario un análisis de la misma, ya que como se estudió previamente, los apartados 9-1 y 9-2 de la IEC-61850 definen los valores que debe albergar cada campo.

La inclusión de estos cuatro bytes al comienzo de la trama no modifica ni la sintaxis ni el cálculo de los campos *Length/Type* ni FCS.

Este Capítulo resume las especificaciones generales del estándar de comunicación Ethernet recogidas en el documento [8]. Sólo resta por tanto conocer cómo queda encapsulada la trama SAV descrita en el Capítulo 2 dentro de una estructura 802.3 que sea reconocible por un receptor físico, para lo que se reserva el punto siguiente.

5. Encapsulado de trama SAV según protocolo Ethernet

Al igual que se hizo al final del anterior Capítulo, se recurre a un ejemplo para facilitar la comprensión de la información descrita antes de pasar a la exposición de la arquitectura implementada.

De nuevo encontramos en los anexos de [1] y [3] un patrón a tal efecto que describe una trama Ethernet completa destinada al envío de valores analógicos muestreados dentro del modelo de un SAS. No obstante, como ocurría a la hora de exponer el formato de un APDU, la referencia [7] ofrece un ejemplo muy similar pero con mayor cantidad de información, el cual se presenta en la Figura 3-4.

Debido a la claridad de la imagen presentada, no resulta necesaria una explicación detallada, ya que aún los contenidos clave expuestos en los dos últimos capítulos. Únicamente a modo de guía, se suministra la Tabla 3-2, que recoge la fuente de información necesaria para rellenar cada uno de los campos cuyo valor aparece en la Figura.

<i>Campo</i>	<i>Valor Hex.</i>	<i>Fuente</i>
Destination Address	0x01-0C-CD-04-XX-XX	Tabla 2-12
TPID	0x81-00	Figura 2-9, Tabla 2-13
User priority	0x4	Figura 2-9, Tabla 2-13
CFI	0x0	Figura 2-9, Tabla 2-13
VID	0x0-00	Figura 2-9, Tabla 2-13
Ethertype	0x88-BA	Tabla 2-14
APPID	0x40-00	Tabla 2-14
Reserved1	0x00-00	Figura 2-10
Reserved 2	0x00-00	Figura 2-10

Tabla 3-2. Desglose de la información de la trama.

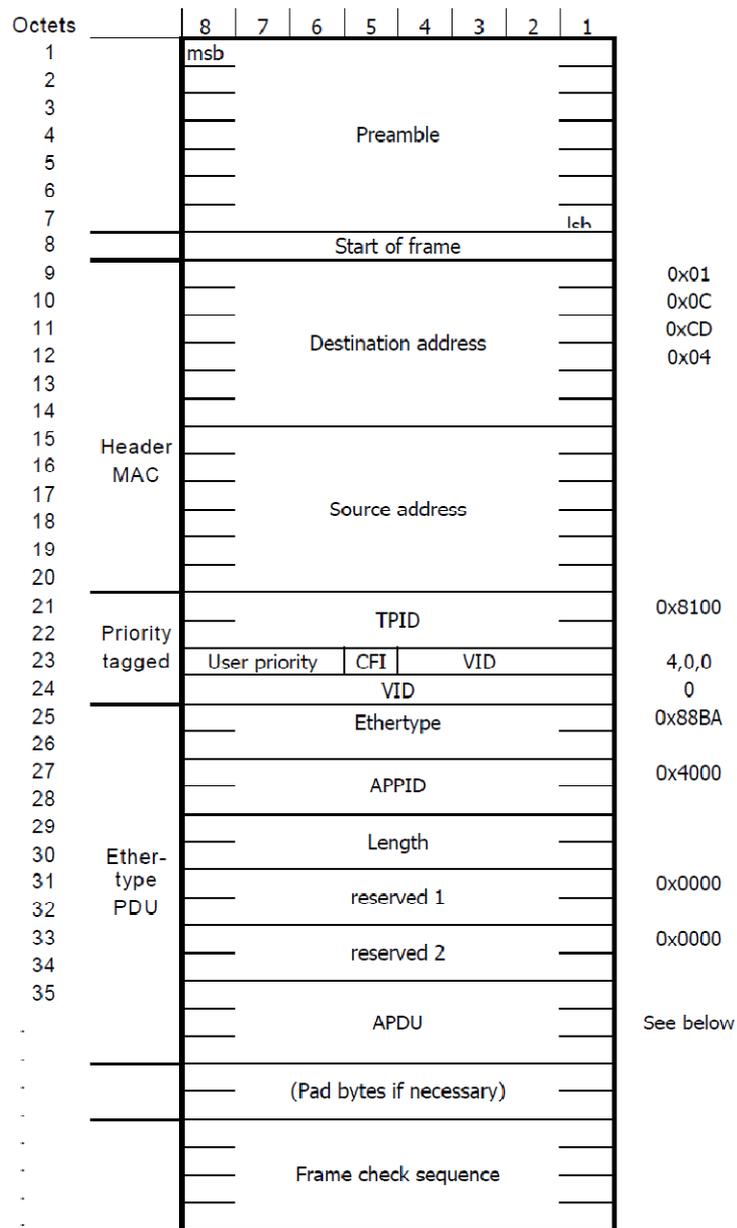


Figura 3-4. Trama SAV encapsulada bajo el estándar 802.3 (Fuente:[7])

Capítulo 4. ARQUITECTURA DESARROLLADA

Hasta el momento se han expuesto las especificaciones de la transmisión de datos extraídas de los diferentes documentos necesarios para la realización del proyecto.

Este Capítulo describe la estructura de la solución hardware desarrollada según dichas especificaciones e implementada en la tarjeta de desarrollo disponible. El primer apartado aporta una visión general del funcionamiento del flujo de datos diseñado, mientras que los apartados siguientes detallan las características de cada bloque y su utilidad dentro del esquema global.

1. Esquema general de funcionamiento

La arquitectura hardware desarrollada ha sido diseñada en base al flujo de datos exigido tanto por la IEC-61580 como por el protocolo Ethernet. En la Figura 4-1 puede encontrarse un diagrama de bloques que esquematiza este diseño.

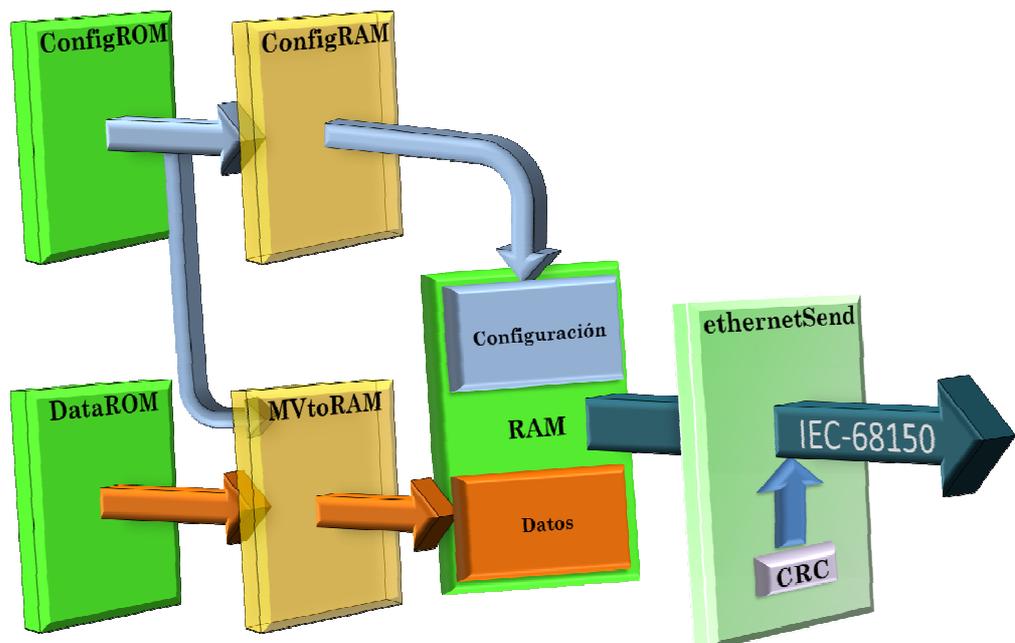


Figura 4-1. Estructura de la solución implementada.

De la observación del diagrama presentado se puede deducir que las flechas indican el flujo de datos a través de los distintos módulos, representados por bloques de diferentes colores. La denominación de cada uno de ellos se corresponde con la del fichero VHDL (*.vhd*) al que hacen referencia dentro del proyecto.

Como se ha mencionado previamente, el módulo implementado responde a las especificaciones de encapsulado de la información relativas al servicio de envío de valores muestreados vía Ethernet.

Una vez que las muestras son capturadas, cabe preguntarse cuáles son los pasos que han de seguirse para conseguir un empaquetado adecuado de los datos. La estructura organizativa del diseño responde a esta cuestión, ya que presenta una división en bloques que de manera secuencial construyen la trama requerida para el envío.

Uno de los objetivos fundamentales de este proyecto es conseguir una plataforma que permita su configuración, y es precisamente este proceso el que se realiza en primer lugar. De ello se encarga el módulo *ConfigRAM*, cuya misión es recoger la información contenida en la memoria de solo lectura *ConfigROM* y depositarla de manera ordenada dentro de la trama que se articula en el interior de la memoria *RAM*.

Este registro de los valores de configuración tanto del dispositivo como de los parámetros de envío se realiza una única vez. Seguidamente se activa el proceso de captura y encapsulado de los datos SAV, función que se reserva para el módulo *MVtoRAM*.

El cometido de este bloque es tomar los valores almacenados en *DataROM* y empaquetarlos añadiéndoles las cabeceras y etiquetas necesarias de manera que el resultado que se registra en la memoria *RAM* complete la trama Ethernet con el modelo de datos IEC-61850. Nótese que, tal y como se recoge en la Figura 4-1, parte de la información que se requiere para el encapsulado procede de la memoria *ConfigROM*.

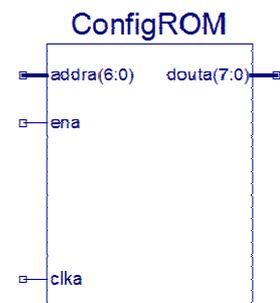
El último paso para completar la transmisión consiste en tomar los valores digitales y lanzarlos a través del enlace físico, de lo que se encarga el bloque *ethernetSend*.

Este módulo es transparente respecto a la información, ya que su misión consiste en realizar un barrido completo de la memoria *RAM* con el fin de enviar dentro de una única trama todos los datos almacenados en ella. Cabe señalar dentro de este elemento la existencia de una función dedicada al cálculo del valor del CRC para su posterior adición a la trama Ethernet.

Tras haber sido expuesto el funcionamiento general del diseño completo, se procederá ahora a detallar las características de cada uno de los módulos que intervienen en el proceso.

2. Módulo *ConfigROM*

Tal y como se expuso en el apartado anterior, la memoria de sólo lectura *ConfigROM* almacena los datos relativos a la configuración del dispositivo y del envío.



La única especificación reseñable de este sencillo bloque es el modo de acceso al mismo para la introducción de los datos, ya que las señales que intervienen en su gobierno son únicamente las de activación (*Enable, ena*), dirección de lectura (*addra*) y reloj (*clka*), con un bus de salida correspondiente a los datos devueltos por el módulo (*douta*).

El software *Xilinx ISE Design Suite* empleado en el proyecto contempla la opción de configurar las memorias ROM que se generan automáticamente desde el *CORE Generator* (véase Figura 4-2) volcando en ellas un fichero de texto de extensión *.coe*.

El tipo de información, como se ha comentado, puede ser de dos tipos:

- Referente a la trama Ethernet. En este caso incluye los campos correspondientes a dirección de destino, dirección fuente, y etiquetas del protocolo VLAN (TPID y TCI), así como el valor del Ethertype de datos SAV, el APPID, y los bytes reservados.

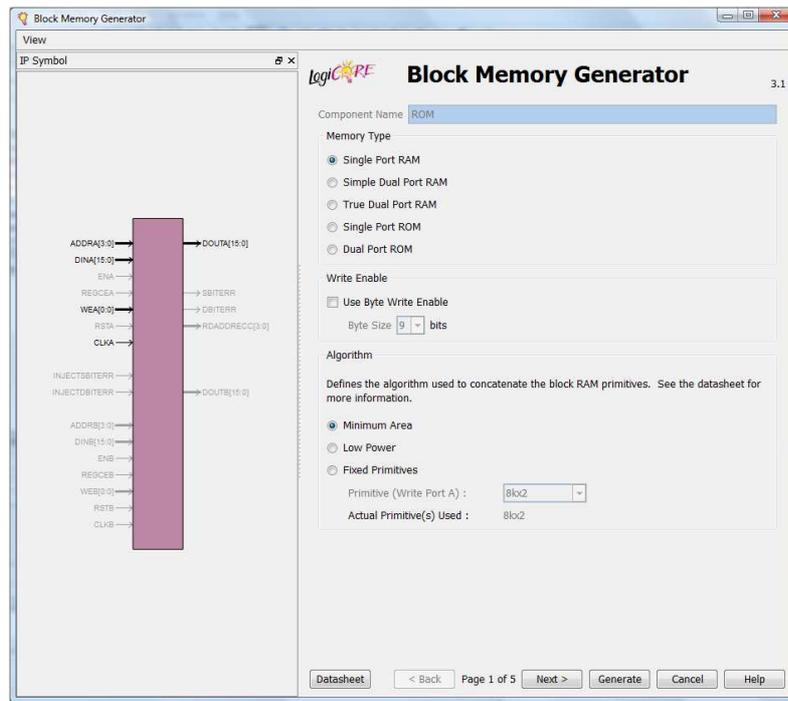


Figura 4-2. Captura de pantalla del CORE Generator.

- Referentes a los datos muestreados. VHDL es un instrumento de descripción lógica, por lo que carece de funciones que en el caso de lenguaje de alto nivel pueden parecer básicas, como es el caso del cálculo de la longitud de un vector. Por ello, en aras de simplificación, se requiere para la configuración del módulo una serie de datos tales como el tamaño de la trama que debe ser enviada (*DataSize*), el número de ASDUs (*n*) o la frecuencia de muestreo (*Sampling Rate*). Asimismo, se incluye el campo *svID*, que como se indicó en las Tabla 2-19 y Tabla 2-20 tiene una longitud de 35 bytes destinados a acoger una cadena de caracteres que identifique los valores medidos.

A continuación se muestra el fichero de texto mencionado necesario para la configuración inicial de la plataforma. Nótese que la sintaxis que ofrece el software permite la definición, en la primera línea del texto de la base en la que se expresan los valores consecuentes, así como la inclusión de comentarios precedidos de punto y coma.

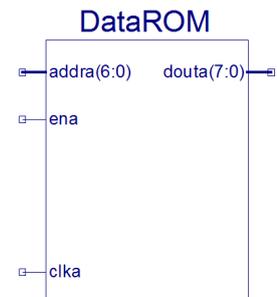
De esta forma, en el caso mostrado la base es 16, lo que impone una interpretación hexadecimal de los valores que contiene el archivo.

MEMORY_INITIALIZATION_RADIX=16;	33,
MEMORY_INITIALIZATION_VECTOR=	44,
; Destination address	55,
00,	66,
12,	77,
3F,	88,
8D,	99,
14,	aa,
E4,	11,
; Source address	22,
00,	33,
AA,	44,
00,	55,
62,	66,
c6,	77,
08,	88,
; TPID	99,
81,	aa,
00,	11,
; TCI	22,
80,	33,
00,	44,
; Ethertype	55,
88	66,
BA	77,
; APPID	88,
40,	99,
00,	aa,
; Sampling Rate	66,
01,	77,
90,	88,
; n	99,
05,	aa,
; Data Size	;Reserved 1
00,	00,
40,	00,
; svID	;Reserved 2
11,	00,
22,	00,

Antes de pasar a describir otro de los bloques del diseño, es necesario realizar una puntualización referente al tamaño de esta memoria. Evidentemente, tanto la longitud como el ancho de la ROM son configurables. Debido a razones que se detallarán en el apartado 7.3, en este caso se ha elegido un ancho de 8 bits con una longitud de 64 octetos para dar cabida a toda la información. El envío de tramas como las planteadas en el desarrollo de este proyecto no precisa la variación de ninguna de las dos dimensiones de esta memoria. Sin embargo, la inclusión de datos no obligatorios como los descritos en el punto 7.4.4 quizá precise un cambio de longitud.

3. Módulo *DataROM*

Las características generales de este bloque coinciden con las del anterior, por lo que no es precisa una nueva explicación, salvo la referente al tipo de datos que se almacenan en ella.



Ante la falta de una muestra de valores reales que se puedan emplear en este modelo de comunicación, se rellenó la memoria con datos con el único propósito de que fueran identificables dentro de una trama. Por ello, el vector de 64 bytes comienza y termina con dos octetos con la palabra 0xDA. Nótese la correspondencia entre esta longitud y el valor del campo *Data Size* de la ROM de configuración (0x0040=64 decimal).

En el caso concreto de esta implementación, se procuró trabajar con tramas de las que se tuviera algún tipo de referencia. Como ya se explicó en el punto 7.5.2 de este documento, es necesario recurrir a la bibliografía para encontrar un ejemplo lo suficientemente detallado para constituir una guía útil. De esta forma, el conjunto de datos almacenado en *DataROM* pretende imitar un Data-Set (o ASDU) como el presentado en la Figura 2-22.

Sin embargo, los atributos *Mag* y *q* de los datos muestreados requieren desde el punto de vista de la plataforma disponible entradas tanto analógicas como digitales de las que no se dispone. Por ello, el empleo del Data-Set del documento [7] como base al modelo de encapsulamiento de datos se reduce a la consideración de una trama de la misma dimensión.

Evidentemente, el tamaño de la memoria es configurable e igual al número de bytes que contiene el fichero que alberga los datos. Contrariamente a lo que sucedía con el módulo anterior, en este caso una posible variación en los requisitos de funcionamiento para una aplicación concreta con una definición de Data-Set particular requeriría una adaptación tanto de la longitud de la ROM como del archivo de texto *datos.coe*, el cual se muestra a continuación.

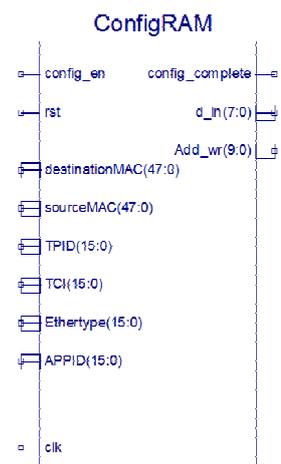


4. Módulo *ConfigRAM*

El siguiente bloque es el encargado de tomar la información referente a la trama Ethernet contenida en *configROM* y volcarla en la memoria RAM donde se configura la trama completa.

Este módulo presenta, como señales de gobierno, una señal de activación (*config_en*), un reset (*rst*) y un reloj de sincronización (*clk*).

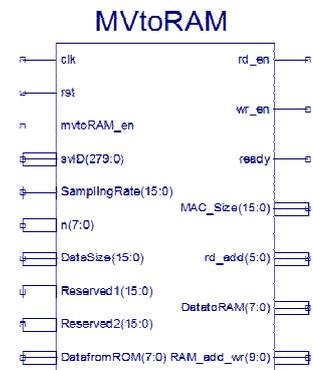
Asimismo, una señal de salida (*config_complete*) indica que ya se ha terminado la configuración y por tanto se puede comenzar la captura de datos.



Respecto al flujo de la información gestionada por el bloque, éste multiplexa los datos de entrada correspondientes a cada uno de los campos almacenados en la memoria ROM de configuración, y los vuelca en la RAM en función de la dirección que ocupen en la misma.

5. Módulo *MVtoRAM*

Tal y como se muestra en el esquema de la Figura 4-1, la misión de este módulo es realizar la extracción de los valores muestreados disponibles para su encapsulado en base a la información de configuración del envío que se almacena en *ConfigROM*.



De esta forma, aparte de las señales de activación, reset, reloj e indicación de tarea realizada comunes a todos los bloques lógicos, éste presenta dos tipos de entradas.

Por una parte, las correspondientes a la configuración de la trama IEC-61850, con campos como el número de muestras enviadas en cada APDU (*n*), la identificación (*svID*) o el tamaño en bytes del Data-Set.

Por otra, presenta un bus de 8 bits que le permite obtener los datos de la memoria ROM a la que accede mediante una señal de salida de dirección (*rd_add*) y un enable de lectura (*rd_en*). La captura de estos datos se realiza como si de valores analógicos actualizados se tratara, de forma que el periodo sea coherente con el *Sampling Rate*.

¿Cómo consigue este bloque realizar un encapsulado IEC-61850? Tal y como se ha comentado en el apartado 3 de este Capítulo, se empleará el ejemplo extraído de la referencia [7] para realizar las pruebas sobre la plataforma desarrollada.

Retomando el ilustrativo contenido de la Figura 2-21 y completando los campos con la información disponible, se ha realizado un esquema similar que en este caso refleja el resultado del empaquetamiento de los datos extraídos de la memoria *DataROM* en el caso implementado.

Para una comprensión correcta de los campos correspondientes a las etiquetas y las longitudes, se recomienda la consulta del apartado 7.5.1 del Capítulo 2 de este documento.

La Figura 4-3 presenta esta trama APDU teórica.

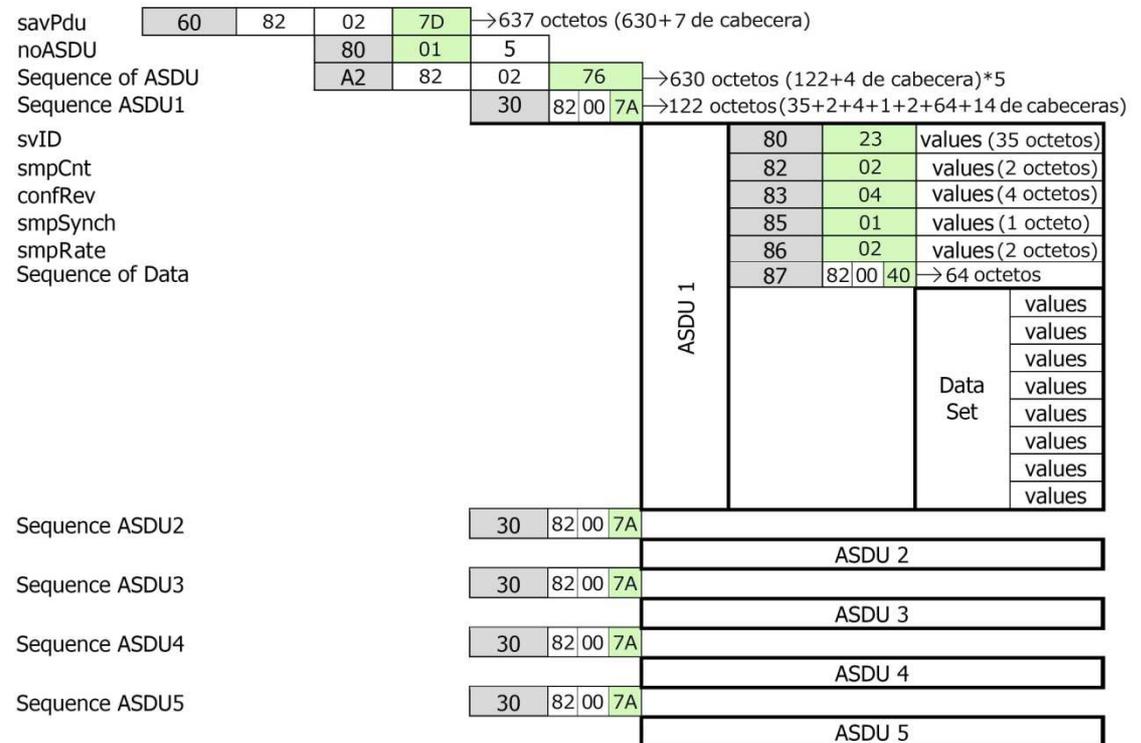


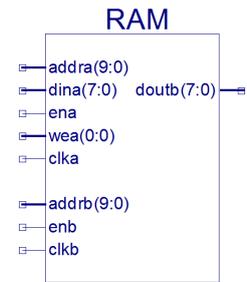
Figura 4-3. Esquema del APDU enviado.

Una vez que los datos son empaquetados siguiendo el modelo marcado por la norma, estos son enviados hacia la RAM para completar la trama antes de la transmisión. Para ello el módulo consta de dos buses para el acceso a la memoria (*RAM_add_wr* y *DatatoRAM*) y una señal de activación de la escritura (*wr_en*).

Hasta el momento se han descrito en los apartados 4 y 5 dos bloques con acceso a la memoria RAM para su configuración. Evidentemente, la escritura sobre este módulo no permite la superposición de señales que indiquen cuándo y dónde deben alojarse los diferentes valores enviados. A continuación se describe cómo se negocia esta cuestión desde el diseño global.

6. Módulo *RAM*

A la hora de gestionar la información de entrada y salida de este tipo de memoria, existe la posibilidad de configurar el módulo de manera que presente dos puertos.



En el diseño presentado se optó por reservar uno de ellos para permitir el acceso del bloque *ethernetSend* con el fin de que realice un barrido de la trama completa almacenada en *RAM*; es decir, que cumpliera la función de puerto de lectura.

La escritura de datos de la memoria supuso mayor complicación, ya que eran dos los módulos que debían acceder a los campos contenidos en ella. Afortunadamente, la configuración sólo se realiza una vez y de manera previa al inicio de la transmisión, lo que anula el riesgo de colisión entre las direcciones, datos o señales de activación.

De cualquier forma, los buses correspondientes al acceso de escritura de la memoria deben ser multiplexados en función del bloque al que le corresponda grabar su información en la trama. Para ello se optó por incluir un proceso asíncrono desde el bloque superior de control del diseño.

El contenido de este multiplexor se describe de la siguiente manera:

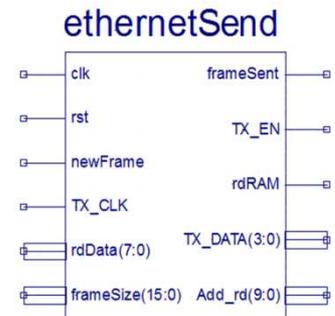
```

mux:
process(mvtoRAM_en, RAM_wr_en_config, RAM_wr_en_mv, RAM_wr_data_mv, RAM_wr_add_mv,
RAM_wr_data_config, RAM_wr_add_mv)
begin
    if mvtoRAM_en='1' then
        RAM_wr_en<=RAM_wr_en_mv;
        RAM_wr_data<=RAM_wr_data_mv;
        RAM_wr_add<=RAM_wr_add_mv;
    else
        RAM_wr_en<=RAM_wr_en_config;
        RAM_wr_data<=RAM_wr_data_config;
        RAM_wr_add<=RAM_wr_add_config;
    end if;
end process;

```

7. Módulo *ethernetSend*

Siguiendo la secuencia del flujo de datos del esquema de la Figura 4-1, el último paso para el envío de los datos vía Ethernet es la formación de la trama completa y la adaptación al medio físico teniendo en cuenta los siguientes puntos.



7.1 Composición de la trama

La Figura 3-4 muestra la trama Ethernet completa en la que se basa el diseño. Como se puede observar, los únicos campos que no han sido registrados en la memoria RAM son los correspondientes a la estructura del protocolo IEEE 802.3 para la transmisión de datos.

Así, una vez que este módulo recibe la señal de comienzo de la transmisión (*newFrame*), la comunicación con el controlador Ethernet *DP83865DVH Gig PHY* se activa y se inicia el lanzamiento de los campos correspondientes al preámbulo y comienzo de la trama (SFD).

Posteriormente, los datos comienzan a ser recogidos del bloque *RAM* para el cálculo del CRC de la trama MAC y su posterior envío.

La lectura de la memoria continúa hasta realizar un barrido completo de la trama IEC-61850. Para conocer el instante en el que se registra el último byte de la información, el módulo presenta un bus de entrada (*frameSize*) cuyo valor es calculado por el bloque *MVtoRAM* y transmitido a través de un bus de salida (*MAC_Size*).

Una vez que el ciclo de lectura de la RAM ha concluido, se procede a la adición del campo FCS a la trama, para cuyo cálculo se emplea una función independiente.

7.2 Cálculo del CRC

Tal y como se recogió en el Capítulo de introducción a esta memoria, durante la fase de investigación previa al proyecto se encontró un módulo VHDL que se adaptaba a las especificaciones de transmisión Ethernet (ver apartado 5.3.2 del Capítulo 1).

Este diseño sirvió como punto de apoyo para realizar el enlace de comunicación de la plataforma, aunque evidentemente hubo de ser modificado de una manera conveniente para adaptarlo a las especificaciones concretas de este proyecto.

El cambio más significativo que fue necesario realizar se refiere al paquete encargado del cálculo de CRC. La alta velocidad exigida por la transmisión Ethernet según la norma IEC-61850 hacía incompatible una obtención correcta de su valor con una comunicación a 100Mbps, por lo que no era posible registrar tramas enviadas debido a que la tarjeta del PC rechazaba la información al considerarla corrupta.

La implementación secuencial de esta función fue sustituida por otra puramente combinacional que permitía el paso de los datos byte a byte en cada ciclo de reloj, lo que ofrecía la posibilidad de sincronizar el cálculo con el envío de cada octeto independientemente de la velocidad requerida para ello. Se presenta aquí la razón de que la estructura de datos de la memoria emplee palabras de 8 bits, ya que el cálculo del CRC así lo exige.

A continuación se muestran los resultados de una prueba realizada con la herramienta de simulación *ISim* que permite observar la discrepancia entre los valores del CRC de cada uno de los módulos probados. El patrón que permitió conocer si el cálculo era correcto se obtuvo de diferentes herramientas web que posibilitan realizar este algoritmo sobre una secuencia de datos, generalmente en código ASCII.



Figura 4-4. Cálculo incorrecto del CRC ante los datos ASCII 123456789.

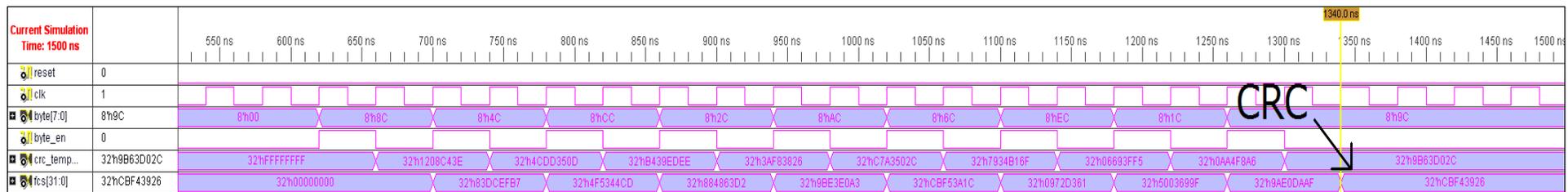


Figura 4-5. Cálculo correcto del CRC ante los datos ASCII 123456789

7.3 Transmisión física

La adaptación de un diseño digital a la tarjeta física en la que se desea implementar es ineludible. En este caso, la existencia de un controlador dedicado a la comunicación con el puerto Ethernet obligó al estudio del mismo para conocer los detalles de su funcionamiento.

En la hoja de características del *NS DP83865* recogidas en la Parte IV de este volumen se encuentra la información relevante a este respecto.

Así, elementos como la señal *Reset* o el tipo de negociación activado, al que sólo se podía acceder mediante recolocación de los *jumpers* soldados en la tarjeta (véase la página 114 de esta memoria) resultaron cruciales para un correcto funcionamiento de la tarjeta.

Con el modo de auto-negociación seleccionado por defecto, se ofrecía la posibilidad de trabajar a 10Mbps o a 100Mbps, dependiendo siempre del dispositivo receptor. Nótese que este diseño soporta únicamente el envío de datos, por lo que la característica de funcionamiento *Full Duplex/Half Duplex* no altera el comportamiento del módulo obtenido.

La única variable que puede influir sobre el efecto de la velocidad en el tratamiento de los datos es el reloj de entrada. Debido a esto, se habilitó el oscilador más rápido presente en la tarjeta como señal de reloj para la FPGA, con una frecuencia de 125MHz.

Como se puede extraer de la documentación correspondiente al controlador IEEE 802.3, la transmisión por medio de Ethernet (10Mbps) o Fast Ethernet (100Mbps) emplea cuatro salidas digitales para el lanzamiento de los datos, por lo que cada uno de los bytes de la trama descrita anteriormente se divide en dos para su envío. El diseño de este módulo coloca en el buffer de salida en primer lugar los cuatro bits menos significativos; es decir, del 0 al 3 siendo el 0 el LSB.

La señal de entrada *TX_CLK* registra un valor periódico obtenido desde el controlador Ethernet de la tarjeta, el cual impone la velocidad de transmisión. Este reloj puede ser de 2.5MHz (10Mbps) o 25MHz (100Mbps), y es el que se emplea para sincronizar el envío y captura de

los valores. Téngase en cuenta que en el peor de los casos la velocidad a la que se tratan los datos es cinco veces mayor que la de transmisión, lo que aporta cierta holgura a la hora de gestionar los diferentes cálculos.

7.4 Máquina de estados

Debido a la cantidad de estados que presenta este módulo, resulta de gran ayuda disponer de un diagrama que muestre al menos la función de cada uno de ellos así como las condiciones que se deben dar para las transiciones intermedias. La Figura mostrada a continuación representa dicha información de manera esquemática.

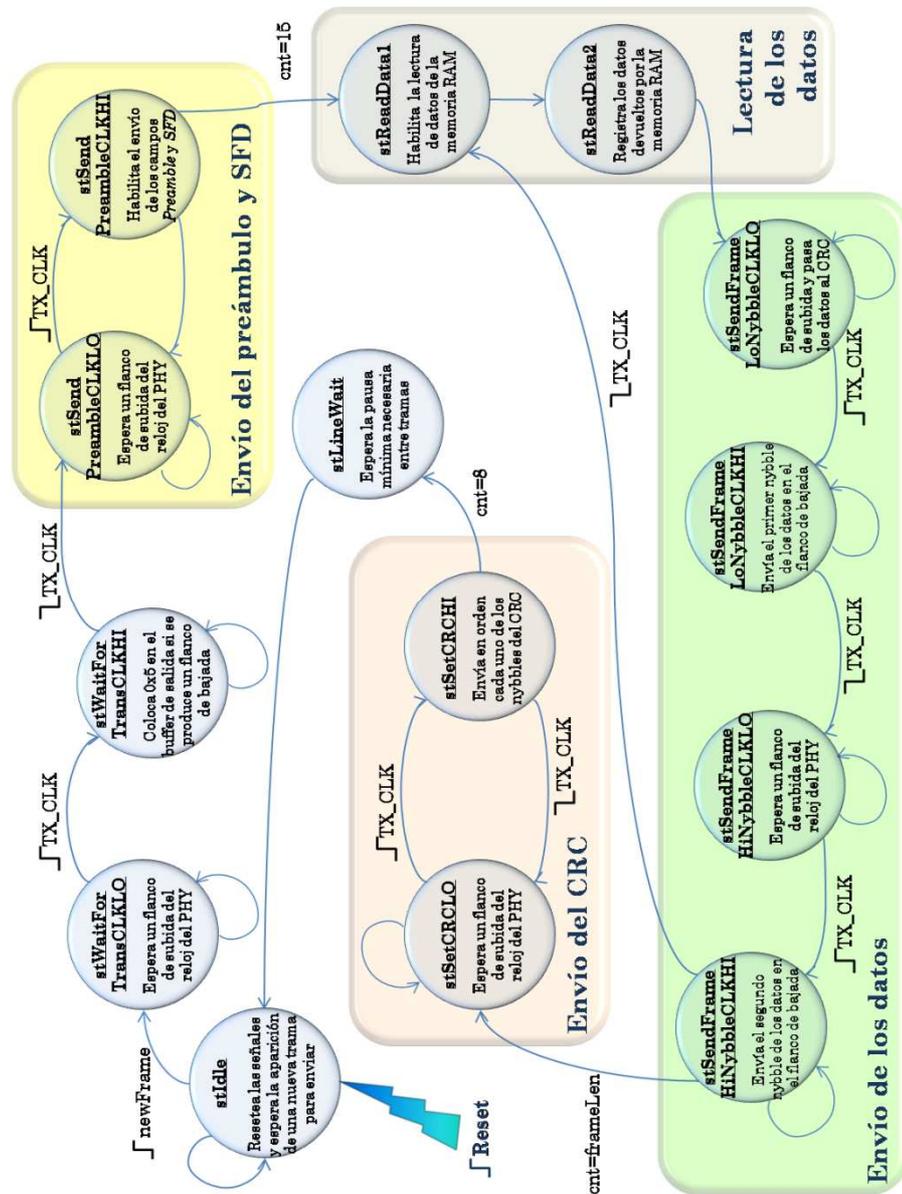
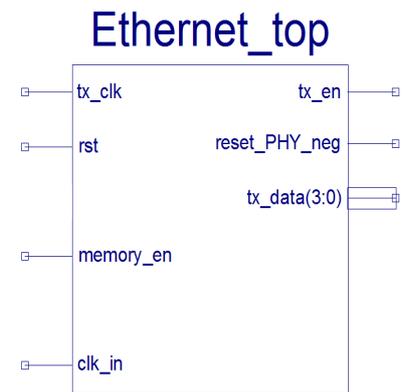


Figura 4-6. Diagrama de estados del módulo ethernetSend.

8. Módulo *Ethernet_top*

Aunque este módulo no figure en el esquema general del diseño debido a que no tiene una función concreta y específica dentro del flujo de datos, es evidente que la gestión de las conexiones de los distintos bloques y la sincronización entre ellos precisan de un bloque de jerarquía superior.



8.1 Puertos

El módulo *Ethernet_top* representa en cuanto a nivel de estructuración a la pastilla que contiene la FPGA.

Por ello, la interfaz está constituida en este caso únicamente por dos señales que se activan desde sendos pulsadores de la tarjeta de desarrollo (*rst* y *memory_en*) que se emplean para el reseteo e inicio del proceso de configuración y envío.

Como es de suponer, el funcionamiento de la plataforma requiere en adición la entrada de los relojes de sincronización (*clk_in*) y de envío de datos (*tx_clk*) según las especificaciones Ethernet.

En cuanto a las salidas de este bloque sólo son necesarias las correspondientes al envío de los datos (*tx_data*), habilitación del controlador (*tx_en*) y reseteo de este último (*reset_PHY_neg*), cuya temporización es fundamental para el establecimiento del diálogo entre el módulo Ethernet emisor y el receptor correspondiente.

Además de soportar entradas y salidas, y de establecer las distintas conexiones entre los diferentes módulos del diseño, el bloque *Ethernet_top* gestiona la información intercambiada entre ellos, así como su temporización y activación.

8.2 Máquina de estados

Para que estas tareas fueran realizadas de forma eficaz, se diseñó una máquina de estados en función de señales internas y las entradas.

La Figura 4-7 muestra el diagrama de este diseño.

De manera esquemática se representan en forma de círculo los diferentes estados en los que puede operar el módulo, además de información sobre la tarea que realiza cada uno.

Las flechas que constituyen las uniones entre ellos reflejan las transiciones causadas por los flancos de subida o bajada de las señales correspondientes.

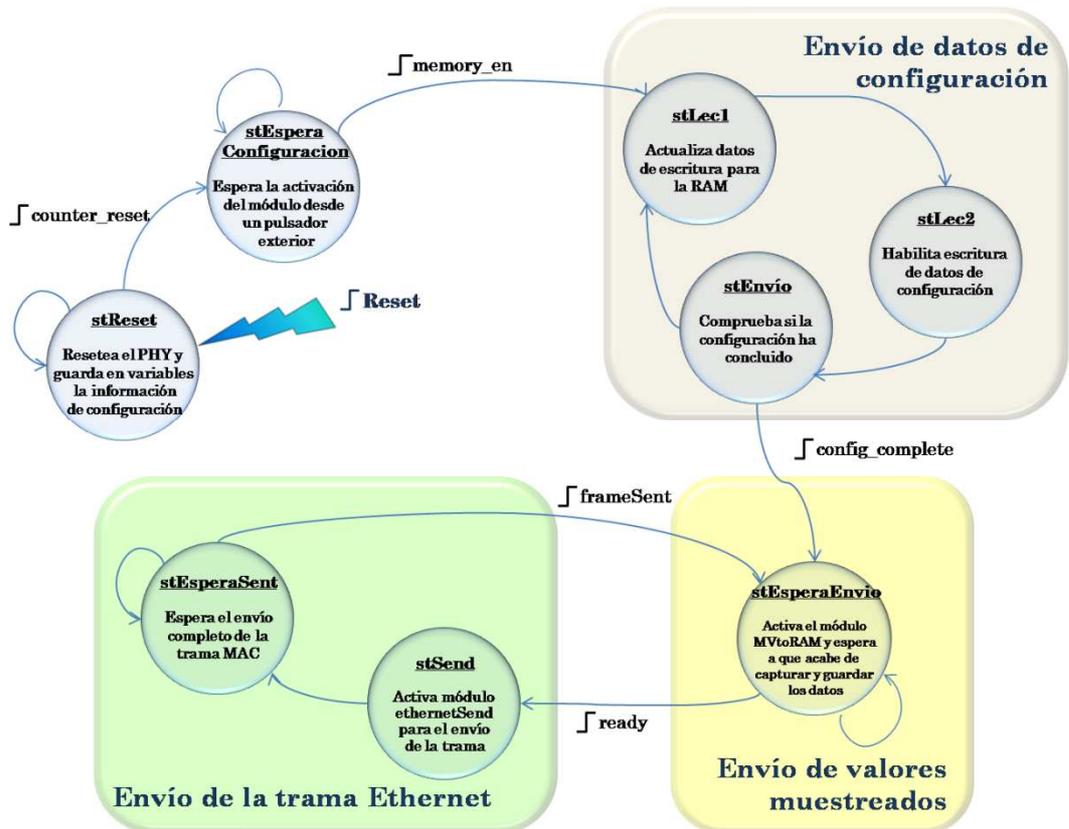


Figura 4-7. Diagrama de estados del módulo Ethernet_top

Capítulo 5. EXPERIMENTOS Y RESULTADOS

En el Capítulo 4 se detallan las características de la solución desarrollada como respuesta a las especificaciones del proyecto.

Para verificar la funcionalidad del diseño es necesario someter el módulo obtenido a diversas pruebas, cuyos resultados se presentan en los siguientes apartados.

En primer lugar se realiza un banco de pruebas sobre la arquitectura con el fin de extraer una simulación de su comportamiento ante estímulos predefinidos. Estos resultados se recogen en el primer apartado.

El apartado 2 muestra las capturas realizadas desde el PC de los datos enviados por la plataforma una vez que ésta ha sido descargada a la FPGA y se ha puesto a prueba la validez del protocolo de comunicación vía Ethernet.

1. Simulación del módulo

Antes de mostrar las respuestas del módulo a las diferentes simulaciones, resulta interesante realizar una introducción sobre el modo en que se han obtenido. Para ello se reserva el punto siguiente, donde se explica de qué modo se generan las entradas que influyen en el comportamiento de la plataforma.

1.1 *Test bench*

Un *test bench* o banco de pruebas es un fichero que describe una serie de estímulos de manera textual, sin estar sometida su sintaxis a las restricciones del lenguaje HDL, ya que se pueden añadir funciones más complejas, como por ejemplo sentencias de espera.

La excitación de un bloque funcional de esta manera resulta muy potente, ya que permite mucha mayor flexibilidad que la ardua

configuración de señales de entrada en modelos de ondas (archivos *waveform*).

Otra virtud de este tipo de ficheros es que posibilita actuar con el módulo testeado de manera bidireccional, comprobando si las diferentes señales presentes en el diseño han tomado los valores esperados.

Este modo de funcionamiento se representa en el esquema mostrado en la siguiente Figura.

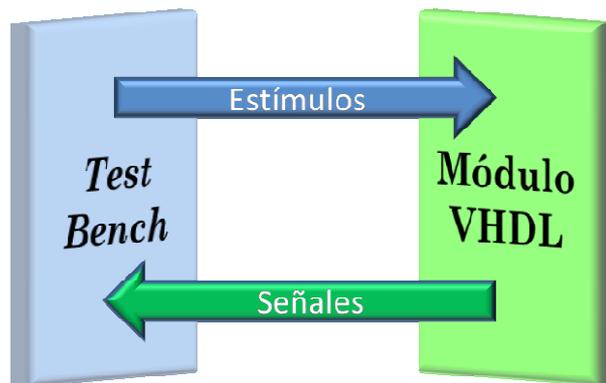


Figura 5-1. Estructura de un proceso de test.

Una de las ventajas del diseño hardware es que su gran modularidad permite testear el comportamiento de los diferentes bloques por separado.

Evidentemente, a lo largo del proceso de elaboración de este proyecto se emplearon multitud de *test benches* para chequear el correcto funcionamiento de cada uno de los módulos. Por ejemplo, se construyeron bancos de pruebas para comprobar la temporización de la escritura y lectura sobre las memorias para evitar colisiones.

La inclusión de cada uno de estos diseños en el documento resulta innecesaria, ya que aportaría información parcial sobre las pruebas realizadas al módulo completo, que son las que realmente merecen un análisis debido a su importancia.

En la página 79 se puede encontrar el resultado de aplicar el mismo *test bench* a dos módulos diferentes, que por lo tanto reciben datos de entrada idénticos.

1.2 Resultados de la simulación

La realización de un banco de pruebas para el diseño completo únicamente requiere la configuración de las señales de entrada del bloque *Ethernet_top*, que se corresponden a relojes, reset y activación. Téngase en cuenta que los datos que se envían y sus características están recogidos en las distintas memorias ROM, tal y como se expuso en el Capítulo 4.

El objetivo de este apartado es la presentación de diferentes capturas correspondientes a la simulación del bloque jerárquico superior.

La extensión de la trama Ethernet enviada (nótese que el tamaño máximo es de 1518 bytes) hace que sea imposible mostrar un resultado completo de forma definida, por lo que se opta por presentar únicamente algunos campos de especial relevancia en la estructura de datos. Las pruebas se realizaron con una velocidad de envío de 100Mbps.

En las páginas consecuentes se recoge el comportamiento de las diferentes señales al someter al módulo a un proceso de transmisión con las características tratadas a lo largo de los distintos Capítulos de esta memoria.

Únicamente, para facilitar la comprensión de las tramas capturadas, es necesario recordar que el envío Ethernet se realiza anteponiendo el *nybble* (conjunto de cuatro bits) menos significativo al más significativo.

Resulta de utilidad la comparación de los distintos campos señalados con la información contenida en la Figura 4-7, además de una comprensión general del funcionamiento de cada uno de los bloques descritos en el Capítulo 4.

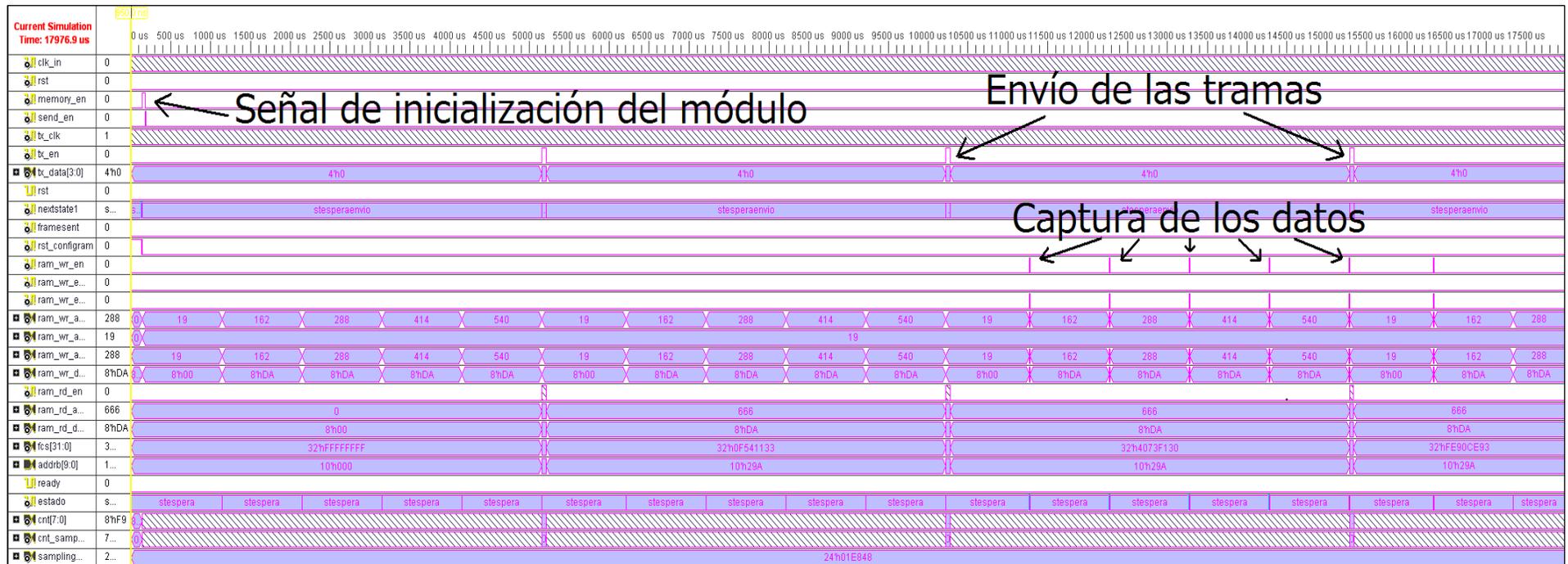


Figura 5-2. Respuesta del diseño ante simulación. Captura y envío de datos

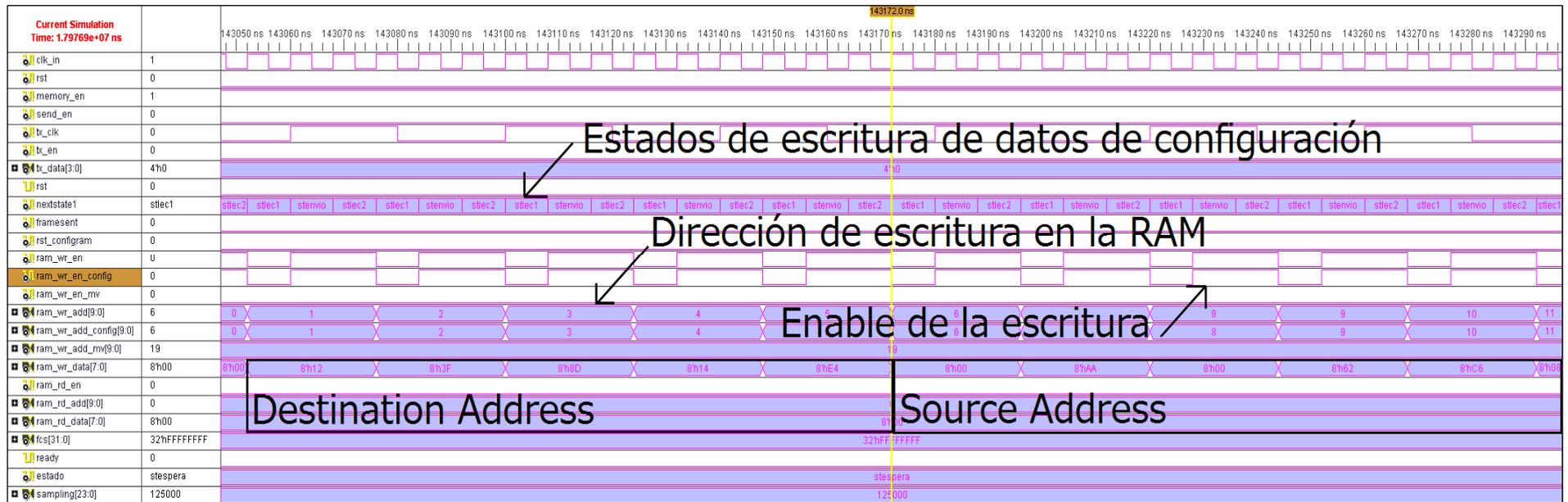


Figura 5-3. Respuesta del diseño ante simulación. Escritura de información de configuración en la memoria RAM

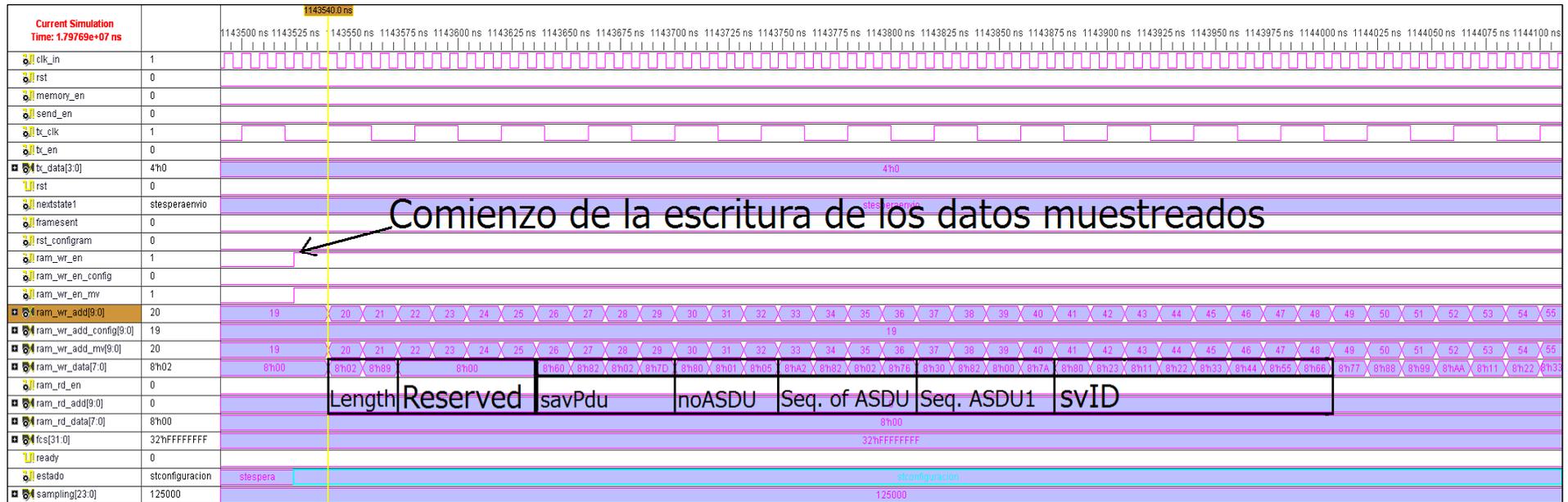


Figura 5-4. Respuesta del diseño ante simulación. Escritura de datos muestreados en la memoria RAM (I)

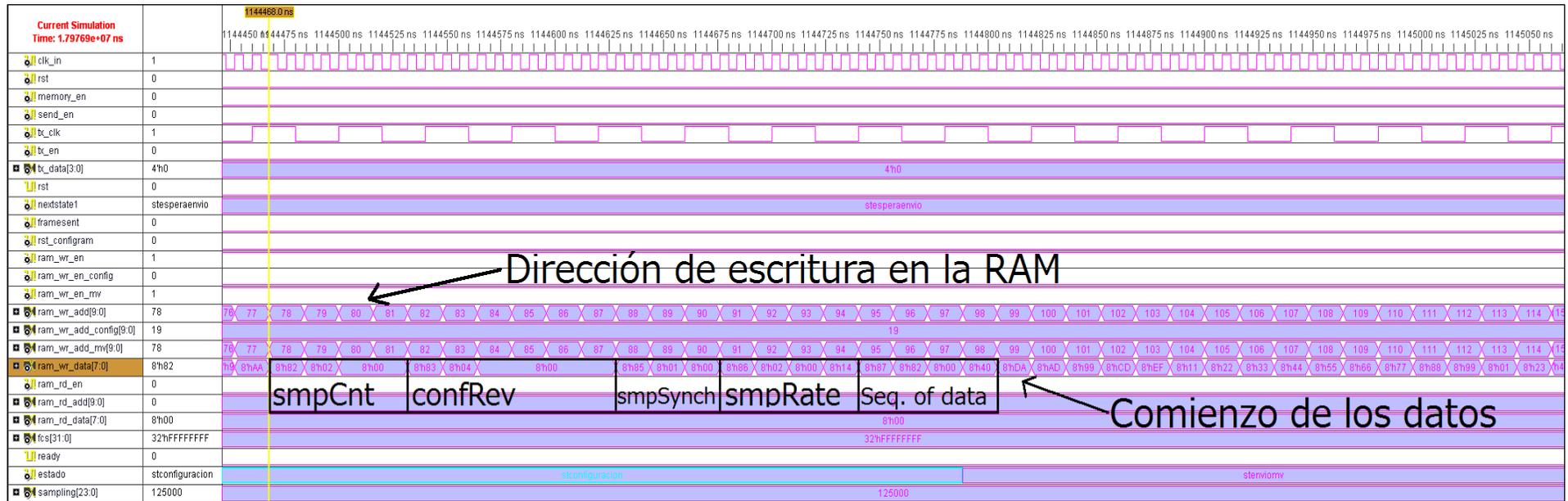


Figura 5-5. Respuesta del diseño ante simulación. Escritura de datos muestreados en la memoria RAM (II)

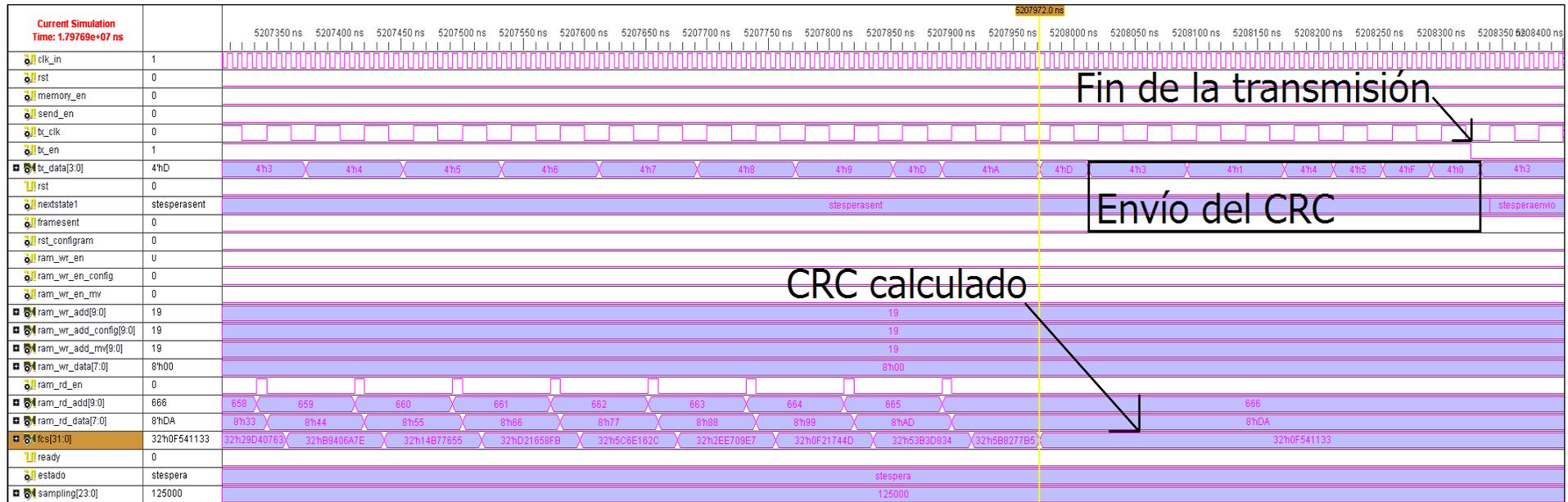


Figura 5-6. Respuesta del diseño ante simulación. Envío del CRC

A la vista de estos resultados, se puede comprobar que coinciden con lo esperado. Sin embargo, este test no es definitivo para dar por válido el diseño, ya que es ineludible superar la prueba de campo que consiste en volcar la implementación dentro de la FPGA y conectar ésta a la tarjeta Ethernet del PC para examinar la validez del funcionamiento.

2. Pruebas de transmisión

Antes de incluir las tramas físicas obtenidas de manera experimental, parece necesario describir una herramienta disponible en el software empleado de gran utilidad para la depuración de los diseños hardware.

2.1 *ChipScope Pro Analyzer*

La comprobación de circuitos lógicos resulta compleja en aplicación directa, debido a la inviabilidad de realizar una monitorización de todas las señales presentes en un proyecto.

En el diseño hardware la temporización es crítica, y en las aplicaciones habituales de este tipo de tecnología se trabaja con relojes de sincronización con frecuencias del orden de MHz, lo que provoca cambios en los valores de las señales en periodos de nanosegundos. Por ejemplo, el oscilador empleado en este trabajo es de 125MHz, lo que se corresponde con variaciones separadas entre sí 8ns.

Es imposible observar estos cambios a menos que se cuente con un analizador lógico potente. Sin embargo, esto no es suficiente, ya que los pines de una FPGA suelen ser inaccesibles, y los puertos de salida de datos son muy limitados. La única solución pasaría por conectar desde la fase de diseño la señal que se desea controlar a un LED de la tarjeta de desarrollo. Esto reduce el número de señales monitorizables aproximadamente a una decena.

El software *Xilinx ISE Design Suite* aporta una herramienta para tratar de paliar este problema de depuración. Se trata de una aplicación que genera un núcleo (*core*) a la vez que se sintetiza el diseño lógico que se quiere testear, el cual se vuelca en el chip de la misma forma que se hace con el circuito lógico implementado.

Este nuevo módulo consiste en un vector de tamaño configurable capaz de almacenar datos binarios de las señales elegidas dentro del proceso.

Pese a que a primera vista puede parecer que se ha encontrado una solución eficaz a los problemas que presenta el chequeo de una implementación física, lo cierto es que esta herramienta resulta mucho más tosca y oscura que el resto de utilidades del paquete *Design Suite*.

Por ejemplo, el simple hecho de comprobar la situación actual de una máquina de estados, lo cual se puede realizar de manera muy sencilla en un proceso de simulación, exige en primer lugar la identificación de las señales que codifican cada uno de los estados de la máquina en el proceso de sintetizado, su inclusión en el núcleo del *ChipScope* y por último la decodificación de la información binaria de cada señal.

Las limitaciones de esta utilidad no se quedan ahí, ya que ocupa un gran tamaño dentro de la FPGA, lo que desemboca en la situación paradójica de que sobre una misma placa, cuanto más complejo sea el diseño implementado menos espacio habrá para almacenar información de las señales. Otro inconveniente es la necesidad de mantener la plataforma conectada al PC durante las pruebas.

Una vez descrita esta herramienta, sólo queda mostrar los resultados de las pruebas realizadas sobre el envío de los datos

2.2 Resultado experimental

Tal y como se mostrará en la siguiente Figura, el empleo del software *WireShark* permitió la captura de las tramas enviadas al PC empleado en la realización del proyecto.

Llegados a este apartado, sólo cabe la comparación de la información mostrada con los valores que se incluyen en las Figura 4-3, Figura 5-2, Figura 5-3, Figura 5-4, Figura 5-5 y Figura 5-6.

En la siguiente página se muestra una captura de las tramas realizadas. Obsérvese que el mismo analizador reconoce el tipo de datos enviados (*IEC61850/SV*) gracias a que el campo *Ethertype* contiene un valor reservado por la IEEE a tal efecto.

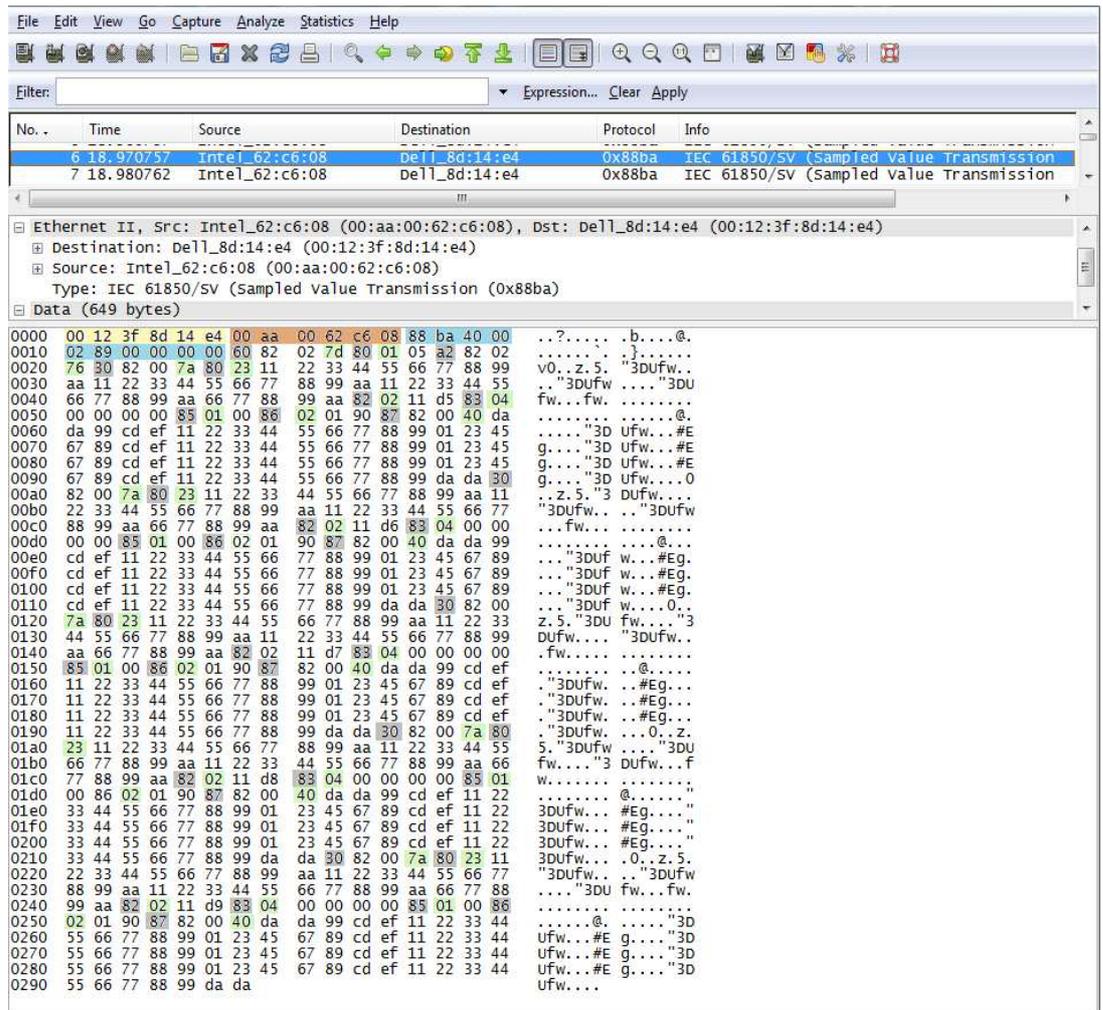


Figura 5-7. Captura de la trama enviada una vez leída por el PC.

En la Figura anterior, con cuyo óptimo resultado se cierran los test realizados sobre el módulo, se han indicado con diferentes colores los campos más fácilmente identificables de la transmisión.

Así, se muestran las direcciones de destino y fuente, y los campos configurables de la trama Ethernet, además de las etiquetas correspondientes a *Tag* y *Length* definidas por el estándar ASN.1 con la misma leyenda cromática que en la Figura 4-3.

Como último apunte, señalar que ni el valor del CRC calculado ni los campos que gestionan la prioridad dentro del protocolo VLAN aparecen en la captura. Esto es debido a que la tarjeta de adquisición de datos Ethernet emplea el CRC solamente para ignorar aquellos mensajes erróneos, y de la misma forma gestiona internamente las prioridades de los mensajes recibe.

Capítulo 6. CONCLUSIONES

A la vista de los resultados logrados y las especificaciones iniciales del proyecto, se puede afirmar que el desarrollo de éste ha concluido de manera satisfactoria.

Como resultado del trabajo realizado, se ha obtenido una plataforma que precisa únicamente de una pastilla FPGA y un dispositivo controlador para el envío de datos a través de un puerto Ethernet. Con estas mínimas exigencias tanto de dispositivos hardware digitales como de alimentación, se cumple un requisito fundamental para que el empotramiento del módulo sea posible, ya que se ha conseguido un diseño ligero y compacto.

El modelo de encapsulado de datos constituía la otra exigencia básica para el desarrollo de este proyecto. A este respecto, se ha implementado una arquitectura lógica fácilmente configurable y adaptable tanto a las especificaciones de comunicación de unos dispositivos emisor-receptor determinados como al modelo de datos recogido en la norma IEC-61850, concretamente en lo correspondiente al servicio de envío de valores analógicos muestreados.

Todas estas características confluyen en la obtención de una plataforma de comunicación cuya utilidad se adapta perfectamente a los objetivos perseguidos por el proyecto DENISE donde se enmarca este trabajo.

Por ello se espera que en un futuro sirva de base para realizar aplicaciones reales que faciliten la integración y la gestión eficiente de todos los niveles de los sistemas de generación, distribución y consumo de energía eléctrica.

Capítulo 7. FUTUROS DESARROLLOS

Tomando como punto de partida los objetivos alcanzados en este proyecto, la obtención de una plataforma completamente empotrable a nivel físico dentro de un dispositivo eléctrico requiere aún el desarrollo de diferentes interfaces.

Como se describió en el Capítulo 1 de esta memoria, la plataforma buscada consta de un módulo VHDL configurable capaz de enviar datos a través de un puerto Ethernet.

Pese a que efectivamente existe un módulo encargado de almacenar la información relativa a la configuración del dispositivo, no es posible acceder a él durante el funcionamiento de la tarjeta, debido a que se trata de una memoria ROM que únicamente admite modificación a través del programa de diseño HDL y exige una nueva síntesis y descarga del código completo a la FPGA.

Esta situación se repite con la información que se desea enviar. En este caso, ante la falta de una muestra real de valores analógicos o digitales, se optó por incluir los datos en otra memoria del mismo tipo.

Por lo tanto, parece importante completar este diseño con los siguientes desarrollos:

1. Enlace de configuración del dispositivo

No tiene sentido que la configuración de un sistema empotrado flexible y adaptable provenga de la información de una ROM, sino que es necesario que ésta pase a ser una memoria RAM a la que se pueda acceder desde el exterior sin necesidad de volcar de nuevo todo el diseño sobre la tarjeta. A tal efecto, se plantean dos opciones:

- Por un lado se podría hacer uso de otro enlace Ethernet; en este caso de recepción.
- Otra posibilidad es acceder a través del puerto serie RS-232.

Debido a los recursos de los que consta la tarjeta de desarrollo utilizada en el proyecto (véase la Figura 1-10 o la hoja de características), en principio se presenta el empleo del puerto serie como la única opción viable para establecer el enlace de configuración.

2. Etapa de captura de datos

La información tratada en el transcurso de este proyecto era de tipo digital. Parece obvio que un dispositivo de comunicación encargado del envío de medidas de variables eléctricas requiere el empleo de un módulo cuyo propósito sea el muestreo de dichas variables.

Por ello, resulta necesaria la inclusión de un conversor A/D que transforme los valores analógicos en información incorporable al flujo del diseño lógico, tal y como se ilustró en la Figura 1-3.

Una vez que los datos son capturados, deben ser tratados y encapsulados por el bloque de gestión, lo que requiere la sustitución de la memoria ROM actual por un buffer de entrada que se actualice de manera síncrona con el resto del diseño.

3. Empleo de fibra óptica como medio de transmisión

Como se indicó en el apartado 7.2 de este documento, la norma IEC-61850-9-2 recomienda el empleo del enlace físico 100BASE-FX.

Si bien en principio se puede pensar que el uso de una fibra óptica restringiría la adaptabilidad del dispositivo al requerir un cableado más delicado y costoso, parece claro que las mejores prestaciones en cuanto a velocidad y perspectivas de futuro de esta tecnología aconsejan incorporar esta opción de comunicación.

Nótese que la tarjeta Spartan 3A-DSP carece de puerto de fibra óptica (véase la Figura 1-10), por lo que un futuro desarrollo en este sentido obligaría a realizar un salto hacia otra plataforma. No obstante, el empleo de esta vía para la transmisión de los datos liberaría al módulo Ethernet del envío de valores muestreados y habilitaría su uso como receptor para el acceso a la memoria de configuración.

BIBLIOGRAFÍA

- [1] Santodomingo, Rafael. *Modelos de información en sistemas eléctricos de potencia*. Trabajo de investigación preparado en el IIT de la Universidad Pontificia Comillas para la obtención del Diploma de Estudios Avanzados (DEA). Director: José Antonio Rodríguez Mondéjar.
- [2] International Electrotechnical commission, IEC 61850.
Communication Networks and Systems in Substations – Part 9-1: Specific communication service mapping (SCSM) – Sampled values over serial unidirectional multidrop point to point link. 57/619/FDIS. Ed. 1, 2002.
- [3] International Electrotechnical commission, IEC 61850.
Communication Networks and Systems in Substations – Part 9-2: Specific communication service mapping (SCSM) – Sampled values over ISO/IEC 8802-3. 57/690/FDIS. Ed. 1, 2003.
- [4] International Electrotechnical commission, IEC 61850.
Communication Networks and Systems for in Substations - Part 1: Introduction and overview.. Revision 2.0: February 16, 2001.
- [5] International Electrotechnical commission, IEC 61850.
Communication Networks and Systems for power utility automation – Part 7-1: Basic communication structure – Principles and models. 57/942/CDV. Ed. 2, 2008.
- [6] International Electrotechnical commission, IEC 61850.
Communication Networks and Systems for power utility automation – Part 7-2: Basic information and communication structure – Abstract communication service interface (ACSD). 57/891/CD. Ed. 2, 2007.
- [7] Brunner, C; Lang, G; Leconte, F; Steinhauser, F. *Implementation guideline for digital interface to instrument transformers using IEC 61850-9-2*. International Users Group, 2004-03-01.
- [8] Institute of Electrical and Electronics Engineering, IEEE Std. 802.3.
Standard for Information technology-Telecommunications and

information exchange between systems-Local and metropolitan area networks-Specific requirements. Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. Approved 9 June 2005.

- [9] Stallings, W. *Comunicaciones y redes de computadores*. Séptima edición. Prentice Hall, Pearson Educación, 2004.
- [10] Brown, S.; Vranesic, Z. *Fundamentos de lógica digital con diseño VHDL*. Segunda edición. McGraw Hill, 2006.
- [11] Ashenden, P. *The VHDL Cookbook*. Dept. Computer Science. University of Adelaida, July 1990.

Parte II. ESTUDIO ECONÓMICO

Como se expuso en el Capítulo introductorio de este documento, la intención del proyecto es el desarrollo de una plataforma empotrable, ligera y configurable para la transmisión de valores analógicos que aporten información sobre las condiciones eléctricas de un dispositivo.

Los beneficios de la implantación de soluciones de este tipo son evidentes.

Un caso de aplicación podría ser la medida del consumo energético de diferentes dispositivos integrados dentro de un sistema; por ejemplo, una serie de electrodomésticos en una vivienda. El acceso a este tipo de información en tiempo real facilitaría enormemente la gestión eficiente de la energía tanto a nivel de usuario consumidor como a nivel de generación y distribución por parte de la compañía eléctrica.

La estandarización es otro punto clave del proyecto. La ardua tarea de estudio de la norma IEC-61850 se ve compensada por la obtención de una plataforma perfectamente integrable dentro de un sistema de energía eléctrica desarrollado según las convenciones citadas, con independencia del fabricante de cada dispositivo.

Estas ventajas particulares correspondientes al desarrollo de este diseño concreto complementan a los beneficios generales del proyecto DENISE donde se enmarca, los cuales se enumeran a continuación:

- Mejorar la calidad de suministro de energía.
- Posibilitar el uso de información en tiempo real con el objetivo de permitir la óptima gestión de la demanda y la oferta, en aras de conseguir mayor eficiencia energética y seguridad de suministro y mayor control del consumo.
- Soportar una nueva generación de servicios de energía y comunicaciones interactivas.
- El desarrollo de una arquitectura de control y de una nueva generación de dispositivos para las redes eléctricas de distribución, de alta interoperabilidad, sincronismo e integración, que puedan integrarse fácilmente en redes ya existentes.

- La potenciación de la capacidad y la imagen de España como país innovador y exportador de tecnología en el ámbito de la nueva generación de redes de distribución eléctrica.
- Situar a España en un área tecnológica en la que se espera una ruptura científico-tecnológica que vendrá motivada por las exigencias medioambientales, la necesidad de permitir la entrada masiva a las fuentes de energía renovables, así como por la calidad que la sociedad y la industria exigen al sector eléctrico.
- La alta proyección internacional de una investigación que abarca áreas tecnológicas de futuro, tanto por el interés que el tema suscita en el mundo como por la condición de los socios participantes en el consorcio.

Parte III. CÓDIGO FUENTE

DESCRIPCIÓN VHDL

Dada la amplia cantidad de código generado en este proyecto y a fin de evitar añadir numerosas páginas a esta memoria, se incluye el mismo en un CD adjunto, el cual puede encontrarse al final del presente volumen.

Parte IV. DATASHEETS

DOCUMENTACIÓN

En las siguientes páginas se pueden encontrar extractos de las hojas de características de cada dispositivo con la información que resultó relevante para el desarrollo del proyecto.

La documentación completa está disponible en las web de los fabricantes, cuyos enlaces son los siguientes:

- Spartan 3A-DSP 1800A

<http://www.xilinx.com/support/documentation/hw-sd1800a-dsp.htm>

- National Semiconductor DP83865DVH Gig PHYTER” V

<http://www.national.com/mpf/DP/DP83865.html>

SPARTAN 3A-DSP 1800-A



Spartan-3A DSP 1800A Board Features

Overview

The purpose of this manual is to describe the functionality and contents of the Spartan®-3A DSP Starter Platform from Xilinx. This document includes instructions for operating the board and descriptions of the hardware features.

For more information about this product, reference designs, and additional documentation, please visit the product home page found at www.xilinx.com/s3adspstarter or www.xilinx.com/s3adspmb or www.xilinx.com/s3adsp_sk.

Design Description

The Spartan-3A Starter Platform provides a platform for engineers designing with the Xilinx Spartan-3A DSP FPGA. The board provides the hardware to not only evaluate the advanced features of the Spartan-3A DSP, but also to implement complete user applications using peripherals on the Spartan-3A DSP Starter Platform and EXP modules, or both, plugged into EXP expansion connectors on the Spartan-3A DSP Starter Platform.

Features

The Spartan-3A DSP Starter Platform provides the following features:

- Xilinx 3SD1800A-FG676 FPGA
- Clocks
 - ◆ 125 MHz LVTTTL SMT oscillator
 - ◆ LVTTTL oscillator socket
 - ◆ 25.175 MHz LVTTTL SMT oscillator (video clock)
- Memory
 - ◆ 32M x 32 (128 MB) DDR2 SDRAM
 - ◆ 16Mx8 parallel and BPI configuration flash
 - ◆ 64Mb SPI Configuration and Storage Flash (with 4 extra SPI selects)
- Interfaces
 - ◆ 10/100/1000 PHY
 - ◆ JTAG programming and configuration Port
 - ◆ RS232 Port
 - ◆ Low-cost VGA

Design Description



- Buttons and switches
 - ◆ 8 User LEDs
 - ◆ 8-position user DIP switch
 - ◆ 4 User push button switches
 - ◆ Reset push button switch
- User I/O and expansion
 - ◆ Digilent 6-pin header (2)
 - ◆ EXP expansion connector (2)
- Configuration and debug
 - ◆ JTAG
 - ◆ SystemACE™ module connector
 - ◆ Eridon debug connector (SATA)

Ordering Information

Table 1 lists the evaluation kit part numbers.

Table 1: Evaluation Kit and Hardware Ordering Information

Part Number	Hardware
HW-SD1800A-DSP-SB-UNI-G	XtremeDSP Starter Platform - Spartan-3A DSP 1800A Edition



Functional Description

A high-level block diagram of the Spartan-3A DSP Starter Platform is shown in Figure 1. Subsequent sections provide details of the board design.

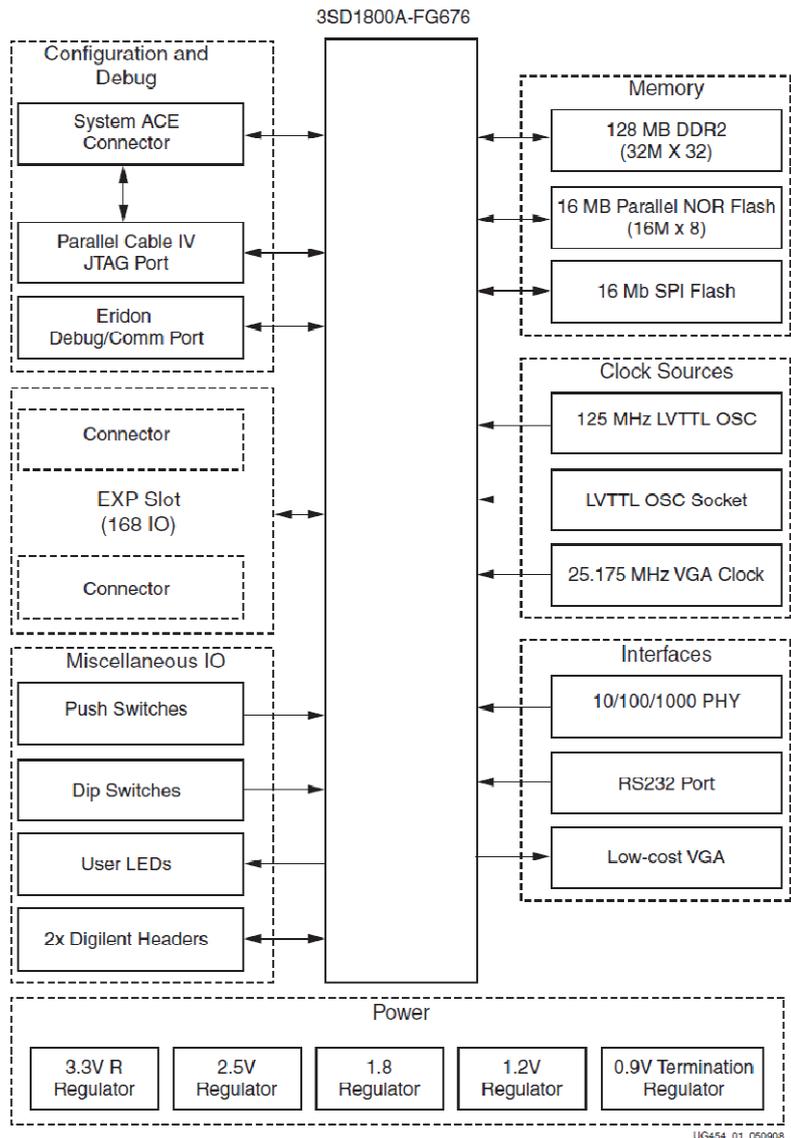


Figure 1: Spartan-3A DSP Starter Platform Block Diagram

Xilinx Spartan-3A DSP FPGA

The Xilinx XC3SD1800A-4FG676C device designed into the Spartan-3A DSP Starter Platform provides four I/O banks — two are fixed voltage and two are I/O voltage-selectable. The four I/O banks are described in Table 2 and detailed I/O pin usage is provided throughout this document. In some cases, voltage translation of I/O signals may be necessary in order to meet requirements of peripheral devices that are connected to a particular I/O bank, or EXP modules plugged into the EXP expansion connectors. Table 2 indicates where voltage translation will occur or, if a source is fixed at a particular voltage, even though the bank to which it is connected may be at different I/O voltage.

Table 2: XC3SD1800A Input and Output Allocation

I/O Bank Number	I/O Function	Number of I/O Pins	Number of Input-only Pins	Voltage Translation	Bank I/O Voltage
0	EXP Connector JX1	84	0		2.5 or 3.3 V
0	User Pushbuttons	0	4		
0	8-pos DIP Switch	0	8		
0	125 MHz Clock	1	0		
0	SMA Connector	1	0		
0	Configuration (PUDC_B)	1	0		
1	SystemACE Module	28	1		3.3 V
1	J3 Flash Memory	28	0		
1	SPI Selects (4)	4	0		
1	DAC	14	0		
1	User LEDs	8	0		
1	Digilent Connectors	8 ⁽¹⁾	0		
1	RS-232	2	0		
1	Eridon Debug Connector	4	0		
1	Configuration (Suspend)	1	0		
1	25.175 MHz Clock	1	0		
1	EXP Connector JX2	8	0	2.5 or 3.3 V	
2	EXP Connector JX2	76	0		2.5 or 3.3 V
2	J3 Flash Memory	8 ^[Ref 1]	0	2.5 or 3.3 V	
2	SPI	4 ^[Ref 1]	0		
2	Configuration	5	0		
2	Plug-in Reset	1	0		
2	LED	1	0		
2	Power-on Reset	0	1		



Table 2: XC3SD1800A Input and Output Allocation (Cont'd)

I/O Bank Number	I/O Function	Number of I/O Pins	Number of Input-only Pins	Voltage Translation	Bank I/O Voltage
3	DDR2 Memory	73	0		1.8 V
3	Ethernet	17	13	2.5 V	

Note:

1. FLASH_DO and SPI_MISO are a common pin.

Memory

The Spartan-3A DSP Starter Platform is populated with both high-speed RAM (128Mbytes DDR2) and non-volatile ROM (16Mbytes parallel, and 64Mbit serial) to support various types of applications. Additionally, a 50-pin connector is provided for SystemACE interface (not included) that can be used to configure the Spartan-3A DSP FPGA, and to provide storage for A/V media files from removable Compact Flash cards. Figure 2 shows a high-level block diagram of the memory interfaces on this board.

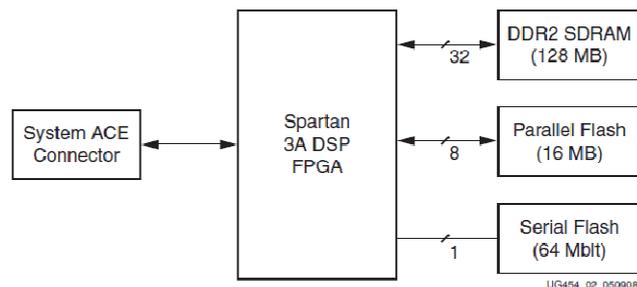


Figure 2: Spartan-3A DSP Memory Interface

Micron DDR2 SDRAM Interface

Two Micron MT47H32M16BM DDR2 devices provide 128Mbytes of 32-bit wide memory to the FPGA. The FPGA DDR2 interface supports SSTL18 signaling and all DDR2 signals are controlled impedance. The DDR2 data, mask, and strobe signals are matched length across all signals. Using I/O Bank 3 the DDR2 pinout on the FPGA is MIG-compliant. Pinout of the FPGA DDR2 interface is provided in Table 3 below.

SSTL18 Class I termination (series termination at the driving node and stub termination at the receiving node) is used between the FPGA and DDR on all unidirectional signals, such as Address, Control, and Clock. SSTL18 Class II termination (series and stub termination at both nodes) is used on the FPGA side of the interface for all bidirectional signals such as Data and Strobe. No board terminations are implemented for Data and Strobe on the memory side. ODT is strapped as enabled, although an FPGA I/O is still connected to allow the FPGA to drive this off. Note that stub terminations for address and control signals are not populated in production based on HyperLynx simulation and exhaustive, error-free testing of the FPGA-DDR2 interface at 133MHz.

Other interfaces consist of two 0.1" 6-pin headers to accept Digilent plug-in modules, a 7-pin Serial ATA connector (this is not a serial ATA interface) to connect to an Eridon debug module, and a 0.1" 2 x 6 header for SPI interface expansion.

National Semiconductor 10/100/1000 Ethernet PHY

The PHY is a National DP83865DVH Gig PHYTER™ V. The DP83865 is a low power version of the National Gig PHYTER V with a 1.8V core voltage and 2.5V I/O voltage. The PHY also supports 3.3V I/O, but the 2.5V option is used on the board. The PHY is connected to a Tyco-AMP RJ-45 jack with integrated magnetics (part number: 1-6605833-1). The jack also integrates two LEDs and their corresponding resistors as well as several other passive components. External logic is used to logically OR the three link indicators for 10, 100 and 1000 Mb/s to drive a Link LED on the RJ-45 jack. The external logic is for the default strap options and may not work if the strap options are changed. Four more LEDs are provided on the board for status indication. These LEDs indicate Link at 10 Mb/s, Link at 100 Mb/s, Link at 1000 Mb/s and Full Duplex operation. The PHY clock is generated from its own 25 MHz crystal (FOX FX325BS).

Figure 5 shows a high-level block diagram of the interface to the DP83865 Tri-mode Ethernet PHY. The PHY signal connections at the FPGA are listed in Table 7. These signals are connected to FPGA Bank 3 which is fixed at +1.8V I/O voltage, necessitating voltage translation between +1.8V and +2.5V to match the PHY I/O voltage requirements.

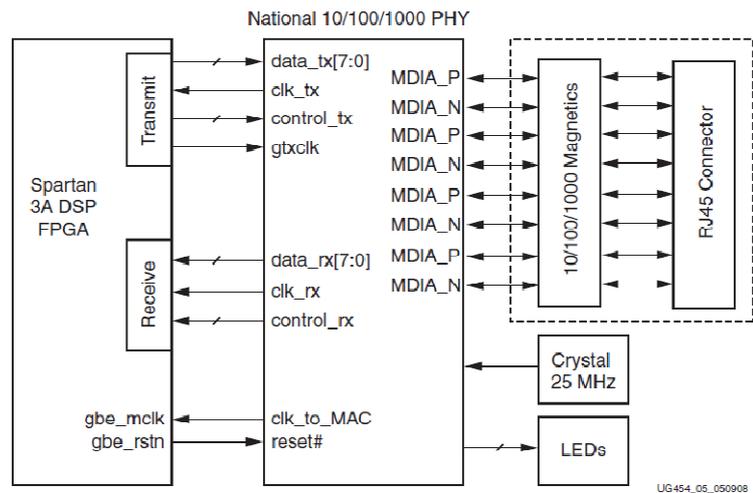


Figure 5: 10/100/1000 Mb/s Ethernet Interface



Table 7: Ethernet PHY Interface Signals

Ethernet PHY Signal	FPGA Pin	Ethernet PHY Signal	FPGA Pin
ETH_Tx_D	J8	ETH_COL	Y3
ETH_Tx_D	J9	ETH_INT#	J1
ETH_Tx_D	B2	ETH_Rx_DV	D1
ETH_Tx_D	B1	ETH_Rx_ER	J3
ETH_Tx_D	G6	ETH_MCLK	N6
ETH_Tx_D	H7	ETH_Rx_CLK	P1
ETH_Tx_D	K9	ETH_Tx_CLK	P2
ETH_Tx_D	K8	ETH_Rx_D0	C2
ETH_Tx_EN	D3	ETH_Rx_D1	G2
ETH_Tx_ER	E4	ETH_Rx_D2	G5
ETH GTX CLK	E3	ETH Rx D3	D2
ETH_MDC	F4	ETH_Rx_D4	Ab3
ETH_MDIO	F5	ETH_Rx_D5	Aa4
ETH_RST#	G4	ETH_Rx_D6	Ab4
ETH_CRIS	G1	ETH_Rx_D7	Y4

The PHY address is set to 0b00001 by default. PHY address 0b00000 is reserved for a test mode and should not be used. Three-pad resistor jumpers are used to set the strapping options. These jumper pads provide the user with the ability to change the settings by moving the resistors. The dual-function pins that are used for both a strapping option and to drive an LED have a set of two jumpers per pin. The dual-function pins are indicated by an asterisk in the table.

The default options as indicated in Table 8 are Auto-Negotiation enabled, Full Duplex mode, Speed advertised as 10/100/1000 Mb/s, PHY address 0b00001, IEEE Compliant and Non-compliant support, straight cable in non-MDIX mode, auto-MDIX mode enabled, Single node (NIC) and CLK_TO_MAC enabled. These and other settings are enabled by three-pad jumpers with a resistor connecting either pads 1 and 2 or pads 2 and 3.

Table 8: Ethernet PHY Hardware Strapping Options

Function	Jumper Installation	Resistor	Mode Enabled
Auto Negotiation	JT8: Pins 1-2	0 Ohm	Auto-negotiation enabled (default)
	JT9: Pins 1-2		
	JT8: Pins 2-3	0 Ohm	Auto-negotiation disabled
	JT9: Pins 2-3		
Full/Half Duplex*	JT10: Pins 1-2	0 Ohm	Full Duplex (default)
	JT10: Pins 1-2		
	JT10: Pins 2-3	0 Ohm	Half Duplex
	JT11: Pins 2-3		
Speed 1*	JT12: Pins 1-2 JT13: Pins 1-1 (Speed1 – 0)	0 ohm	Speed Selection: (Auto-Neg enabled) Speed1 Speed0 Speed Advertised 1 1 1000BASE-T, 10BASE-T 1 0 1000BASE-T 0 1 1000BASE-T, 100BASE-Tx 0 0 10BASE-T Default: 1000BASE-T, 100BASE-TX, 10BASE-T
	JT16: Pins 1-2 JT7: Pins 1-1 (Speed0 – 0)		
PHY address 0*	JT14: Pins 1-2	0 Ohm	PHY Address 0b00001 (default)
	JT15: Pins 1-2		
	JT24: Pins 2-3		PHY Address 0b00000
	JT25: Pins 2-3		
Non-IEEE Compliant Mode	JT1: Pins 1-2	1 K	Compliant and non-compliant operation (default)
	JT1: Pins 2-3		Inhibits non-compliant operation
Manual MDIX Setting	JT39: Pins 1-2	1 K	Straight Mode (default)
	JT2: Pins 2-3		Cross-over Mode
Auto MDIX Setting	JT4: Pins 1-2	1 K	Automatic Pair Swap – MDIX (default)
	JT4: Pins 2-3		Set to manual preset – Manual MDIX Setting (JT12)
Multiple Node Enable	JT3: Pins 1-2	1 K	Single node – NIC (default)
	JT3: Pins 2-3		Multiple node priority – switch/hub
Clock to MAC Enable	JT5: Pins 1-2	1 K	CLK_TO_MAC output enabled (default)
	JT5: Pins 2-3		CLK_TO_MAC output disabled

Functional Description



Table 13: VGA Pin Assignments

VGA Signal	FPGA Pin	VGA Signal	FPGA Pin
DAC_B2	G23	CLK_25.175MHz	P26
DAC_B3	G24		

Miscellaneous I/O

An 8-position DIP switch, 4 user Pushbuttons, and 8 user LEDs are provided on the Spartan-3A DSP Starter Platform. The connection of these devices to the FPGA is detailed in Table 14. The DIP switch is connected to FPGA Bank 0 and each switch is pulled low in the “OFF” position. Turning the switch “ON” causes the corresponding FPGA pin to be pulled to 2.5V or 3.3V, depending on the setting of the Bank 0 I/O voltage (VCCO_0). Like the DIP switch, the four user pushbuttons are also pulled low and depressing any button will cause the corresponding FPGA pin to be driven to the value of VCCO_0. Driving a “High” to the LEDs will cause them to light.

Table 14: Devices and Pin Assignments

Device	Name	FPGA Pin
DIP Switch	SW3.1	A7
	SW3.2	G16
	SW3.3	E9
	SW3.4	D15
	SW3.5	D19
	SW3.6	B24
	SW3.7	A5
	SW3.8	A23
Push Buttons	SW5 (SWITCH_PB1)	J17
	SW6 (SWITCH_PB2)	J15
	SW7 (SWITCH_PB3)	J13
	SW8 (SWITCH_PB4)	J10
LEDs	LED1 (D14)	P18
	LED2 (D13)	P25
	LED3 (D12)	N19
	LED4 (D11)	K22
	LED5 (D10)	H20
	LED6 (D9)	G21
	LED7 (D8)	D24
	LED8 (D7)	D25

Table 20: Clock Sources

Clock Source	FPGA Pin No.	Part Number
125 MHz oscillator (U7)	F13	Fox FXO-HC535-125.000
25.175 MHz oscillator (U4)	P26	Fox FXO-HC530-25.175
Socket	AE13	Populate with Fox 350LF-type oscillator
SMA connector J1	K14	Tyco-AMP part #221789-3

The SMA connector is AC-terminated through a 0.1uF 0402 capacitor. Between J1 and the capacitor is a 0-ohm 0402 resistor. Between the capacitor and FPGA pin K14 are two 49.9-ohm 0402 resistors, one pulling high to the Bank 0 I/O voltage (VCCO_0) and the other pulling low to GND. The user may alter this configuration to suit his requirements.

PCB Stackup

Figure 12 shows the 12-layer stackup of the Spartan-3A DSP Starter Platform Printed Circuit Board (PCB). The PCB substrate is FR4-class epoxy glass with 0.5 ounce (oz) copper used for all layers.

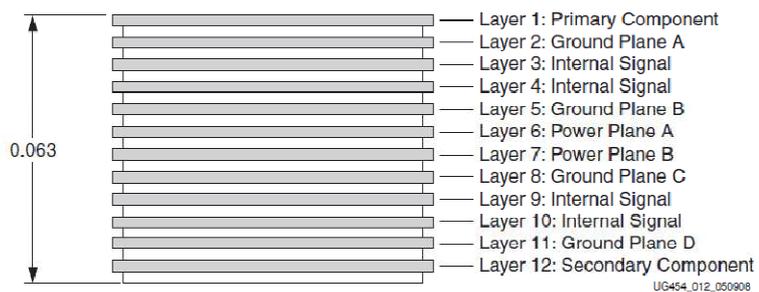


Figure 12: PCB Stackup

Related Resources

Xilinx would like to acknowledge the following key partners for their key contributions to this project.

For more information about this product, reference designs, and additional documentation, please visit the product home page found at www.xilinx.com/s3adspstarter or www.xilinx.com/s3adspmb or www.xilinx.com/s3adsp_sk.

Avnet Electronics Marketing

www.em.avnet.com/xilinx
www.em.avnet.com/drc

NS DP83865 GIG PHYTER



October 2004

DP83865 Gig PHYTER[®] V 10/100/1000 Ethernet Physical Layer

General Description

The DP83865 is a fully featured Physical Layer transceiver with integrated PMD sublayers to support 10BASE-T, 100BASE-TX and 1000BASE-T Ethernet protocols.

The DP83865 is an ultra low power version of the DP83861 and DP83891. It uses advanced 0.18 um, 1.8 V CMOS technology, fabricated at National Semiconductor's South Portland, Maine facility.

The DP83865 is designed for easy implementation of 10/100/1000 Mb/s Ethernet LANs. It interfaces directly to Twisted Pair media via an external transformer. This device interfaces directly to the MAC layer through the IEEE 802.3u Standard Media Independent Interface (MII), the IEEE 802.3z Gigabit Media Independent Interface (GMII), or Reduced GMII (RGMII).

The DP83865 is a fourth generation Gigabit PHY with field proven architecture and performance. Its robust performance ensures drop-in replacement of existing 10/100 Mbps equipment with ten to one hundred times the performance using the existing networking infrastructure.

Applications

The DP83865 fits applications in:

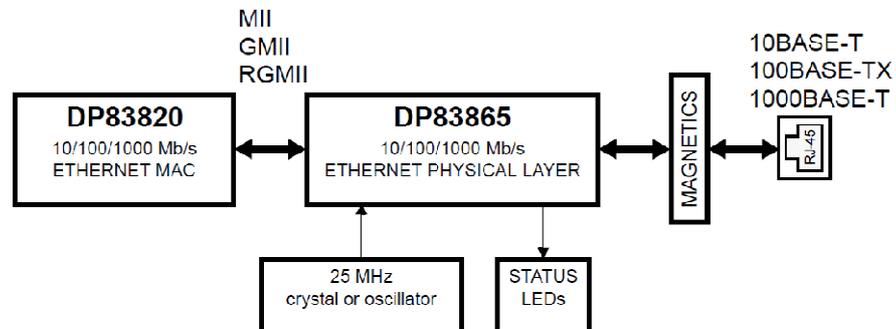
- 10/100/1000 Mb/s capable node cards
- Switches with 10/100/1000 Mb/s capable ports
- High speed uplink ports (backbone)

Features

- Ultra low power consumption typically 1.1 watt
- Fully compliant with IEEE 802.3 10BASE-T, 100BASE-TX and 1000BASE-T specifications

- Integrated PMD sublayer featuring adaptive equalization and baseline wander compensation according to ANSI X3.T12
- 3.3 V or 2.5 V MAC interfaces:
- IEEE 802.3u MII
- IEEE 802.3z GMII
- RGMII version 1.3
- User programmable GMII pin ordering
- IEEE 802.3u Auto-Negotiation and Parallel Detection
- Fully Auto-Negotiates between 1000 Mb/s, 100 Mb/s, and 10 Mb/s full duplex and half duplex devices
- Speed Fallback mode to achieve quality link
- Cable length estimator
- LED support for activity, full / half duplex, link1000, link100 and link10, user programmable (manual on/off), or reduced LED mode
- Supports 25 MHz operation with crystal or oscillator.
- Requires only two power supplies, 1.8 V (core and analog) and 2.5 V (analog and I/O). 3.3V is supported as an alternative supply for I/O voltage
- User programmable interrupt
- Supports Auto-MDIX at 10, 100 and 1000 Mb/s
- Supports JTAG (IEEE1149.1)
- 128-pin PQFP package (14mm x 20mm)

SYSTEM DIAGRAM



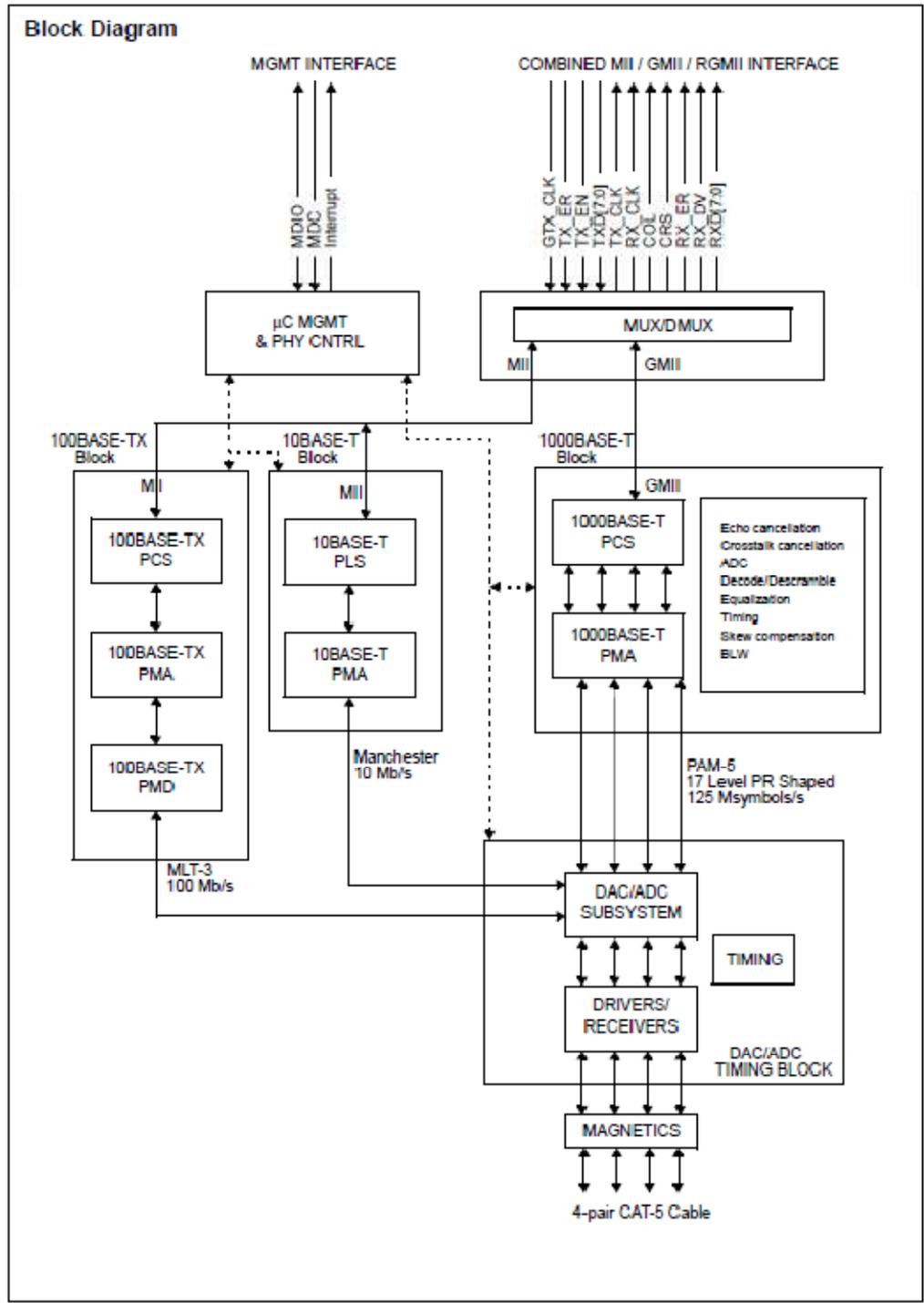
PHYTER[®] is a registered trademark of National Semiconductor Corporation

© 2004 National Semiconductor Corporation

www.national.com

DP83865 Gig PHYTER[®] V 10/100/1000 Ethernet Physical Layer

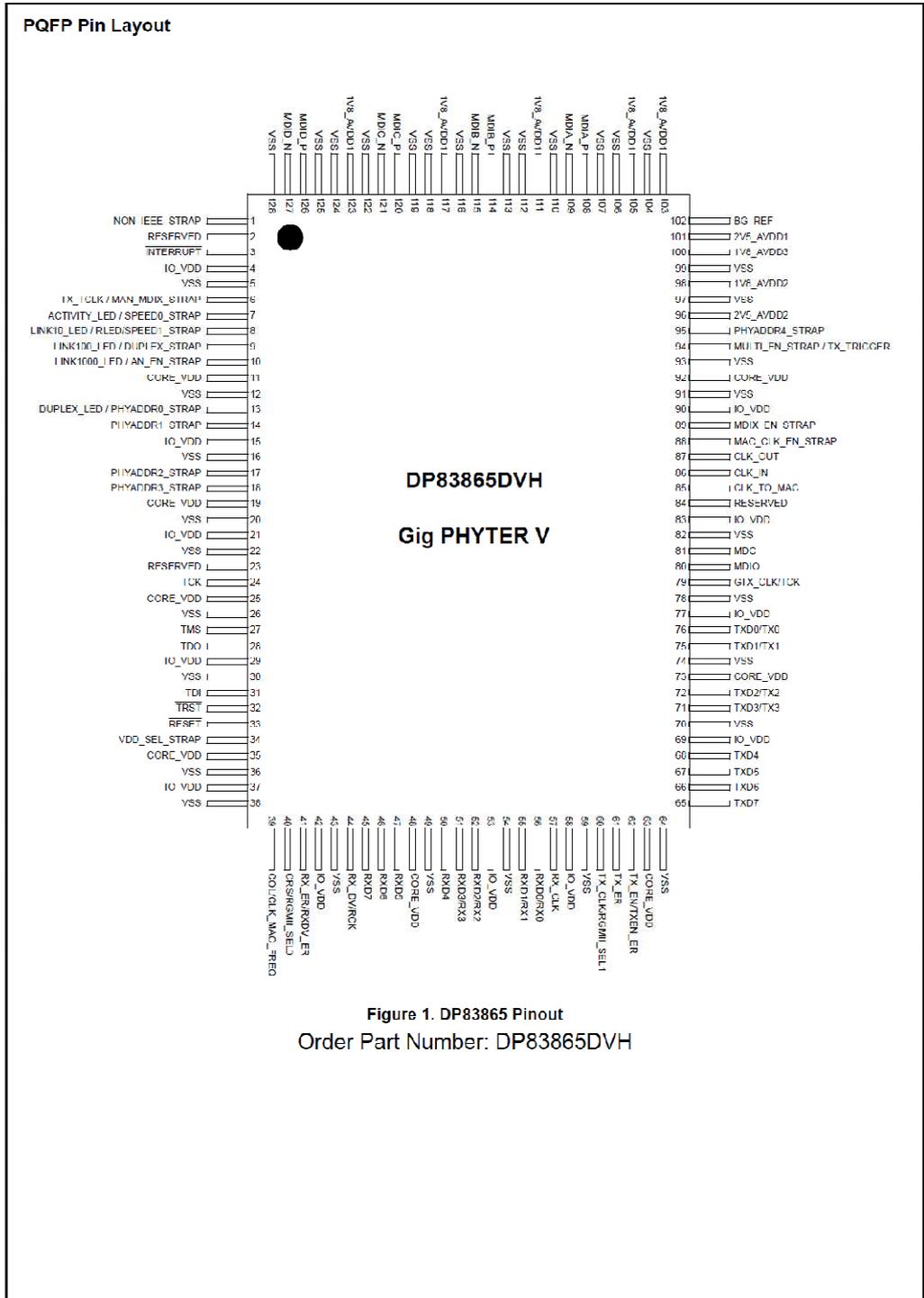
DP83865



www.national.com

2

DP83865



1.0 Pin Description

The DP83865 pins are classified into the following interface categories (each is described in the sections that follow):

- MAC Interfaces Type: I Inputs
- Management Interface Type: O Output
- Media Dependent Interface Type: O_Z Tristate Output
- JTAG Interface Type: I/O_Z Tristate Input_Output
- Clock Interface Type: S Strapping Pin
- Device Configuration and LED Interface Type: PU Internal Pull-up
- Reset Type: PD Internal Pull-down
- Power and Ground Pins
- Special Connect Pins

1.1 MAC Interfaces (MII, GMII, and RGMII)

Signal Name	Type	PQFP Pin #	Description															
CRS/RGMII_SEL0	O_Z, S, PD	40	<p>CARRIER SENSE or RGMII SELECT: CRS is asserted high to indicate the presence of a carrier due to receive or transmit activity in Half Duplex mode. For 10BASE-T and 100BASE-TX Full Duplex operation CRS is asserted when a received packet is detected. This signal is not defined for 1000BASE T Full Duplex mode.</p> <p>In RGMII mode, the CRS is not used. This pin can be used as a RGMII strapping selection pin.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>RGMII_SEL1</th> <th>RGMII_SEL0</th> <th>MAC Interface</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>= GMII</td> </tr> <tr> <td>0</td> <td>1</td> <td>= GMII</td> </tr> <tr> <td>1</td> <td>0</td> <td>= RGMII - HP</td> </tr> <tr> <td>1</td> <td>1</td> <td>= RGMII - 3COM</td> </tr> </tbody> </table>	RGMII_SEL1	RGMII_SEL0	MAC Interface	0	0	= GMII	0	1	= GMII	1	0	= RGMII - HP	1	1	= RGMII - 3COM
RGMII_SEL1	RGMII_SEL0	MAC Interface																
0	0	= GMII																
0	1	= GMII																
1	0	= RGMII - HP																
1	1	= RGMII - 3COM																
COL/CLK_MAC_FREQ	O_Z, S, PD	39	<p>COLLISION DETECT: Asserted high to indicate detection of a collision condition (assertion of CRS due to simultaneous transmit and receive activity) in Half Duplex modes. This signal is not synchronous to either MII clock (GTX_CLK, TX_CLK or RX_CLK). This signal is not defined and stays low for Full Duplex modes.</p> <p>CLOCK TO MAC FREQUENCY Select: 1 = CLOCK TO MAC output is 125 MHz 0 = CLOCK TO MAC output is 25 MHz</p>															
TX_CLK/RGMII_SEL1	O_Z, S, PD	60	<p>TRANSMIT CLOCK or RGMII SELECT: TX_CLK is a continuous clock signal generated from reference CLK_IN and driven by the PHY during 10 Mbps or 100 Mbps MII mode. TX_CLK clocks the data or error out of the MAC layer and into the PHY.</p> <p>The TX_CLK clock frequency is 2.5 MHz in 10BASE-T and 25 MHz in 100BASE-TX mode.</p> <p>Note: "TX_CLK" should not be confused with the "TX_TCLK" signal.</p> <p>In RGMII mode, the TX_CLK is not used. This pin can be used as a RGMII strapping selection pin. This pin should be pulled high for RGMII interface.</p>															

DP83865

1.0 Pin Description (Continued)			
Signal Name	Type	PQFP Pin #	Description
TXD0/TX0	I	76	TRANSMIT DATA: These signals carry 4B data nibbles (TXD[3:0]) during 10 Mbps and 100 Mbps MII mode, 4-bit data (TX[3:0]) in RGMII mode, and 8-bit data (TXD[7:0]) in 1000 Mbps GMII mode. They are synchronous to the transmit clocks (TX_CLK, TCK, GTX_CLK). Transmit data is input to PHY. In MII or GMII mode, the transmit data is enabled by TX_EN. In RGMII mode, the transmit data is enabled by TXEN_ER.
TXD1/TX1		75	
TXD2/TX2		72	
TXD3/TX3		71	
TXD4		68	
TXD5		67	
TXD6		66	
TXD7		65	
TX_EN/TXEN_ER	I	62	TRANSMIT ENABLE or TRANSMIT ENABLE/ERROR: In MII or GMII mode, it is an active high input sourced from MAC layer to indicate transmission data is available on the TXD. In RGMII mode, it combines the transmit enable and the transmit error signals of GMII mode using both clock edges.
GTX_CLK/TCK	I	79	GMII and RGMII TRANSMIT CLOCK: This continuous clock signal is sourced from the MAC layer to the PHY. Nominal frequency is 125 MHz.
TX_ER	I	61	TRANSMIT ERROR: It is an active high input used in MII mode and GMII mode forcing the PHY to transmit invalid symbols. The TX_ER signal is synchronous to the transmit clocks (TX_CLK or GTX_CLK). In MII 4B nibble mode, assertion of Transmit Error by the controller causes the PHY to issue invalid symbols followed by Halt (H) symbols until deassertion occurs. In GMII mode, assertion causes the PHY to emit one or more code-groups that are invalid data or delimiter in the transmitted frame. This signal is not used in the RGMII mode.
RX_CLK	O_Z	57	RECEIVE CLOCK: Provides the recovered receive clocks for different modes of operation: 2.5 MHz in 10 Mbps mode. 25 MHz in 100 Mbps mode. 125 MHz in 1000 Mps GMII mode. This pin is not used in the RGMII mode.
RXD0/RX0	O_Z	56	RECEIVE DATA: These signals carry 4-bit data nibbles (RXD[3:0]) during 10 Mbps and 100 Mbps MII mode and 8-bit data bytes (RXD[7:0]) in 1000 Mbps GMII mode. RXD is synchronous to the receive clock (RX_CLK). Receive data is sourced from the PHY to the MAC layer. Receive data RX[3:0] is used in RGMII mode. The data is synchronous to the RGMII receive clock (RCK). The receive data available (RXDV_EN) indicates valid received data to the MAC layer.
RXD1/RX1		55	
RXD2/RX2		52	
RXD3/RX3		51	
RXD4		50	
RXD5		47	
RXD6		46	
RXD7	45		
RX_ER/RXDV_ER	O_Z	41	RECEIVE ERROR or RECEIVE DATA AVAILABLE/ERROR: In 10 Mbps, 100 Mbps and 1000 Mbps mode this active high output indicates that the PHY has detected a Receive Error. The RX_ER signal is synchronous with the receive clock (RX_CLK). In RGMII mode, the receive data available and receive error is combined (RXDV_ER) using both rising and falling edges of the receive clock (RCK).
RX_DV/RCK	O_Z	44	RECEIVE DATA VALID or RECEIVE CLOCK: In MII and GMII modes, it is asserted high to indicate that valid data is present on the corresponding RXD[3:0] in MII mode and RXD[7:0] in GMII mode. In RGMII mode, this pin is the recovered receive clock (125MHz).

www.national.com

6

DP83865

1.0 Pin Description (Continued)

Signal Name	Type	PQFP Pin #	Description
TCK	I	24	TEST CLOCK: IEEE 1149.1 Test Clock input, primary clock source for all test logic input and output controlled by the testing entity. This pin should be left floating if not used.

1.5 Clock Interface

Signal Name	Type	PQFP Pin #	Description
CLK_IN	I	86	CLOCK INPUT: 25 MHz oscillator or crystal input (50 ppm).
CLK_OUT	O	87	CLOCK OUTPUT: Second terminal for 25 MHz crystal. Must be left floating if a clock oscillator is used.
CLK_TO_MAC	O	85	CLOCK TO MAC OUTPUT: This clock output can be used to drive the clock input of a MAC or switch device. This output is available after power-up and is active during all modes except during hardware or software reset. Note that the clock frequency is selectable through CLK_MAC_FREQ between 25 MHz and 125 MHz. To disable this clock output the MAC_CLK_EN_STRAP pin has to be tied low.

1.6 Device Configuration and LED Interface

(See section "3.7 PHY Address, Strapping Options and LEDs" on page 45 and section "5.9 LED/Strapping Option" on page 67.)

Signal Name	Type	PQFP Pin #	Description
NON_IEEE_STRAP	I/O, S, PD	1	NON IEEE COMPLIANT MODE ENABLE: This mode allows interoperability with certain non IEEE compliant 1000BASE-T transceivers. '1' enables IEEE compliant operation and non-compliant operation '0' enables IEEE compliant operation but inhibits non-compliant operation Note: The status of this bit is reflected in bit 10 of register 0x10. This pin also sets the default for and can be overwritten by bit 9 of register 0x12.
MAN_MDIX_STRAP / TX_TCLK	I/O, S, PD	6	MANUAL MDIX SETTING: This pin sets the default for manual MDI/MDIX configuration. '1' PHY is manually set to cross-over mode (MDIX) '0' PHY is manually set to straight mode (MDI) Note: The status of this bit is reflected in bit 8 of register 0x10. This pin also sets the default for and can be overwritten by bit 14 of register 0x12. TX_TCLK: TX_TCLK is enabled by setting bit 7 of register 0x12. It is used to measure jitter in Test Modes 2 and 3 as described in IEEE 802.3ab specification. TX_TCLK should not be confused with the TX_CLK signal. See Table 12 on page 29 regarding Test Mode setting. This pin should be left floating if not used.

1.0 Pin Description (Continued)																																	
Signal Name	Type	PQFP Pin #	Description																														
ACTIVITY_LED / SPEED0_STRAP	I/O, S, PD	7	<p>SPEED SELECT STRAP: These strap option pins have 2 different functions depending on whether Auto-Negotiation is enabled or not.</p> <p><u>Auto-Neg disabled:</u></p> <table border="1"> <thead> <tr> <th>Speed[1]</th> <th>Speed[0]</th> <th>Speed Enabled</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>= Reserved</td> </tr> <tr> <td>1</td> <td>0</td> <td>= 1000BASE-T</td> </tr> <tr> <td>0</td> <td>1</td> <td>= 100BASE-TX</td> </tr> <tr> <td>0</td> <td>0</td> <td>= 10BASE-T</td> </tr> </tbody> </table> <p><u>Auto-Neg enabled (Advertised capability):</u></p> <table border="1"> <thead> <tr> <th>Speed[1]</th> <th>Speed[0]</th> <th>Speed Enabled</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>1</td> <td>= 1000BASE-T, 10BASE-T</td> </tr> <tr> <td>1</td> <td>0</td> <td>= 1000BASE-T</td> </tr> <tr> <td>0</td> <td>1</td> <td>= 1000BASE-T, 100BASE-TX</td> </tr> <tr> <td>0</td> <td>0</td> <td>= 1000BASE-T, 100BASE-TX, 10BASE-T</td> </tr> </tbody> </table> <p>Note: The status of this bit is reflected in register 0x10.12.</p> <p>ACTIVITY LED: The LED output indicates the occurrence of either idle error or packet transfer.</p>	Speed[1]	Speed[0]	Speed Enabled	1	1	= Reserved	1	0	= 1000BASE-T	0	1	= 100BASE-TX	0	0	= 10BASE-T	Speed[1]	Speed[0]	Speed Enabled	1	1	= 1000BASE-T, 10BASE-T	1	0	= 1000BASE-T	0	1	= 1000BASE-T, 100BASE-TX	0	0	= 1000BASE-T, 100BASE-TX, 10BASE-T
Speed[1]	Speed[0]	Speed Enabled																															
1	1	= Reserved																															
1	0	= 1000BASE-T																															
0	1	= 100BASE-TX																															
0	0	= 10BASE-T																															
Speed[1]	Speed[0]	Speed Enabled																															
1	1	= 1000BASE-T, 10BASE-T																															
1	0	= 1000BASE-T																															
0	1	= 1000BASE-T, 100BASE-TX																															
0	0	= 1000BASE-T, 100BASE-TX, 10BASE-T																															
LINK10_LED /RLED/ SPEED1_STRAP	I/O, S, PD	8	<p>SPEED SELECT STRAP: The strap option pins have 2 different functions depending on whether Auto-Neg is enabled or not. See SPEED0_STRAP for details.</p> <p>Note: The status of this bit is reflected in register 0x10.13.</p> <p>10M GOOD LINK LED: In the standard 5-LED display mode, this LED output indicates that the PHY has established a good link at 10 Mbps.</p> <p>RLED MODE: There are two reduced LED modes, the 3-in-1 and 4-in-1 modes. Each RLED mode is enabled in register 0x13.5 and 0x1A.0.</p> <ul style="list-style-type: none"> – 3-in-1: 10, 100, and 1000 Mbps good links are combined into one LED. – 4-in-1: 3-in-1 and activity are combined. <p>Note: LED steady on indicates good link and flashing indicates Tx/Rx activities.</p>																														
LINK100_LED / DUPLEX_STRAP	I/O, S, PU	9	<p>DUPLEX MODE: This pin sets the default value for the duplex mode. '1' enables Full Duplex by default, '0' enables Half Duplex only.</p> <p>Note: The status of this bit is reflected in bit 14 of register 0x10.</p> <p>100M SPEED AND GOOD LINK LED: The LED output indicates that the PHY has established a good link at 100 Mbps.</p> <p>In 100BASE-T mode, the link is established as a result of an input receive amplitude compliant with TP-PMD specifications which will result in internal generation of Signal Detect. LINK100_LED will assert after the internal Signal Detect has remained asserted for a minimum of 500 μs. LINK100_LED will de-assert immediately following the de-assertion of the internal Signal Detect.</p>																														
LINK1000_LED / AN_FN_STRAP	I/O, S, PU	10	<p>AUTO-NEGOTIATION ENABLE: Input to initialize Auto-Negotiation Enable bit (register 0 bit-12)</p> <p>'1' enables Auto-Neg and '0' disables Auto-Neg.</p> <p>Note: The status of this bit is reflected in bit 15 of register 0x10. This pin also sets the default for and can be overwritten by bit 12 of register 0x00.</p> <p>1000M SPEED AND GOOD LINK LED: The LED output indicates that the PHY has established a good link at 1000 Mbps.</p> <p>In 1000BASE-T mode, the link is established as a result of training. Auto-Negotiation completed, valid 1000BASE-T link established and reliable reception of signals transmitted from a remote PHY is received.</p>																														

DP83865

1.0 Pin Description (Continued)

Signal Name	Type	PQFP Pin #	Description
DUPLEX_LED / PHYADDR0_STRAP	I/O, S, PU	13	PHY ADDRESS [4:0]: The DP83865 provides five PHY address-sensing pins for multiple PHY applications. The setting on these five pins provides the base address of the PHY.
PHYADDR1_STRAP	PD	14	The five PHYAD[4:0] bits are registered as inputs at reset with PHYADDR4 being the MSB of the 5-bit PHY address. Note: The status of these bit is reflected in bits 4:0 of register 0x12.
PHYADDR2_STRAP	PD	17	
PHYADDR3_STRAP	PD	18	
PHYADDR4_STRAP	PD	95	
MULTI_EN_STRAP / TX_TRIGGER	I/O, S, PD	94	MULTIPLE NODE ENABLE: This pin determines if the PHY advertises Master (multiple nodes) or Slave (single node) priority during 1000BASE-T Auto-Negotiation. ‘1’ Selects multiple node priority (switch or hub) ‘0’ Selects single node priority (NIC) Note: The status of this bit is reflected in bit 5 of register 0x10. TX_TRIGGER: This output can be enabled during the IEEE 1000BASE-T test-modes. This signal is not required by IEEE to perform the tests, but will help to take measurements. TX_TRIGGER is only available in test modes 1 and 4 and provides a trigger to allow for viewing test waveforms on an oscilloscope.
MDIX_EN_STRAP	I/O, S, PU	89	AUTO MDIX ENABLE: This pin controls the automatic pair swap (Auto-MDIX) of the MDI/MDIX interface. ‘1’ enables pair swap mode ‘0’ disables the Auto-MDIX and defaults the part into the mode preset by the MAN_MDIX_STRAP pin. Note: The status of this bit is reflected in bit 6 of register 0x10. This pin also sets the default for and can be overwritten by bit 15 of register 0x12.
MAC_CLK_EN_STRAP / TX_SYN_CLK	I, S, PU	88	CLOCK TO MAC ENABLE: ‘1’ CLK_TO_MAC clock output enabled ‘0’ CLK_TO_MAC disabled Note: This status of this pin is reflected in bit 7 of register 0x10. TX_SYN_CLK: This output can be enabled during the IEEE 1000BASE-T test-modes. This signal is not required by IEEE to perform the tests, but will help to take measurements. TX_SYN_CLK is only available in test modes 1 and 4. TX_SYN_CLK = TX_TCLK / 4 in test mode 1 TX_SYN_CLK = TX_TCLK / 6 in test mode 4
VDD_SEL_STRAP	I/O, S	34	IO_VDD SELECT: This pin selects between 2.5V or 3.3V for I/O VDD . ‘1’ selects 3.3V mode ‘0’ selects 2.5V mode This pin must either be connected directly to ground or directly to a supply voltage (2.5V to 3.3V).

1.7 Reset

Signal Name	Type	PQFP Pin #	Description
RESET	I	33	RESET: The active low RESET input allows for hard-reset, soft-reset, and TRI-STATE output reset combinations. The RESET input must be low for a minimum of 150 μs.

Documento 2

Pliego de

condiciones

ÍNDICE GENERAL

<i>Pliego de condiciones generales y económicas</i>	<i>1</i>
1. Condiciones generales.....	1
2. Condiciones económicas	2
<i>Pliego de condiciones técnicas y particulares</i>	<i>3</i>
1. Equipo informático.....	3
2. Normas de calidad	3
3. Normas de seguridad e higiene	3
4. Vida útil del equipo.....	3
5. Otros criterios de diseño	4

PLIEGO DE CONDICIONES GENERALES Y

ECONÓMICAS

1. Condiciones generales

Las condiciones y cláusulas que se establecen en este documento son de obligado cumplimiento por las partes contratantes.

- I. Tanto el administrador como el cliente se comprometen desde la fecha de la firma del contrato a llevar a cabo lo que se estipule.
- II. Ante cualquier reclamación o discrepancia en lo concerniente al cumplimiento de lo pactado por cualquiera de las partes, una vez agotada toda vía de entendimiento, se tramitará el asunto por la vía de lo legal. El dictamen o sentencia que se dicte será de obligado cumplimiento para las dos partes.
- III. Al firmarse el contrato, el suministrador se compromete a facilitar toda la información necesaria para la instalación y buen funcionamiento del equipo, siempre que sea requerido para ello.
- IV. Asimismo, el cliente entregará al suministrador todas las características distintivas del equipo comprado y aquellas otras que considere oportunas para el necesario conocimiento de la misma a efectos del diseño del presente equipo.
- V. El plazo de entrega será de nueve meses, a partir de la fecha de la firma del contrato, pudiendo ampliarse en tres meses. Cualquier modificación de los plazos deberá contar con el acuerdo de las dos partes.
- VI. En caso de retrasos imputables al suministrador, se considerará una indemnización del 1 % del valor estipulado por semana de retraso.
- VII. Existirá un plazo de garantía de un año a partir de la entrega del sistema. Dicha garantía quedará sin efecto si se demostrase

que el sistema ha estado sometido a manipulación o uso indebido.

- VIII. Cumplido dicho plazo de garantía, el suministrador queda obligado a la reparación del sistema durante un plazo de cinco años, fuera del cual quedará a su propio criterio atender la petición del cliente.
- IX. En ningún momento tendrá el suministrador obligación alguna frente a desperfectos o averías por uso indebido por personas no autorizadas por el suministrador.

2. Condiciones económicas

Las condiciones económicas a tener en cuenta son:

- I. Los precios indicados en este proyecto son firmes y sin revisión por ningún concepto, siempre y cuando se acepten dentro del periodo de validez del presupuesto que se fija para el año 2009.
- II. El pago se realizará como sigue:
- a. 75% a la firma del contrato.
 - b. 25% en el momento de entrega.
- III. La forma de pago será al contado mediante cheque nominativo o mediante transferencia bancaria. En ningún caso se aceptarán letras de cambio.
- IV. El suministrador se hará cargo de los gastos de embalaje y del transporte, dentro de la ciudad donde se encuentre la instalación. En caso de ser necesario transporte interurbano, el gasto correrá por cuenta del cliente. En todo caso, el responsable de posibles desperfectos ocasionados por el transporte será el suministrador.
- V. Durante el plazo de garantía, los gastos originados por las reparaciones correrán por cuenta del suministrador.
- VI. Fuera de dicho plazo y durante los siguientes cinco años, los costes serán fijados mediante acuerdo por ambas partes. Pasados 5 años, éstos los fijará exclusivamente el suministrador

PLIEGO DE CONDICIONES TÉCNICAS Y PARTICULARES

1. Equipo informático

El equipo sobre el que se implemente el diseño habrá de ser un FPGA; éste habrá de contar con las siguientes características:

- I. El equipo informático debe estar homologado conforme a la normativa Española y Europea vigente.
- II. El código debe instalarse conforme a las indicaciones del fabricante, manteniendo las condiciones de humedad y temperatura entre los límites marcados.
- III. Los programas informáticos empleados han de contar con la licencia preceptiva y cumplir con las condiciones de la misma. En caso de usar programas de licencia GNU, se deberán respetar las condiciones de la misma.

2. Normas de calidad

Los sistemas se diseñarán de forma que cumplan las normas UNE CEI y EN aplicables a este tipo de productos.

3. Normas de seguridad e higiene

El proyecto cumplirá con la Ley 31/95 de Prevención de Riesgos Laborales.

4. Vida útil del equipo

Los sistemas se diseñarán para una vida útil en uso continuo no inferior a diez años.

5. Otros criterios de diseño

- I. Se emplearán componentes normalizados para los circuitos electrónicos considerando los márgenes de seguridad necesarios para trabajar en entornos de alta radiación electromagnética.
- II. Todos los protocolos de comunicación, modelos de datos y puertos de conexión implantados deberán cumplir las especificaciones indicadas en la norma IEC-61850.

Documento 3

Presupuesto

ÍNDICE GENERAL

<i>Capítulo 1.</i>	<i>Mediciones.....</i>	<i>1</i>
<i>Capítulo 2.</i>	<i>Precios unitarios.....</i>	<i>3</i>
<i>Capítulo 3.</i>	<i>Sumas parciales</i>	<i>4</i>
<i>Capítulo 4.</i>	<i>Presupuesto general.....</i>	<i>5</i>

ÍNDICE DE TABLAS

TABLA 1-1. MEDICIONES DE LOS RECURSOS HUMANOS.....	1
TABLA 1-2. MEDICIONES DE LOS MEDIOS MATERIALES.	2
TABLA 2-1. PRECIO UNITARIO DE LOS RECURSOS HUMANOS.....	3
TABLA 2-2. PRECIO UNITARIO DE LOS RECURSOS MATERIALES.....	3
TABLA 3-1. SUMAS PARCIALES DE LOS RECURSOS HUMANOS.....	4
TABLA 3-2. SUMAS PARCIALES DE LOS RECURSOS MATERIALES.....	4
TABLA 3-3. ESTIMACIÓN DE LOS GASTOS INDIRECTOS.....	4
TABLA 4-1. IMPORTE TOTAL DEL PRESUPUESTO.....	5

Capítulo 1. MEDICIONES

En este documento se recogen los costes de las diferentes partes necesarias para el desarrollo del proyecto, agrupados en distintas partidas, definiendo tanto los presupuestos de cada una de ellas como el presupuesto total.

A la hora de detallar los conceptos que serán incluidos en el presupuesto final correspondiente al presente proyecto, se han seguido las premisas que se exponen a continuación:

- Los precios de los componentes detallados corresponden al importe pagado en su fecha de compra, y pueden no coincidir con el importe de compra en caso de requerirse una reproducción del proyecto, en cuyo caso el presente presupuesto podrá ser revisado y actualizado.
- Se incluyen los costes correspondientes al equipo informático y al software utilizado en el desarrollo del proyecto.
- El presupuesto final incluye la totalidad de los componentes empleados en el proyecto que constituye el concepto global desarrollado, pero la mano de obra incluida se corresponde únicamente con la empleada por el proyectista encargado de la parte del concepto global desarrollada en el presente proyecto.

El primer paso para realizar el presupuesto del proyecto es detallar las cantidades empleadas de los diferentes medios necesarios.

Las partidas correspondientes a Recursos Humanos se recogen a continuación, en la Tabla 1-1.

<i>Concepto</i>	<i>Número de horas</i>
Estudio y auditoría	75
Ingeniería	400
Elaboración de documentación	50

Tabla 1-1. Mediciones de los Recursos Humanos.

En la Tabla 1-2 se hace referencia a los elementos y materiales empleados en la composición del proyecto, indicando las unidades necesarias de cada uno de ellos.

<i>Concepto</i>	<i>Cantidad</i>
PC Dell	1 unidad
	Intel Pentium 4 CPU 3GHz.
	RAM 300GB.
	Microsoft Windows XP Professional Versión 2002.
	Adaptador de red Broadcom NetXtreme 57xx Gigabit Controller.
	Microsoft Office 2003.
Xilinx ISE Design Suite 10.1	1 licencia
Spartan 3A-DSP 1800A	1 unidad
Xilinx Platform Cable USB II	1 unidad
Cable Ethernet cruzado	1 unidad

Tabla 1-2. Mediciones de los medios materiales.

Capítulo 2. PRECIOS UNITARIOS

Una vez conocidos cuáles son los recursos que se han empleado en el desarrollo del proyecto, es necesario determinar qué coste económico supone cada uno de ellos de manera unitaria. La información se presenta en forma de tablas siguiendo la estructura anterior.

En primer lugar se detallarán los precios unitarios de las partidas correspondientes a los Recursos Humanos. La Tabla 2-1 muestra dichos valores.

<i>Concepto</i>	<i>Precio en €/hora</i>
Estudio y auditoría	45
Ingeniería	30
Elaboración de documentación	20

Tabla 2-1. Precio unitario de los Recursos Humanos.

Respecto a los Recursos Materiales empleados, los costes unitarios de cada elemento se recogen en la siguiente Tabla:

<i>Concepto</i>	<i>Precio en €/Ud.</i>
PC Dell	Amortización correspondiente al periodo: 1200€, 5 años. 240
Xilinx ISE Design Suite 10.1	Licencia Industrial 1 año. 4000
Spartan 3A-DSP 1800A	275
Xilinx Platform Cable USB II	350
Cable Ethernet cruzado	12

Tabla 2-2. Precio unitario de los Recursos Materiales.

Capítulo 3. SUMAS PARCIALES

Una vez conocidas las cantidades de cada recurso, y sus precios unitarios, quedan por establecer los costes correspondientes a cada una de las partidas.

Las sumas parciales de los Recursos Humanos empleados se muestran en la Tabla 3-1.

<i>Concepto</i>	<i>Número de horas</i>	<i>Precio en €/hora</i>	<i>Total (€)</i>
Estudio y auditoría	75	45	3375
Ingeniería	400	30	12000
Elaboración de documentación	50	20	1000
TOTAL RECURSOS HUMANOS			16375

Tabla 3-1. Sumas parciales de los Recursos Humanos.

La partida de los Recursos Materiales presenta los siguientes valores:

<i>Concepto</i>	<i>Unidades</i>	<i>Precio en €/Ud.</i>	<i>Total (€)</i>
PC Dell	1 unidad	240	240
Xilinx ISE Design Suite 10.1	1 licencia	4000	4000
Spartan 3A-DSP 1800A	1 licencia	275	275
Xilinx Platform Cable USB II	1 unidad	350	350
Cable Ethernet cruzado	1 unidad	12	12
TOTAL RECURSOS MATERIALES			4877

Tabla 3-2. Sumas parciales de los Recursos Materiales.

Una última partida no indicada anteriormente pero necesaria para el desarrollo del proyecto es la que refleja los Gastos Indirectos ocasionados. La Tabla 3-3 muestra el coste correspondiente.

<i>Concepto</i>	<i>Total (€)</i>
Consumo eléctrico	
Alquiler de oficina y laboratorio	
Conexión telefónica	
Línea ADSL	
Papelería	
ESTIMACIÓN GASTOS INDIRECTOS	4250

Tabla 3-3. Estimación de los Gastos Indirectos.

Capítulo 4. PRESUPUESTO GENERAL

El último paso para completar el presupuesto del proyecto es realizar el monto de todas las partidas descritas en los Capítulos anteriores de este documento.

En la Tabla 4-1 se recogen los valores totales de cada tipo de recurso y el Presupuesto General resultado de la suma de todos ellos.

<i>Partida</i>	<i>Importe (€)</i>
Total Recursos Humanos	16375
Total Recursos Materiales	4877
Total Gastos Indirectos	4250
Total de presupuesto	25502
16% IVA	4080
TOTAL PRESUPUESTO GENERAL	29582

Tabla 4-1. Importe total del presupuesto.