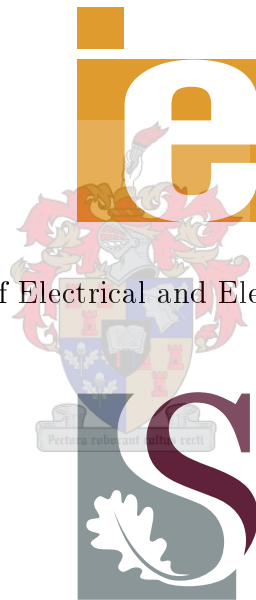


Satellite Data Transmission by means of a Multi-Channel System

David Smith

Department of Electrical and Electronic Engineering



University of Stellenbosch

Thesis presented in partial fulfilment of the requirements for the
degree of Master of Engineering at the University of Stellenbosch

Supervisor: Prof. J.B. de Swardt

December 2006

Declaration

I, the undersigned, hereby declare that the work contained in this thesis is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature _____

Date _____



Abstract

The aim of this thesis is to find an affordable and upgradeable manner of increasing the data rate of current satellite systems. The compression, modulation and amplification techniques and telecommunication regulations restrict the data rate, which are bypassed by designing a parallel channel configuration.

In order to test this solution a system is developed that is based on the existing standards of DVB-S and MPEG-2. The combination of these standards protect the data from interference, package the data for ease of storage, modulate the data to more effectively radiate the signal and shape the spectrum to adhere to telecommunication regulations.

The spacing between the channels is reduced and the transmitted signal is amplified to inject interference into the system. The effect of this interference on the receiver is shown by transmitting image data and comparing the received image with the original.

A faint watermark of a university crest is centered on the page. The crest features a shield with various symbols, topped by a crown and surrounded by decorative flourishes. Below the shield is a banner with the Latin motto "Pectora roburant cultus recti".

Pectora roburant cultus recti

Opsomming

Hierdie tesis poog om 'n bekostigbare en opgradeerbare metode te vind om die datatransmissietempo van 'n bestaande satellietstelsel te verbeter. Die kompressie-, modulatie- en versterkingstegnieke en telekommunikasie regulasies beperk die datatempo, wat omseil word deur 'n parallelle kanaal konfigurasie te ontwerp.

Om hierdie oplossing te toets, is 'n stelsel gebou wat gebaseer is op die bestaande standaarde van DVB-S en MPEG-2. Die kombinasie van hierdie standaarde beskerm die data teen steuring, verpak die data vir maklike berging, moduleer die data meer vir verbeterde uitstraling van die sein en vorm die spektrum volgens die vereistes van telekommunikasie regulasies.

Die spasiëring tussen kanale is verminder en die uitgaande sein is versterk om steuring binne die stelsel te veroorsaak. Die effek van steuring op die ontvanger word getoon deur beelddata te versend, en die ontvangde beeld te vergelyk met die oorspronklike.



Pectora roborant cultus recti

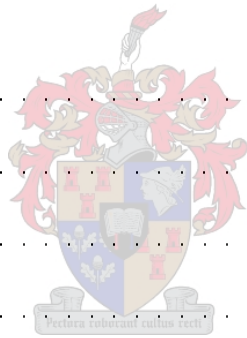
Acknowledgements

I would like to thank the following people and institutes for their support:

- My Mom, for being a mother
- The Ashcroft Trust, for all their monetary support throughout my tertiary education
- ETSE, for the use of their equipment to test and complete the system
- Graham Hardie, for being my proof reader and photographer
- Jacques Malan, for the use of his board
- Jacques Roux, for all the late nights helping me test the system
- Johan Arendse, for soldering the components together
- Johan de Swardt, for being a supportive supervisor
- Mark Thomas, for being my proof reader
- Marlize Schoeman, for the use of her L^AT_EX thesis layout
- SunSpace, for their monetary support and equipment to test the system
- Tinus Stander, for the Opsomming of my Abstract
- The University of Stellenbosch, for being my Alma Mater

Contents

List of Tables	xii
List of Figures	xiv
Abbreviations	xx
1 Introduction	1
1.1 Problem Statement	2
1.2 Proposed Solution	3
1.3 Implementation	4
1.3.1 System	5
1.3.2 MPEG-2	6
1.3.3 DVB-S	6
1.3.4 LVDS	6
1.4 Thesis Outline	7
2 MPEG-2	8
2.1 Overview	8
2.2 Elementary Stream	8
2.2.1 Video Compression	9
2.2.2 Audio Compression	9
2.2.3 Implementation	9

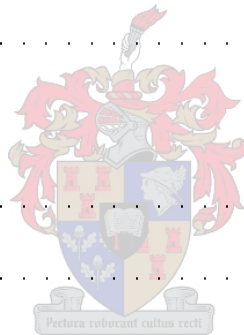


2.3	Packetised Elementary Stream	10
2.3.1	Implementation	10
2.4	Program Association Table	10
2.4.1	PAT-Section Header	10
2.4.2	PAT-Section Payload	11
2.5	Program Map Table	11
2.5.1	PMT-Section Header	12
2.5.2	PMT-Section Payload	12
2.6	Conditional Access Table	13
2.7	Network Information Table	13
2.8	Transport Stream	14
2.8.1	TS-Packet Header	14
2.8.2	TS-Packet Adaptation Field	15
2.8.3	TS-Packet Payload	15
2.9	Implementation	16
2.10	Summary	16
3	DVB-S	17
3.1	Transport Multiplex Adaptation	17
3.2	Energy Dispersal	17
3.2.1	Spectra of PSK Digital Signals	18
3.2.2	Implementation	18
3.3	Reed-Solomon Coding	19
3.3.1	Galois Fields	20
3.3.2	Reed-Solomon Encoding	22

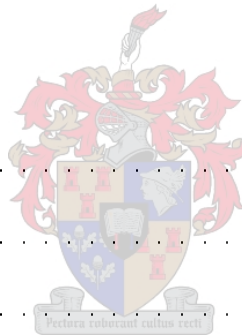
3.3.3	Reed-Solomon Decoding	23
3.3.4	Implementation	25
3.4	Interleaving	25
3.4.1	Convolutional Interleaving	25
3.4.2	Implementation	26
3.5	Convolutional Coding	27
3.5.1	Convolutional Encoding	27
3.5.2	Convolutional Decoding	28
3.5.3	Implementation	30
3.6	Puncturing	31
3.6.1	Implementation	31
3.7	Baseband Shaping for Modulation	31
3.7.1	Implementation	31
3.8	Phase Shift Keying Modulation	32
3.8.1	Implementation	32
3.9	Summary	32
4	LVDS	33
4.1	LVDS	33
4.1.1	Configuration	33
4.1.2	Functional Table	34
4.1.3	Electrical Characteristics	34
4.1.4	Implementation	35
4.2	DVB-SPI	35
4.2.1	Signal Format	35

4.2.2	Mechanical Details	36
4.2.3	Implementation	36
4.3	Summary	36
5	Audio and Video	37
5.1	Video	37
5.1.1	VGA Screen	37
5.2	Audio	39
5.2.1	Control Interface	39
5.2.2	Digital Audio Interface	40
5.3	Summary	40
6	Memory	41
6.1	Memory Initialisation File Format	41
6.1.1	Bitmap File Format	41
6.1.2	Wave File Format	42
6.1.3	Hexadecimal Object File Format	42
6.2	Internal Memory	43
6.2.1	ROM Mode	43
6.2.2	FIFO Buffer Mode	43
6.3	External Memory	44
6.3.1	Write Process	44
6.3.2	Read Process	44
6.3.3	Synchronisation	45
6.4	Summary	46
7	FPGA Implementation	47

7.1	Controllers	47
7.1.1	System Timing Control	47
7.1.2	External Memory Timing Control	47
7.1.3	User Input Control	48
7.1.4	User Output Control	48
7.2	MPEG-2/DVB-S Synchronisation	49
7.2.1	MPEG-2 Initialisation	49
7.2.2	MPEG-2 Activation	49
7.2.3	DVB-S Initialisation	50
7.2.4	DVB-S Activation	50
7.3	Summary	50
8	Set-Up	51
8.1	Frequency Selection	51
8.2	Power Requirements	52
8.3	Connections	52
8.4	Software	54
8.4.1	Quartus Software Set-Up	54
8.4.2	Si2110-EVB Software Set-Up	55
8.5	Summary	55
9	Measurements	56
9.1	Time Domain Measurements	56
9.1.1	Signal Distortion	57
9.1.2	Signal Imbalance	61
9.1.3	Distortion and Imbalance Correction Measures	63



9.2	Frequency Domain Measurements	64
9.2.1	Local Oscillators	64
9.2.2	Signal Flow	65
9.3	Summary	72
10	Conclusion and Recommendations	73
10.1	Summary	73
10.2	Results	73
10.3	Recommendations	74
10.4	Conclusion	74
	Bibliography	75
A	FPGA Pin Assignments	80
A.1	User Interface	80
A.2	VGA Interface	81
A.3	LVDS Interface	82
A.4	External RAM Interface	82
B	Additional Boards	84
B.1	Modifications	84
B.1.1	Power Supply Modifications	84
B.1.2	Data Signal Modifications	85
B.2	Schematics	85
B.3	Bill of Materials	87
B.4	Board Layouts	88
C	FPGA Development Kit	92



C.1 FPGA Development Kit Contents 92

C.2 FPGA DevBoard Modifications 93

C.3 FPGA DevBoard Components 93

D DVB-S Receiver Evaluation Kit 95

D.1 DVB-S Receiver Evaluation Kit Contents 95

D.2 DVB-S Receiver EvBoard Components 95



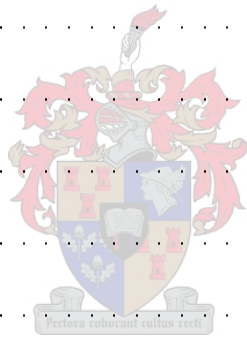
List of Tables

2.1	Program Number Assignments	11
2.2	PMT PID Assignments	11
2.3	ES Type Assignments	12
2.4	ES PID Assignments	12
2.5	PID Assignments	14
2.6	AF Control Assignments	14
3.1	Punctured Code Definition	31
4.1	LVDS Driver Functional Table	34
4.2	LVDS Receiver Functional Table	34
4.3	LVDS Driver Output Characteristics	34
4.4	LVDS Receiver Input Characteristics	34
4.5	DVB-SPI Pin Assignments	36
5.1	Audio Codec Register Assignment	39
7.1	FPGA Push-Button Command Table	48
7.2	FPGA DIP-Switch Command Table	48
7.3	MPEG-2 Decoder Error Indication	48
A.1	Timing Control Pin Assignments	80

A.2	Push-Buttons Pin Assignments	80
A.3	DIP-Switch Pin Assignments	80
A.4	LED Pin Assignments	81
A.5	LCD Pin Assignments	81
A.6	VGA Control Pin Assignments	81
A.7	VGA Colour Pin Assignments	81
A.8	LVDS Device Control Pin Assignments	82
A.9	LVDS Driver Pin Assignments	82
A.10	LVDS Receiver Pin Assignments	82
A.11	External RAM Clock Pin Assignments	82
A.12	External RAM Address Pin Assignments	83
A.13	External RAM Control Pin Assignments	83
A.14	External RAM Data Mask Pin Assignments	83
A.15	External RAM Data Strobe Pin Assignments	83
A.16	External RAM Data Bus Pin Assignments	83
B.1	Devices	87
B.2	Connectors	87
B.3	Capacitors	87
B.4	Resistors	87

List of Figures

1.1	Satellite System Functions	1
1.2	Simplified Satellite System	2
1.3	Proposed Solution	3
1.4	Frequency Band Limitation Bypass	3
1.5	Implemented System	4
1.6	System Block Diagram	4
1.7	System Implementation	5
1.8	MPEG-2 Block Diagram	6
1.9	DVB-S Block Diagram	6
1.10	LVDS Block Diagram	6
2.1	TS Format	8
2.2	Video Redundancy	9
2.3	Audio Masking	9
2.4	PES-Packet	10
2.5	PAT-Section	10
2.6	PAT-Section Header	10
2.7	PAT-Section Payload	11
2.8	PMT-Section	11



2.9	PMT-Section Header	12
2.10	PMT-Section Payload	12
2.11	Video Stream Descriptor	13
2.12	Audio Stream Descriptor	13
2.13	TS-Packet	14
2.14	TS-Packet Header	14
2.15	Continuity Counter	15
2.16	Adaptation Field	15
2.17	Conversion of PES and PSI to TS	15
2.18	Transmitter Implementation	16
2.19	Receiver Implementation	16
3.1	TS-Packet	17
3.2	Energy Scrambler Diagram	18
3.3	Randomised Transport Packets	19
3.4	Reed-Solomon Codeword	19
3.5	Retrievable Codeword	20
3.6	Irretrievable Codeword	20
3.7	Reed Solomon Error Protected Packet	25
3.8	B -by- N Convolutional Interleaver	25
3.9	Interleaved Frames	26
3.10	Encoder Implementation	26
3.11	Convolutional Encoder	27
3.12	State Diagram	27
3.13	Trellis Diagram of Possible States	27

3.14	Trellis Diagram	28
3.15	Trellis Diagram of Possible Paths	28
3.16	Step One	29
3.17	Step Two	29
3.18	Step Three	30
3.19	Trellis Diagram of Viterbi Decoder	30
3.20	Convolutional Encoder	30
3.21	QPSK Constellation	32
4.1	LVDS Driver and Receiver Configuration	33
4.2	Synchronous Parallel Interface	35
4.3	Signal Format	35
5.1	Video DAC Block Diagram	37
5.2	CRT control	38
5.3	Horizontal Sweep	38
5.4	Vertical Sweep	38
5.5	Stereo Audio Codec Block Diagram	39
5.6	SPI Timing	39
5.7	Left-Justified Mode Timing	40
6.1	Bitmap File Format	41
6.2	Wave File Format	42
6.3	Hexadecimal Object File Format	42
6.4	Internal Memory ROM Control	43
6.5	Internal Memory FIFO Buffer Control	43

6.6	Write Process	44
6.7	Read Process	44
6.8	FIFO Buffer Transfer	45
6.9	MPEG-2 Decoder Output Transfer	45
6.10	Read and Write Contention	46
6.11	Read and Write Contention with Read Commands having Precedence	46
6.12	Read and Write Contention with Write Commands having Precedence	46
7.1	System Timing Controls	47
7.2	External Memory Timing Control	47
7.3	MPEG-2 Initialisation Synchronisation	49
7.4	MPEG-2 Activation Synchronisation	49
7.5	DVB-S Initialisation Synchronisation	50
7.6	DVB-S Activation Synchronisation	50
8.1	System Set-Up	51
8.2	Component Frequency Ranges	51
8.3	System Connections	52
8.4	FPGA DevBoard Connections	52
8.5	Malan's Board Connections	53
8.6	DVB-S EvBoard Connections	53
8.7	FPGA DevBoard to Malan's Board Connections	53
8.8	Amplifier and Attenuator Connections	54
8.9	DVB-S EvBoard to FPGA DevBoard Connections	54
9.1	Time Domain Measurement Points	56

9.2	Initial Sequence	57
9.3	Signal Distortion	57
9.4	Affects of Rise-Time and Fall-Time Distortion	58
9.5	Affects of Overshoot and Undershoot Distortion	59
9.6	Affects of Droop Distortion	60
9.7	Comparison between Transmitter Distortion Measurements	60
9.8	Comparison between Receiver Distortion Measurements	60
9.9	Signal Imbalance	61
9.10	Affects of Amplitude Imbalance	61
9.11	Affects of Phase Imbalance	62
9.12	Affects of DC-Offset	63
9.13	Comparison between Transmitter Imbalance Measurements	63
9.14	Frequency Domain Measurement Points	64
9.15	Comparison of Signal Generators	64
9.16	Blind Scan Panel Screenshot for Unamplified 80MHz Channel Spacing	65
9.17	Easy Start Panel Screenshot for Unamplified 80MHz Channel Spacing	65
9.18	Modulator Outputs for 80MHz Channel Spacing	66
9.19	Combiner Outputs for 80MHz Channel Spacing	66
9.20	Combiner Output for 80MHz Channel Spacing	67
9.21	Easy Start Panel Screenshot for Amplified 80MHz Channel Spacing	67
9.22	Amplifier Outputs for 80MHz Channel Spacing	68
9.23	Combined Amplifier Output for 80MHz Channel Spacing	68
9.24	Easy Start Panel Screenshot for Unamplified 45MHz Channel Spacing	69
9.25	Modulator Outputs for 45MHz Channel Spacing	69
9.26	Combiner Outputs for 45MHz Channel Spacing	70

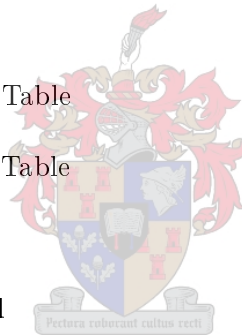
9.27	Combiner Output for 45MHz Channel Spacing	70
9.28	Easy Start Panel Screenshot for Amplified 45MHz Channel Spacing	71
9.29	Amplifier Outputs for 45MHz Channel Spacing	71
9.30	Combined Amplifier Output for 45MHz Channel Spacing	72
B.1	Corrected DVB-SPI Pin Assignments	85
B.2	LVDS-to-TTL Board Schematic	85
B.3	TTL-to-LVDS Board Schematic	86
B.4	Add-On Board Schematic	86
B.5	Duplicate Add-On Board Schematic	86
B.6	Top Layer	88
B.7	Bottom Layer	89
B.8	Top Silkscreen Overlay	90
B.9	Bottom Silkscreen Overlay	91
C.1	FPGA DevBoard Functional Diagram	93
D.1	DVB-S EvBoard Functional Diagram	96

Abbreviations

ADC	Analogue-to-Digital Converter
AF	Adaptation Field
AGC	Automatic Gain Control
AU	Access Unit
BER	Bit Error Rate
CAT	Conditional Access Table
CNR	Carrier-to-Noise Ratio
codec	coder-decoder
CRT	Cathode Ray Tube
DAC	Digital-to-Analogue Converter
DevBoard	Development Board
DIP	Dual In-line Package
DVB-S	Digital Video Broadcasting for Satellite
DVB-SPI	Digital Video Broadcasting - Synchronous Parallel Interface
EMM	Entitlement Management Message
ES	Elementary Stream
EvBoard	Evaluation Board
FEC	Forward Error Correction
FIFO	First-In, First-Out



FPGA	Field-Programmable Gate Array
GF	Galois Fields
ITU	International Telecommunication Union
ITU	International Telecommunications Union
LCD	Liquid Crystal Display
LED	Light Emitting Diode
LEO	Low Earth Orbit
LO	Local Oscillator
LVDS	Low Voltage Differential Signalling
MPEG-2	Motion Pictures Expert Group
MSB	Most Significant Bit
NIT	Network Information Table
PAT	Program Association Table
PC	Personal Computer
PCB	Printed Circuit Board
PES	Packetised Elementary Stream
PID	Packet Identifier
PLL	Phase-Lock Loop
PMT	Program Map Table
PSI	Program Specific Information
PSK	Phase Shift Keying
PU	Presentation Unit
QPSK	Quadrature Phase Shift Keying
RAM	Random Access Memory
RF	Radio Frequency



RGB	Red-Green-Blue
ROM	Read Only Memory
RS	Reed-Solomon
SNR	Signal-to-Noise Ratio
SPI	Serial Peripheral Interface
TS	Transport Stream
VGA	Video Graphics Adapter



Chapter 1

Introduction

Wireless communication is as old as or older than the first lighthouse or smoke signal, however man's desire for fast and accurate information greatly exceeds the capabilities of these primitive systems and shows every sign of continuing to demand for better, faster and more cost effective communication systems, of which satellite imaging systems is an example.

Satellite imaging systems require a wide-bandwidth, high-speed transmission link to cater for large high-resolution images. A better transmission link allows for more images to be downloaded and makes the system more productive. An investigation is made into current systems, which have the functions depicted in Figure 1.1, to find an affordable and upgradeable manner to increase the system's payload.

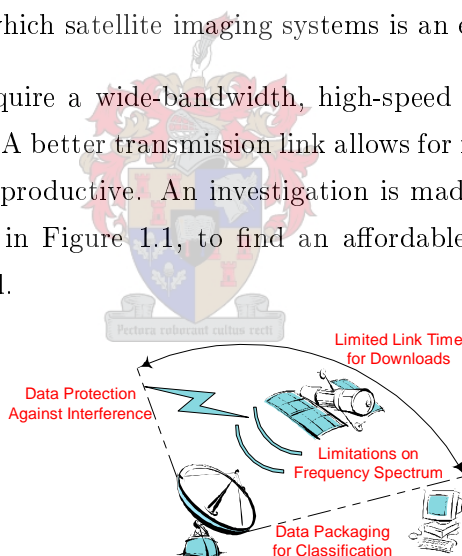


Fig. 1.1. Satellite System Functions

Imaging satellites are in Low Earth Orbit (LEO), they circle the earth in approximately 90 minutes taking pictures as they go along. Images are only downloaded while the LEO satellite is in line-of-sight of the ground station. The amount of data that is downloaded is defined as

$$\text{payload} = \text{data rate [bits per second]} \times \text{link time [seconds]} \quad (1.1)$$

The link time is determined by the orbit and link budget of the system, with the orbit chosen within the LEO range for a given type of image and time between revisiting a location. Therefore the link time is fixed and the only variable in Equation 1.1 is the data rate.

1.1 Problem Statement

The various processes within a satellite system are examined, with a block diagram of a simplified version depicted in Figure 1.2, to look at their effect on the data rate.

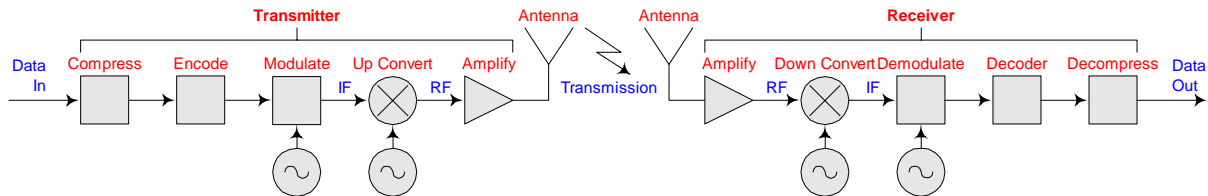


Fig. 1.2. Simplified Satellite System

The level of signal amplification required to obtain a given Signal-to-Noise Ratio (SNR) at the receiver is directly proportional to the signal's bandwidth [1]. The level of output power that devices can deliver decreases as the input signal's frequency increases. Therefore the available output power levels of existing components restrict an increase in data rate.

A transmitted signal's frequency, bandwidth and shape must conform to the telecommunication regulations set down by the International Telecommunication Union (ITU). These regulations limit health risks, prohibit encroachment into other signals' bands and divide the spectrum into bands for specific usages [2]. Therefore an increase in data rate is restricted by the maximum bandwidth within available frequency bands.

The form of modulation and the modulation devices' input frequency ranges limit the signal's data rate. Existing models have a maximum input frequency, which cannot be bypassed until new models are developed. Modulation techniques increase the system's baud rate by transmitting more symbols per carrier signal change, with a limit as to how miniscule the changes can be for the signal to be demodulated. Therefore existing modulation devices restricts an increase in data rate.

Data compression can be used to reduce the size of the images, thereby increasing the amount of information downloaded to the ground station for a given data rate. But, powerful compression algorithms are complex and work on multiple frames of data, with the risk of losing large chunks of irrecoverable data if the decoding process fails. Therefore the level of any compression used to increase the data rate is limited.

Unless linked to technological advances, any improvement in the devices that perform amplification, modulation and compression will only result in an extension of the system's capabilities and will not solve the data rate's reliance on these components. Therefore a solution is needed that is not based on improving upon existing devices, but rather on a technique that bypasses the limit places on the system by existing devices.

1.2 Proposed Solution

The chosen method is a parallel channel configuration, as depicted in Figure 1.3, designed in such a way that any demand for a further increase in the data rate would require little more than duplication of the existing architecture, thereby creating a system that supports an unlimited maximum data rate. A developer would be able to use the system to find the optimal number of channels for a specific application.

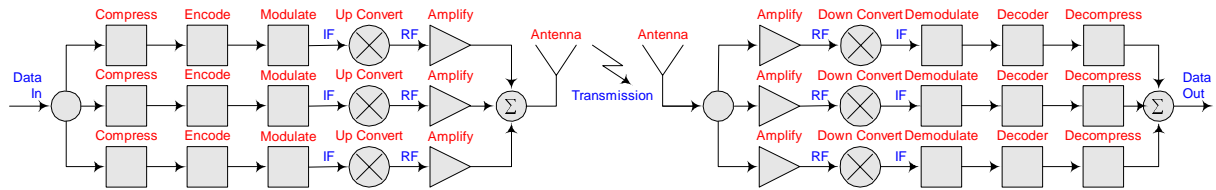


Fig. 1.3. Proposed Solution

Decreasing a channel's bandwidth reduces the required signal level to achieve a certain SNR. This makes available the options to use cheaper amplifiers and to place channels at higher frequencies. Whereas a single-channel system requires a high level of signal amplification, each channel in a multi-channel system would require only a fraction of this amplification.

The channels can be split up into different frequency bands thereby bypassing any regulatory limit on a specific band. A single-channel system cannot be made larger than the largest available bandwidth, whereas the next channel within a multi-channel system can be allocated to another band, as depicted in Figure 1.4.

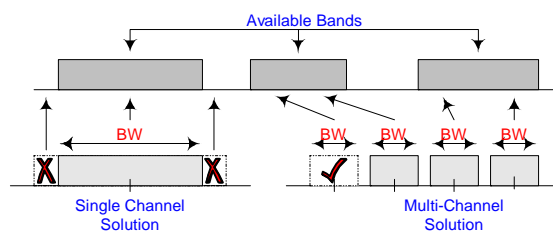


Fig. 1.4. Frequency Band Limitation Bypass

The configuration does not rely on any current modulation devices, therefore the system can always be upgraded with the latest available devices. The number of channels can be adjusted according to the price difference between a single wide-band device and multiple narrow-band devices.

The decrease in the frequency range per channel reduces the required circuitry sophistication, such as matching networks and amplifier gain flatness, thereby resulting in a cheaper system that is easier to design.

1.3 Implementation

To minimise costs and reduce development time only a subset of the whole system is implemented, as depicted in Figure 1.5.

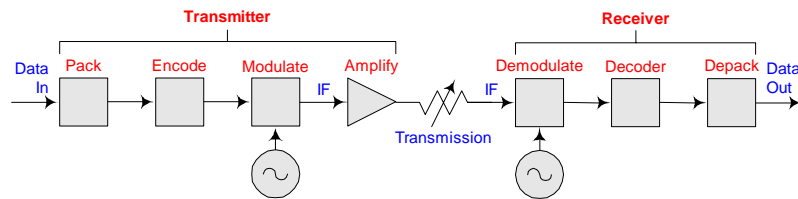


Fig. 1.5. Implemented System

The antennae, frequency up-converters and frequency down-converters are excluded as no signal radiation is performed. The Radio Frequency (RF) stages and transmission link are simulated by the amplifier and attenuator placed between the transmitter and receiver.

The digital processing and modulation stages are duplicated for each channel, the results of which are combined before transmission. The signal amplification position is alternated between being before and after the signal combination to determine the best configuration.

In order to be compared to current systems the system makes use of existing standards and is built from readily-available components, as depicted in Figure 1.6.

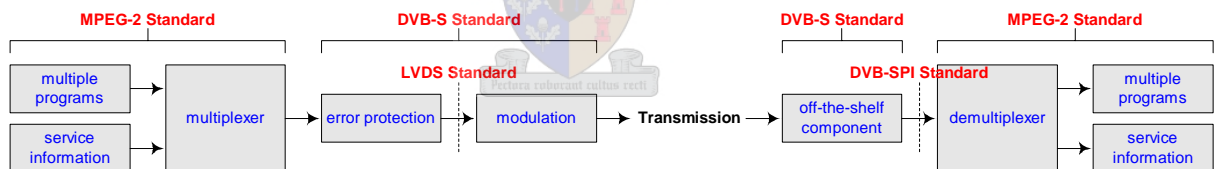


Fig. 1.6. System Block Diagram

Digital Video Broadcasting for Satellite (DVB-S) [3, 4] is the chosen method of data protection, signal modulation and signal dispersal. DVB-S combats internally created and externally encountered interference affecting satellite transmission, determines the modulation technique and shapes the signal to adhere to telecommunication regulations.

The DVB-S compliant [5] Motion Pictures Expert Group (MPEG-2) [6, 7, 8] is the chosen method of data packaging. Only the subset of the MPEG-2 standard that sorts the data according to type and the basic functions for operation are implemented, while the components that compress the audio, video and other data streams are excluded.

Low Voltage Differential Signalling (LVDS) [9, 10] and Digital Video Broadcasting - Synchronous Parallel Interface (DVB-SPI) [11] are the chosen method of data transfers. These standards are used to transfer digital signals between components in the system to prevent signal degradation.

1.3.1 System

The system implementation is depicted in Figure 1.7.

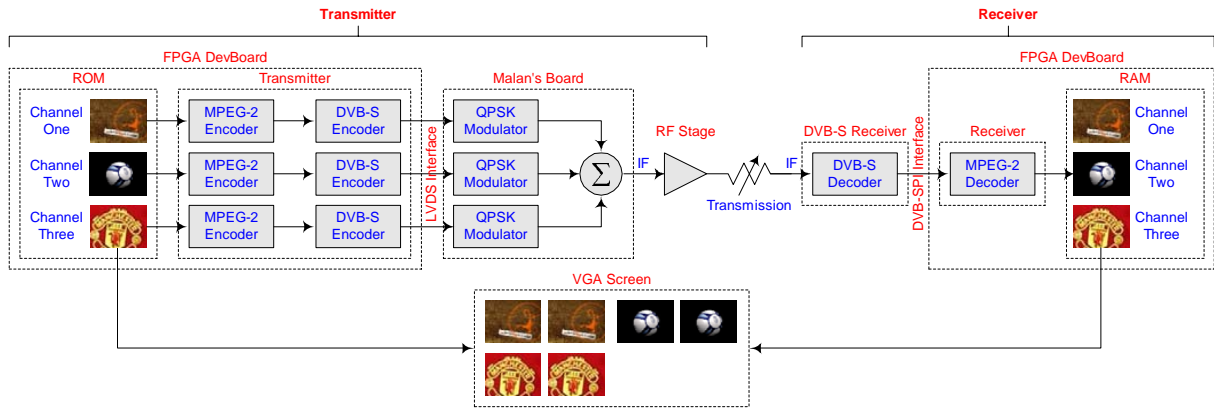


Fig. 1.7. System Implementation

The transmitter's digital processing is implemented within a Field-Programmable Gate Array (FPGA) [12] and the analogue modulation is performed by modulators [13]. The DVB-S decoding is performed by a DVB-S Receiver [14] and the MPEG-2 decoding is implemented within the FPGA.

The effect of the spacing between channels and the non-ideal properties of the amplifier [15, 16] on the DVB-S data correction abilities are visible as a degradation in the receiver's image data displayed on the Video Graphics Adapter (VGA) screen.

Three boards, containing components useful for debugging and testing purposes, are used to speed-up the realisation of a working model of the system. The FPGA Development Board (DevBoard) [17] contains external memory [18], video [19] and audio [20] devices. The DVB-S Evaluation Board (EvBoard) [21] contains an output interface. Malan's Board [22] contains a synthesizer [23] and mixer [24].

The FPGA DevBoard does not contain the necessary connectors and transmission lines to connect to the DVB-S EvBoard output interface of DVB-SPI. Malan's Board is not designed to directly port external data to the modulators. Therefore additional Printed Circuit Boards (PCB) containing components [25, 26] based on the LVDS standard were designed to implement these data transfers.

The current set-up of multiple boards is bulky and cumbersome. It would be wise to design a streamlined version of the system for use outside of this academic purpose. This improved version would be void of all the unnecessary components on the boards and would not require the additional boards and cabling due to the FPGA's support for LVDS [27, 28].

1.3.2 MPEG-2

The functional block diagram of the MPEG-2 encoder and decoder is depicted in Figure 1.8. The encoder converts the source data into Elementary Streams (ES), which are placed in Transport Stream (TS) payloads together with Program Specific Information (PSI) and passed onto the DVB-S encoder. The decoder uses the PSI to separate the received DVB-S decoded data into the respective ES for conversion back into source data.

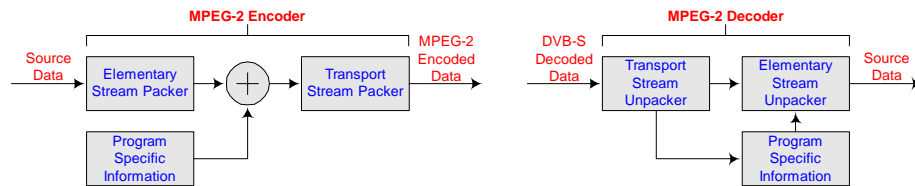


Fig. 1.8. MPEG-2 Block Diagram

1.3.3 DVB-S

The functional block diagram of the DVB-S encoder and decoder is depicted in Figure 1.9. The encoder scrambles, encodes and modulates the received MPEG-2 encoded data. The decoder performs a reverse process on the received RF signal and passes the result onto the MPEG-2 decoder.

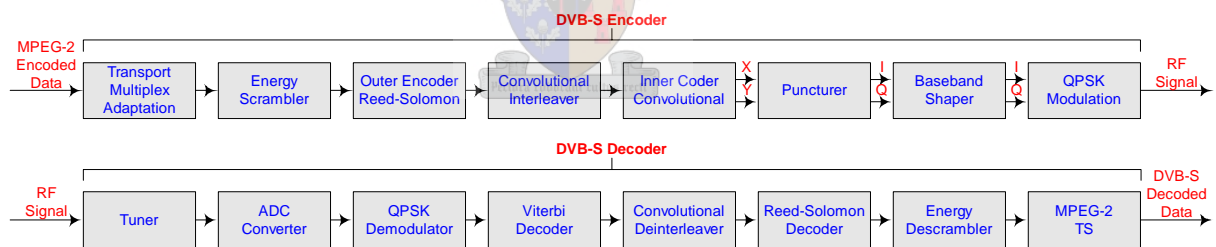


Fig. 1.9. DVB-S Block Diagram

1.3.4 LVDS

The LVDS drivers and receivers are used as depicted in Figure 1.10. LVDS driver-receiver pairs are used to transfer data between the FPGA DevBoard and Malan's Board and a LVDS receiver is used in the data transfer from the DVB-S EvBoard to the FPGA DevBoard.

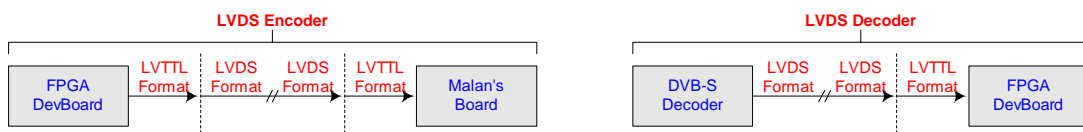


Fig. 1.10. LVDS Block Diagram

1.4 Thesis Outline

The first three chapters discuss the standards that the communication system is based on. Chapter 2 concentrates on the sections of the MPEG-2 standard that are implemented while only a brief overview on the sections that are not used is given. Chapter 3 explains each of the DVB-S processes with examples to help clarify the mathematical fields that they are based on. Chapter 4 explains the basics of the LVDS technology and the DVB-SPI standard and their implementation in the system.

The next three chapters discuss the audio, video, storage and synchronism mechanisms used within the system. Chapter 5 explains how the screen and audio coder-decoder (codec) are controlled to display the images and broadcast the audio. Chapter 6 describes the various memory units used, both the units within the FPGA and the units external to it, as well as the mechanism used to initialise the internal memory units with the audio and video data. Chapter 7 describes the processes within the FPGA with specific emphasis on their interconnections and synchronism.

The last three chapters discuss the completed system. Chapter 8 explains how the components are connected together and operated in order to complete the system. Chapter 9 discusses the system's successfulness based on measurements taken of the system. Chapter 10 is a summary of the system, stating conclusions and giving recommendations.

The appendices contain the PCB layouts, information about the kits and the environment settings on the FPGA DevBoard. Appendix A contains the pin assignments for the connections between the components on the FPGA DevBoard. Appendix B contains the manufactured PCB layout files. Appendix C contains information about the FPGA development kit. Appendix D contains information about the DVB-S evaluation kit.

The attached CD contains the Quartus source code for the program on the FPGA and the Delphi source code for the programs used to convert the image and audio files to a compatible format to be used to initialise the FPGA's internal memory units.

Chapter 2

MPEG-2

The Motion Pictures Expert Group (MPEG-2) standard [5, 6] is explained in this chapter. As MPEG-2 is only used as a packaging mechanism, which is explained in depth, the compression of the data streams is only briefly discussed. While only still pictures are used in this design, the system can be upgraded to moving pictures as MPEG-2 accommodates both data types.

2.1 Overview

The Transport Stream (TS) contains separate programs each with their own set of data streams, as depicted in Figure 2.1. The encoder assigns each data stream an identifier which is transmitted in Program Specific Information (PSI) tables along with the programs. The decoder distinguishes between the programs, and groups their related data streams by making use of these tables [7, 8].

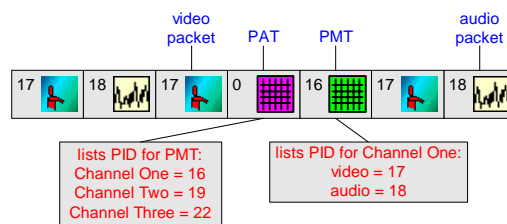


Fig. 2.1. TS Format

2.2 Elementary Stream

An Elementary Stream (ES) is a succession of Access Units (AU), which are encoded Presentation Units (PU). Different PU types and complexities lead to varying levels of compression and varying bit-rates. The most detailed information is selectively discarded to minimise the impact on quality and to obtain a fixed bit-rate [8].

2.2.1 Video Compression

Video compression [29] exploits the chrominance and luminance sensitivity of the human visual system: the eye's inability to see quantisation noise under conditions of visual masking. Temporal masking removes the redundancy of pixel replication in successive frames and spatial masking removes the redundancy of pixel replication within a single frame, as depicted in Figure 2.2.

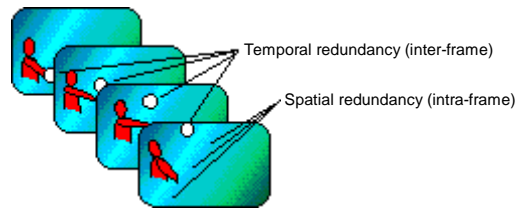


Fig. 2.2. Video Redundancy

The more similar subsequent images are the more references between images are possible and the better the level of compression.

2.2.2 Audio Compression

Audio compression [30, 31] exploits the psychoacoustic properties of the human auditory system: the ear's inability to hear quantisation noise under conditions of auditory masking. Faint sounds are frequency masked by louder spectral neighbours and weak sounds are temporal masked by stronger temporal neighbours, as depicted in Figure 2.3.

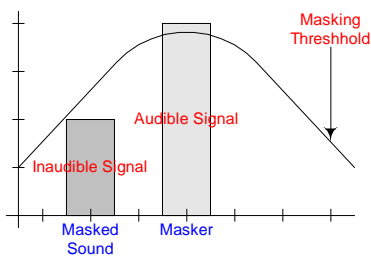


Fig. 2.3. Audio Masking

The more similar subsequent frames are the more references between frames are possible and the better the level of compression.

2.2.3 Implementation

Compression is excluded from the system and the PU are directly mapped to AU of equal size, due to the complexity of implementing compression in the FPGA exceeding the criterion of this thesis.

2.3 Packetised Elementary Stream

A Packetised Elementary Stream (PES) is a succession of PES-Packets made up of 6-byte headers, 3-byte optional headers and variable-length optional fields and payloads, as depicted in Figure 2.4.



Fig. 2.4. PES-Packet

The payload contains data from one ES, the header contains information about the ES and the optional header indicates which optional fields containing additional information are present.

2.3.1 Implementation

As compression is not performed and PES primarily contains information about audio and video decoding packing of ES to PES is not implemented and the ES are fed directly to the TS.

2.4 Program Association Table

A Program Association Table (PAT) is 21-bytes long and contains a 1-byte pointer field, an 8-byte header and a 12-byte payload, as depicted in Figure 2.5.

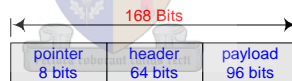


Fig. 2.5. PAT-Section

The header contains information about the PAT-Section, the payload links the program numbers to their associated program and the pointer specifies the number of bytes between the pointer and the rest of the PAT-Section, namely zero.

2.4.1 PAT-Section Header

The header contains the following implemented fields, as depicted in Figure 2.6.

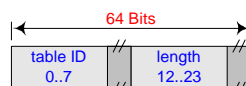


Fig. 2.6. PAT-Section Header

TABLE ID is assigned the value 00_{16} to identify that this is a PAT-Section. **LENGTH** is assigned the value 011_{16} , the number of bytes in the PAT-Section less 3 bytes.

2.4.2 PAT-Section Payload

The payload contains the following two fields for each program, as depicted in Figure 2.7.

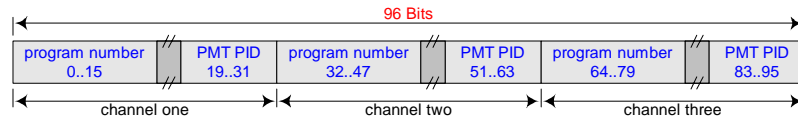


Fig. 2.7. PAT-Section Payload

PROGRAM NUMBER is assigned the program that the PMT Packet Identifier (PID) is associated with, as defined in Table 2.1.

TABLE 2.1
PROGRAM NUMBER ASSIGNMENTS

Program Number	Assignment
0001 ₁₆	Channel One
0002 ₁₆	Channel Two
0003 ₁₆	Channel Three

PMT PID is assigned the TS-Packet PID that contains the PMT-Sections, as defined in Table 2.2.

TABLE 2.2
PMT PID ASSIGNMENTS

PMT PID	Assignment
0010 ₁₆	Channel One - PMT
0013 ₁₆	Channel Two - PMT
0016 ₁₆	Channel Three - PMT

2.5 Program Map Table

A Program Map Table (PMT) is 29-bytes long and contains a 1-byte pointer field, a 12-byte header and a 16-byte payload, as depicted in Figure 2.8.

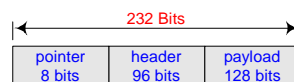


Fig. 2.8. PMT-Section

The header contains information about the PMT-Section, the payload links the ES to their associated program and the pointer specifies the number of bytes between the pointer and the rest of the PMT-Section, namely zero.

2.5.1 PMT-Section Header

The header contains the following implemented fields, as depicted in Figure 2.9.

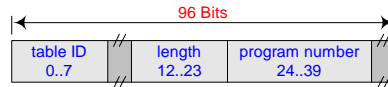


Fig. 2.9. PMT-Section Header

TABLE ID is assigned value 02_{16} to identify that this is a PMT-Section. **LENGTH** is assigned the value 019_{16} , the number of bytes in the PMT-Section less 3 bytes. **PROGRAM NUMBER** is assigned the program to which the PMT PID is applicable, as defined in Table 2.1.

2.5.2 PMT-Section Payload

The payload contains the following implemented fields for each ES within the channel, as depicted in Figure 2.10.

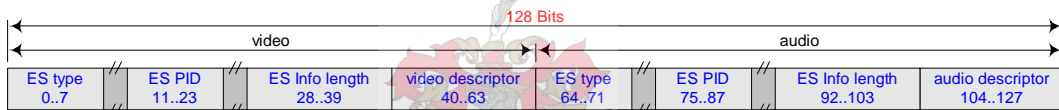


Fig. 2.10. PMT-Section Payload

ES TYPE is assigned the ES type within the TS-Packet payload, as defined in Table 2.3.

TABLE 2.3
ES TYPE ASSIGNMENTS

ES Type	Description	ES Type	Description
$C1_{16}$	Audio Stream One	$E1_{16}$	Video Stream One
$C2_{16}$	Audio Stream Two	$E2_{16}$	Video Stream Two
$C3_{16}$	Audio Stream Three	$E3_{16}$	Video Stream Three

ES PID is assigned the TS-Packet PID carrying the ES, as defined in Table 2.4.

TABLE 2.4
ES PID ASSIGNMENTS

ES PID	Indication	ES PID	Indication
0011_{16}	Channel One - Video ES	0012_{16}	Channel One - Audio ES
0014_{16}	Channel Two - Video ES	0015_{16}	Channel Two - Audio ES
0017_{16}	Channel Three - Video ES	0018_{16}	Channel Three - Audio ES

ES INFORMATION LENGTH is assigned the value 018_{16} , the number of bytes in the descriptors.

Video Stream Descriptor

The video stream descriptor identifies the coding parameters of the video ES. The video stream descriptor has the following implemented fields, as depicted in Figure 2.11.

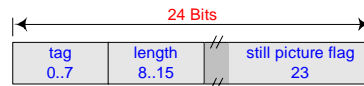


Fig. 2.11. Video Stream Descriptor

TAG is assigned the value 02_{16} to identify that it is a video stream descriptor. **LENGTH** is assigned the value 01_{16} , the number of bytes in the descriptor less 2 bytes. **STILL PICTURE FLAG** is asserted to indicate that the video ES contains only still pictures.

Audio Stream Descriptor

The audio stream descriptor identifies the coding version of an audio ES. The audio stream descriptor has the following implemented fields, as depicted in Figure 2.12.



Fig. 2.12. Audio Stream Descriptor

TAG is assigned the value 03_{16} to identify that it is an audio stream descriptor. **LENGTH** is assigned the value 01_{16} , the number of bytes in the descriptor less 2 bytes.

2.6 Conditional Access Table

The Conditional Access Table (CAT) lists the PID of the Entitlement Management Messages (EMM) used in conditional access systems. The PID and Table ID are assigned the value 01_{16} to identify that it is a CAT-Section. The CAT is optional, is not defined by MPEG-2 and is not implemented in this system.

2.7 Network Information Table

The Network Information Table (NIT) is used to describe the physical network parameters, such as the channel frequencies, satellite transponder details and modulation characteristics. The PID and Table ID are assigned the value 03_{16} to identify that it is a NIT-Section. The NIT is optional, is not defined by MPEG-2 and is not implemented in this system.

2.8 Transport Stream

A TS is organised into 188-byte packets made up of a 4-byte header and 184 bytes divided between the payload and Adaptation Field (AF), as depicted in Figure 2.13.

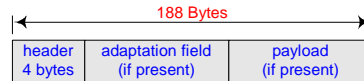


Fig. 2.13. TS-Packet

Each payload contains the data from a PES-Packet; the header contains information about this PES-Packet and the AF contains additional information about this PES-Packet.

2.8.1 TS-Packet Header

The header contains the following implemented fields, as depicted in Figure 2.14.



Fig. 2.14. TS-Packet Header

SYNC BYTE is assigned the value 47_{16} to synchronise the depacking of the TS. **PAYLOAD UNIT START INDICATOR** is asserted to indicate the start of a PES-Packet payload or the start of a PSI-Section. **PACKET ID** is assigned the payload's data type, as defined in Table 2.5.

TABLE 2.5
PID ASSIGNMENTS

PID	Indication	PID	Indication
0000_{16}	PAT	0010_{16}	Channel One - PMT
0001_{16}	CAT	0011_{16}	Channel One - Video ES
0003_{16}	NIT	0012_{16}	Channel One - Audio ES
0013_{16}	Channel Two - PMT	0016_{16}	Channel Three - PMT
0014_{16}	Channel Two - Video ES	0017_{16}	Channel Three - Video ES
0015_{16}	Channel Two - Audio ES	0018_{16}	Channel Three - Audio ES

AF CONTROL is assigned the payload's contents, as defined in Table 2.6.

TABLE 2.6
AF CONTROL ASSIGNMENTS

AF Control	Indication	AF Control	Indication
00_2	Reserved	10_2	AF only
01_2	Payload only	11_2	AF followed by Payload

CONTINUITY COUNTER is incremented with each TS-Packet payload with the same PID. If sequential TS-Packets contain **CONTINUITY COUNTER** with a difference of greater than one, as depicted in Figure 2.15, an error state is reached as at least one TS-Packet has been lost in transmission.

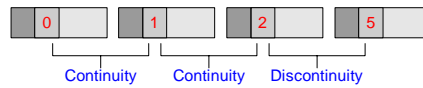


Fig. 2.15. Continuity Counter

2.8.2 TS-Packet Adaptation Field

If present the AF is inserted between the packet's header and payload. The AF contains the following implemented fields, as depicted in Figure 2.16.

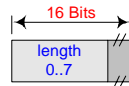


Fig. 2.16. Adaptation Field

LENGTH is assigned the number of bytes in the AF less 1 byte.

2.8.3 TS-Packet Payload

The PES and PSI are broken up into 184-byte blocks and placed sequentially in TS-packet payloads, as depicted in Figure 2.17.

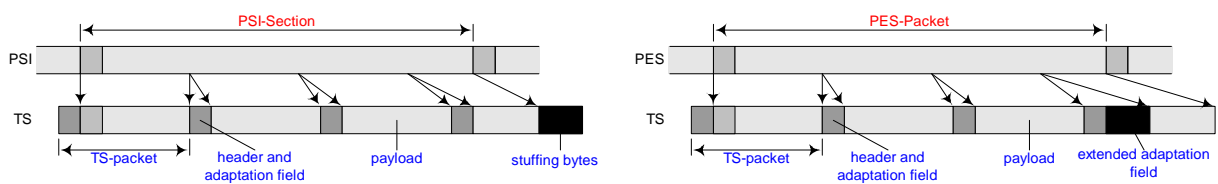


Fig. 2.17. Conversion of PES and PSI to TS

For PES-Sections that do not fill an integral amount of TS-Packet payloads the AF of the last TS-Packet is extended to fill this gap, thereby aligning the end of the PSI-Section with the end of the TS-Packet. The AF **LENGTH** is incremented appropriately in order for the decoder to skip these bytes and the bytes are assigned the value FF_{16} .

For PSI-Sections that do not fill an integral amount of TS-Packet payloads the remainder of the last TS-Packet payload is filled with bytes assigned the value FF_{16} . The PSI-Section header **LENGTH** is not incremented and the decoder skips these bytes.

2.9 Implementation

Transmitters and receivers based on the subset of the MPEG-2 system described in this chapter are coded for use in the FPGA.

Transmitter

Each channel continually transmits the data stream depicted in Figure 2.18. The TS-Packet payloads are image pixels read in from the FPGA's internal memory and the PSI-Sections and TS-Packet headers are dynamically generated to describe the structure of the current payload. The synchronisation of these processes is discussed in Section 7.2

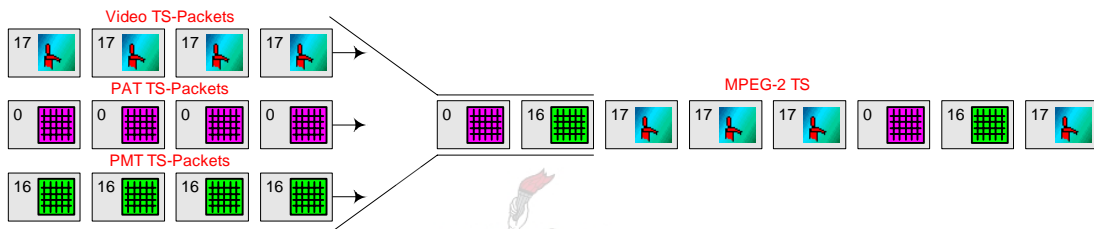


Fig. 2.18. Transmitter Implementation

Receiver

The receiver uses the TS-Packet header's PID field to separate the payload types, as depicted in Figure 2.19. The PAT decoder's output helps to decode the PMT, the PMT decoder's output helps to decode the video payloads and the video decoder's output is stored in external memory.

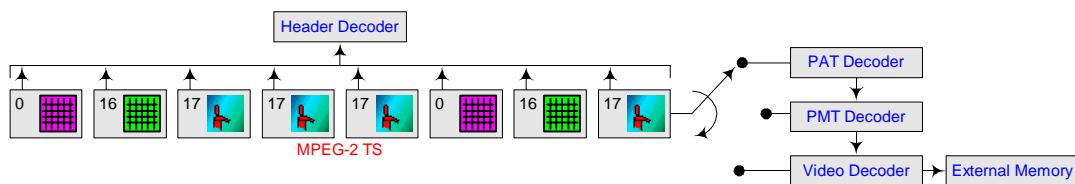


Fig. 2.19. Receiver Implementation

2.10 Summary

This chapter described the implemented subset of the MPEG-2 standard used to package the source data to facilitate the receiver's reconstruction of the original images for storage and post-processing. The MPEG-2 encoded data in the source data for the DVB-S standard. The DVB-S standard is discussed in the next chapter.

Chapter 3

DVB-S

The Digital Video Broadcasting for Satellite (DVB-S) standard [3] is discussed in this chapter. DVB-S is used to counter interference, modulate the data and shape the signal to adhere to telecommunication regulations. How the various processes perform these functions, the mathematical fields that they are based on and their implementation in the system are described. Each section contains examples to help clarify the processes' purposes.

3.1 Transport Multiplex Adaptation

The DVB-S encoder's input is the MPEG-2 encoder's output and the DVB-S decoder's output is the MPEG-2 decoder's input, as depicted in Figure 3.1. In the transmitter the sync byte of every eighth Transport Stream (TS) packet is bit-wise inverted to $B8_{16}$ for synchronisation purposes. In the receiver this inversion is mirrored to regain the original sync byte value of 47_{16} .

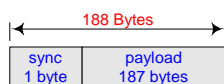


Fig. 3.1. TS-Packet

3.2 Energy Dispersal

A transmitted signal with a fixed pattern creates line components in the Radio Frequency (RF) carrier's spectrum that causes power flux-density peaks to occur at the surface of the Earth that exceed the International Telecommunication Union's (ITU) recommended level. Therefore an energy dispersal technique [32] is required to adequately randomise this signal to disperse the RF carrier's energy, while not degrading the quality of the data stream.

3.2.1 Spectra of PSK Digital Signals

The power spectrum of a carrier modulated by ideal phase reversals consists of lines separated by $\frac{1}{Nt}$ Hz, where N is the number of symbols of the pseudo-random sequence and t is the seconds per symbol. The power spectrum of these lines is given by Equation 3.1

$$W(f) \approx \frac{1}{N} \left\{ \frac{\sin \pi (f - f_c) t}{\pi (f - f_c) t} \right\}^2 \delta \left(f - f_c - \frac{n}{Nt} \right) \quad (3.1)$$

where n is an integer, f_c is the RF carrier frequency and δ is the Dirac-delta function.

The largest line is at $n = 1$. As the sequence length approaches infinity the line separation approaches zero, the power spectrum becomes continuous and the maximum spectral-density occurs at the carrier frequency. Equation 3.2 refers to the idealised situation of a random sequence, while an actual modulating signal may be far from random.

$$W(f) = t \left\{ \frac{\sin \pi (f - f_c) t}{\pi (f - f_c) t} \right\}^2 \quad (3.2)$$

To ensure the desired degree of dispersal a sequence of Nt duration is added to the information stream. For a reference bandwidth of x Hz the energy dispersal factor, D , is defined by Equation 3.3.

$$D = 10 \log \frac{\text{total power}}{\text{maximum power per } x \text{ Hz}} \quad (3.3)$$

The degree of dispersal is estimated by Equation 3.1 and Equation 3.3 when $\frac{1}{Nt} \geq x$ Hz and by Equation 3.2 and Equation 3.3 when $\frac{1}{Nt} < x$ Hz. The degree of dispersal is proportional to N as long as the sequence duration is less than the reciprocal of the reference bandwidth. Little additional dispersal is gained after the sequence duration passes this point.

3.2.2 Implementation

A scrambler generates a sequence to convert the data stream, as depicted in Figure 3.1, into a pseudo-random pattern, depicted in Figure 3.2.

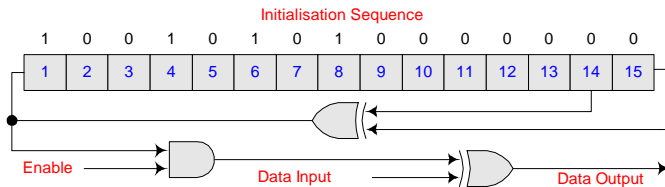


Fig. 3.2. Energy Scrambler Diagram

The polynomial $1 + X^{14} + X^{15}$ organises the exclusive-or gate connections and the initialisation sequence 100101010000000_2 is loaded into the shift register after every eighth TS-Packet.

The first bit of the sequence is applied to the first bit of the first byte following the inverted sync byte. For synchronisation purposes the sequence's output is disabled for the subsequent seven sync bytes thereby leaving these bytes unchanged. Thus the sequence has a 1503-byte period, as depicted in Figure 3.3.

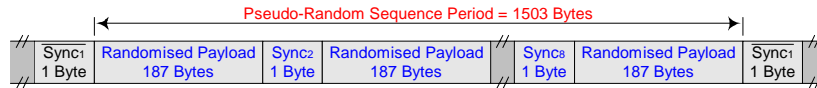


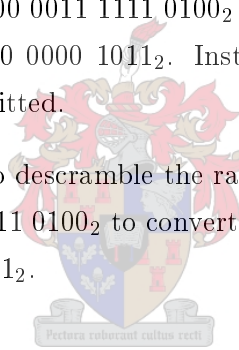
Fig. 3.3. Randomised Transport Packets

A descrambler of identical design is used to remove the energy dispersal to recover the original data stream.

Example

The energy scrambler in Figure 3.2 is used to scramble the fixed pattern data stream of $1111\ 1111\ 1111\ 1111_2$. The sequence $0000\ 0011\ 1111\ 0100_2$ is generated and converts the data stream into the random pattern $1111\ 1100\ 0000\ 1011_2$. Instead of the 16 consecutive '1' bits a more evenly distributed signal is transmitted.

An identical descrambler is used to descramble the random pattern. The descrambler generates the same sequence of $0000\ 0011\ 1111\ 0100_2$ to convert the random pattern back into the original data stream of $1111\ 1111\ 1111\ 1111_2$.



3.3 Reed-Solomon Coding

Reed-Solomon (RS) [33, 34] is a form of Forward Error Correction (FEC) that is used in data transmission to counter burst errors by handling errors on a symbol-length basis.

The RS encoder breaks the digital data into equal-sized blocks and appends parity symbols to each block. The RS decoder attempts to recover the original block by finding and correcting any errors that occurred during transmission.

RS is specified as $RS(n, k, t)$ with m -bit symbols, where n is the codeword length, k is the number of data symbols, t is the number of errors that can be found and corrected, $2t = n - k$ is the number of parity symbols and $n \leq 2^m + 1$, as depicted in Figure 3.4.

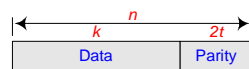


Fig. 3.4. Reed-Solomon Codeword

If a continuous burst of errors is bound within t symbols in a block the RS decoder is able to decode that block. This is depicted for a RS(7, 3, 2) decoder with 3-bit symbols in Figure 3.5.

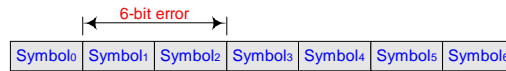


Fig. 3.5. Retrievable Codeword

If the errors occur in a random fashion where more than t symbols are affected, then the RS decoder is unable to decode that block. This is depicted for a RS(7, 3, 2) with 3-bit symbols in Figure 3.6.

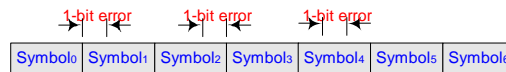


Fig. 3.6. Irretrievable Codeword

Therefore RS coding is used in conjunction with convolutional coding, as discussed in Section 3.5, which performs well with random errors.

3.3.1 Galois Fields

RS coding is based on an area of finite fields known as Galois Fields (GF). In finite fields the arithmetic operations of addition, subtraction, multiplication and division on field elements results in an element in the field.

For any prime number, p , there exists a finite field, $GF(p)$, that contains p elements. The extension field, $GF(p^m)$, extends this to a field of p^m elements, where m is a nonzero positive integer and $GF(p^m)$ contains the elements of $GF(p)$. RS symbols are constructed from $GF(2^m)$.

Each nonzero element in $GF(2^m)$ can be represented by a power of α . An infinite set of elements, F , is formed by starting with the elements $\{0, \alpha^0, \alpha^1\}$ and generating additional elements by progressively multiplying the last entry by α , which yields

$$F = \{0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^j, \dots\} \tag{3.4}$$

To obtain the finite set of elements of $GF(2^m)$ from F the condition $\alpha^{2^m-1} = \alpha^0$ is imposed on F to limit it to only 2^m elements. Therefore any field element with a power greater than $2^m - 1$ is reduced to an element with a power less than $2^m - 1$. This reduces F to the finite field F^* , with

$$F^* = \{0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{2^m-2}, \alpha^0, \alpha^1, \alpha^2\} \tag{3.5}$$

and the elements of $GF(2^m)$ are

$$GF(2^m) = \{0, \alpha^0, \alpha^1, \alpha^2, \dots, \alpha^{2^m-2}\} \tag{3.6}$$

Each of the 2^m elements of $GF(2^m)$ can be represented as a distinct polynomial of degree $m - 1$ or less. Each of the nonzero elements of $GF(2^m)$ is a polynomial, $a_i(X)$, where at least one of the m coefficients is nonzero. For $i = 0, 1, 2, \dots, 2^m - 2$,

$$\alpha^i = a_i(X) = a_{i,0} + a_{i,1}X + \dots + a_{i,m-1}X^{m-1} \quad (3.7)$$

Addition (with subtraction being equivalent) of two elements of $GF(2^m)$ is modulo-two sum of each of the polynomial coefficients of like powers, therefore

$$\alpha^i + \alpha^j = (a_{i,0} + a_{j,0}) + (a_{i,1} + a_{j,1})X + \dots + (a_{i,m-1} + a_{j,m-1})X^{m-1} \quad (3.8)$$

Multiplication of two field elements is the addition of the exponents modulo $2^m - 1$.

Primitive polynomials define the extension finite fields that define the RS codes. An irreducible polynomial, $f(X)$, of degree m is said to be primitive if the smallest positive integer n for which $f(X)$ divides $X^n + 1$ is $n = 2^m - 1$, yielding a nonzero quotient and a zero remainder.

Example

$GF(2^3)$ is defined by $f(X) = 1 + X + X^3$. There are $2^m = 8$ finite elements and $m = 3$ roots. To solve for the roots of $f(X)$ the values of X that correspond to $f(X) = 0$ must be found. The binary elements of 1 and 0 do not satisfy $f(X)$ since $f(1) = 1$ and $f(0) = 1$. Therefore the roots lie in $GF(2^3)$. Define α , an element of $GF(2^3)$, to be a root of $f(X)$. Therefore

$$\begin{aligned} f(\alpha) &= 0 \\ 1 + \alpha + \alpha^3 &= 0 \\ \alpha^3 &= -1 - \alpha \end{aligned} \quad (3.9)$$

Since in the binary field $+1 = -1$ α^3 can be represented as

$$\alpha^3 = 1 + \alpha \quad (3.10)$$

Similarly the other powers are determined to be

$$\begin{aligned} \alpha^4 &= \alpha \times \alpha^3 = \alpha + \alpha^2 \\ \alpha^5 &= \alpha \times \alpha^4 = 1 + \alpha + \alpha^2 \\ \alpha^6 &= \alpha \times \alpha^5 = 1 + \alpha^2 \\ \alpha^7 &= \alpha \times \alpha^6 = \alpha^0 \end{aligned} \quad (3.11)$$

As $\alpha^7 = \alpha^0$ the eight finite field elements of $GF(2^3)$ are

$$GF(2^3) = \{0, \alpha^0, \alpha^1, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6\} = \{0, 1, 2, 4, 3, 6, 7, 5\} \quad (3.12)$$

3.3.2 Reed-Solomon Encoding

The generator polynomial, $g(X)$, for a RS code takes the form of

$$g(X) = g_0 + g_1X + g_2X^2 + \cdots + g_{2t-1}X^{2t-1} + X^{2t} \quad (3.13)$$

There are $2t$ successive powers of α that are the roots of the polynomial, $\alpha, \alpha^2, \dots, \alpha^{2t}$. We describe $g(X)$ in terms of its $2t$ roots as

$$g(X) = (X - \alpha)(X - \alpha^2) \dots (X - \alpha^{2t}) \quad (3.14)$$

The codeword polynomial, $U(X)$, for a RS encoder is

$$U(X) = p(X) + X^{n-k}m(X), \quad (3.15)$$

where $m(X)$ is the message polynomial, X^{n-k} is the algebraic manipulation to right-shift $m(X)$ $n - k$ positions in the codeword and $p(X)$ is the parity polynomial that is placed in the leftmost $n - k$ stages of the codeword. $p(X)$ is defined as

$$p(X) = X^{n-k}m(X) \bmod g(X) \quad (3.16)$$

Example

A RS(7, 3, 2) encoder is used to encode the data stream $111011010_2 = \alpha^5 + \alpha^3 + \alpha^1$, which is equivalent to $m(X) = \alpha^5X^2 + \alpha^3X + \alpha^1$. From Equation 3.14 $g(X)$ is calculated to be

$$\begin{aligned} g(X) &= (X - \alpha)(X - \alpha^2)(X - \alpha^3)(X - \alpha^4) \\ &= (X^2 - \alpha^4 + \alpha^3)(X^2 - \alpha^6 + \alpha^0) \\ &= X^4 - \alpha^3X^3 + \alpha^0X^2 - \alpha^1X + \alpha^3 \end{aligned} \quad (3.17)$$

Since in the binary field $+1 = -1$ $g(X)$ can be represented as

$$g(X) = \alpha^3 + \alpha^1X + \alpha^0X^2 + \alpha^3X^3 + X^4 \quad (3.18)$$

From Equation 3.16 $m(X)$ is multiplied by X^4 to yield $\alpha^5X^6 + \alpha^3X^5 + \alpha^1X^4$ and then divided by $g(X)$ in Equation 3.18 to yield a $p(X)$ of

$$p(X) = \alpha^6X^3 + \alpha^4X^2 + \alpha^2X + \alpha^0 \quad (3.19)$$

From Equation 3.15 $U(X)$ is

$$U(X) = \alpha^5X^6 + \alpha^3X^5 + \alpha^1X^4 + \alpha^6X^3 + \alpha^4X^2 + \alpha^2X + \alpha^0 \quad (3.20)$$

3.3.3 Reed-Solomon Decoding

The RS decoder's received polynomial, $r(X)$, is the sum of the transmitted codeword and any error-pattern, $e(X)$, encountered in transmission, therefore

$$r(X) = U(X) + e(X) \quad (3.21)$$

Non-binary symbols require that both the error locations be found and the correct symbols at those locations be determined. If there are v errors in the codeword at location $X^{j_1}, X^{j_2}, \dots, X^{j_v}$ $e(X)$ would be

$$e(X) = e_{j_1}X^{j_1} + e_{j_2}X^{j_2} + \dots + e_{j_v}X^{j_v} \quad (3.22)$$

where the indices $1, 2, \dots, v$ refer to the first, second, \dots , v^{th} errors and the index j refers to the error location. To correct the corrupted codeword each error value e_{j_l} and its location X^{j_l} , where $l = 1, 2, \dots, v$, must be determined. An error locator number is defined as $\beta_l = \alpha^{j_l}$.

$U(X)$ and $g(X)$ have the same roots as $U(X)$ is a multiple of $g(X)$. Therefore a parity check performed on $r(X)$ results in a zero value to indicate a valid members of the codeword set and a nonzero value to indicate the presence of errors. The syndrome, S , is the result of this parity check and is computed as

$$S_i = r(\alpha^i) = e_{j_1}\beta_1^i + e_{j_2}\beta_2^i + \dots + e_{j_v}\beta_v^i \quad (3.23)$$

with $i = 1, 2, \dots, 2t$.

There are $2t$ unknowns, made up of t error values and t locations, and $2t$ simultaneous equations. These equations can not be solved in the conventional way as they are non-linear. To find the location of these errors an error-locator polynomial, $\sigma(X)$, is defined as

$$\sigma(X) = (1 + \beta_1X)(1 + \beta_2X) \dots (1 + \beta_vX), \quad (3.24)$$

where its roots are $1/\beta_1, 1/\beta_2, \dots, 1/\beta_v$. Using autoregressive-modelling techniques a matrix is formed from S , where the first t syndromes are used to predict the next syndrome:

$$\begin{bmatrix} \sigma_t \\ \sigma_{t-1} \\ \vdots \\ \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} S_1 & S_2 & \cdots & S_{t-1} & S_t \\ S_2 & S_3 & \cdots & S_t & S_{t+1} \\ & & \vdots & & \\ S_{t-1} & S_t & \cdots & S_{2t-3} & S_{2t-2} \\ S_t & S_{t+1} & \cdots & S_{2t-2} & S_{2t-1} \end{bmatrix}^{-1} \begin{bmatrix} -S_{t+1} \\ -S_{t+2} \\ \vdots \\ -S_{2t-1} \\ -S_{2t} \end{bmatrix} \quad (3.25)$$

We determine these roots by exhaustive testing of $\sigma(X)$ with each of the field elements. An element X that yields $\sigma(X) = 0$ is a root and its reciprocal is the location of an error.

Substitution of the location values into the syndrome equations yields the values of the errors. As the RS decoder is limited to finding t error locations and correcting t errors, t syndrome equations are required. This results in the estimated error polynomial, $\hat{e}(X)$, where

$$\hat{e}(X) = e_{j_1}X^{j_1} + e_{j_2}X^{j_2} + \dots + e_{j_v}X^{j_v}, \quad (3.26)$$

The estimated codeword polynomial, $\hat{U}(X)$, is the combination of $r(X)$ and $\hat{e}(X)$ and is equal to $U(X)$ as long as $r(X)$ does not contain more than t errors.

Example

In Section 3.3.2 a RS(7, 3, 2) is used to encode a data stream that resulted in the codeword in Equation 3.20. During transmission the maximum number of symbols, $t = 2$, are corrupted to create the received codeword

$$r(X) = \alpha^5 X^6 + \alpha^3 X^5 + \alpha^6 X^4 + \alpha^0 X^3 + \alpha^4 X^2 + \alpha^2 X + \alpha^0 \quad (3.27)$$

From Equation 3.23 the four syndrome symbols are found to be

$$\begin{aligned} S_1 &= r(\alpha^1) = \alpha^3 \\ S_2 &= r(\alpha^2) = \alpha^5 \\ S_3 &= r(\alpha^3) = \alpha^6 \\ S_4 &= r(\alpha^4) = 0 \end{aligned} \quad (3.28)$$

From Equation 3.25 the autoregressive model is found to be

$$\begin{bmatrix} \sigma_2 \\ \sigma_1 \end{bmatrix} = \begin{bmatrix} \alpha^1 & \alpha^0 \\ \alpha^0 & \alpha^5 \end{bmatrix} \begin{bmatrix} \alpha^6 \\ 0 \end{bmatrix} = \begin{bmatrix} \alpha^0 \\ \alpha^6 \end{bmatrix} \quad (3.29)$$

From Equation 3.24 and Equation 3.29 $\sigma(X)$ is found to be

$$\sigma(X) = \alpha^0 + \alpha^6 X + \alpha^0 X^2 \quad (3.30)$$

From Equation 3.30 the roots and the error locations are found to be

$$\begin{aligned} \sigma(\alpha^3) &= 0 \Rightarrow \beta_1 = \alpha^4 \\ \sigma(\alpha^4) &= 0 \Rightarrow \beta_2 = \alpha^3 \end{aligned} \quad (3.31)$$

From Equation 3.22 and Equation 3.31 the error values are found to be

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} \alpha^2 & \alpha^5 \\ \alpha^0 & \alpha^4 \end{bmatrix} \begin{bmatrix} \alpha^3 \\ \alpha^5 \end{bmatrix} = \begin{bmatrix} \alpha^2 \\ \alpha^5 \end{bmatrix} \quad (3.32)$$

From Equation 3.26 $\hat{e}(X)$ is found to be

$$\hat{e}(X) = \alpha^2 X^3 + \alpha^5 X^4 \quad (3.33)$$

From Equation 3.33 and Equation 3.27 $\hat{U}(X)$ is found to be

$$\hat{U}(X) = \alpha^5 X^6 + \alpha^3 X^5 + \alpha^1 X^4 + \alpha^6 X^3 + \alpha^4 X^2 + \alpha^2 X + \alpha^0 \quad (3.34)$$

This estimated codeword is equal to the transmitted codeword in Equation 3.20.

3.3.4 Implementation

The RS(255, 239, 8) shortened code RS(204, 188, 8) with $f(X) = X^8 + X^4 + X^3 + X^2 + 1$ and $g(X) = (X - \alpha^0)(X - \alpha^1) \dots (X - \alpha^{15})$ is implemented.

The compiler [35] encodes each randomised transport packet, as depicted in Figure 3.3, to generate an error protected packet, as depicted in Figure 3.7. The decoder decodes the RS encoded packets and outputs the randomised transport packets to the descrambler.

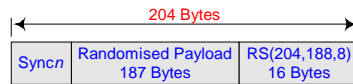


Fig. 3.7. Reed Solomon Error Protected Packet

3.4 Interleaving

The data stream is interleaved to reduce the chance of a burst error from affecting consecutive symbols. A convolutional interleaver scrambles the symbols as a periodic function of time.

3.4.1 Convolutional Interleaving

Convolutional interleaving [4, 36] is created by diagonally splitting a B -by- N rectangular array into two halves, where B is the number of shift registers and N is the error protection frame length, as depicted in Figure 3.8. One half is the interleaver and the other half is the deinterleaver.

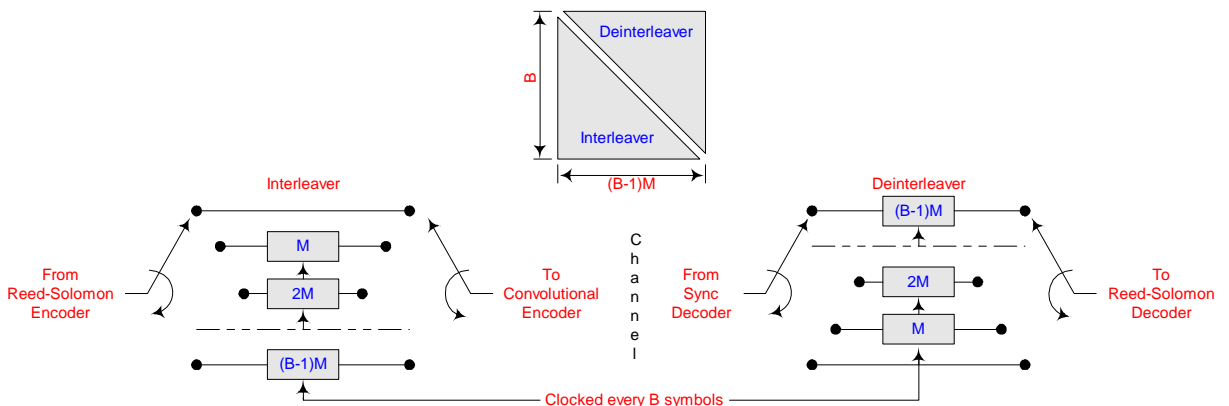


Fig. 3.8. B -by- N Convolutional Interleaver

At the transmitter the coded sequence is fed into a B -by- N triangular array of shift registers. The i^{th} shift register has a depth of $(1 - i)M$ stages, where $1 \leq i \leq B$ and $M = N/B$. The registers are clocked once every B symbols and the oldest symbols in the registers are shifted out to the channel.

The received sequence is restored to its original ordering using an inverse structure of shift registers. Symbols that are delayed by $(i - 1)M$ stages at the transmitter are delayed by $(B - i)M$ stages at the receiver. All symbols receive a total delay of $(B - 1)M$ stages and the total memory space requirement is $N(B - 1)$ symbols.

A B -by- N convolutional interleaver increases a random error correction code's efficiency as any channel burst of γM symbols affects no more than γ of the B register output sequences at a time. The deinterleaver spaces these sequences to give bursts of γ symbols spaces by $(B - \gamma)M$ error free symbols.

3.4.2 Implementation

A 12-by-204 convolutional interleaver processes the error-protected packets, as depicted in Figure 3.7, to create the interleaved frame of overlapping error-protected packets, as depicted in Figure 3.9.



Fig. 3.9. Interleaved Frames

To preserve the 204-byte periodicity the sync bytes are routed to the $i = 1$ shift register, which corresponds to a null delay, as depicted in Figure 3.10. Similarly the first byte following the sync byte is routed to the $i = 2$ shift register, resulting in a 204-symbol delay, the second byte following the sync byte is routed to the $i = 3$ shift register, resulting in a 408-byte delay, etc. . .

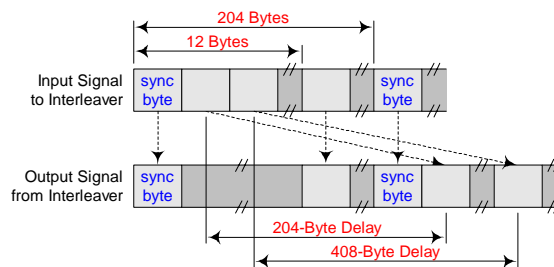


Fig. 3.10. Encoder Implementation

The deinterleaver routes the first recognised sync byte to the $i = 1$ shift register, which corresponds to a 1224-symbol delay and a total interleaver-deinterleaver delay of 1224 symbols. Similarly the first byte following the sync byte is routed to the $i = 2$ shift register, resulting in a 2004-symbol delay and a total interleaver-deinterleaver delay of 1224 symbols, the second byte following the sync byte is routed to the $i = 3$ shift register, resulting in a 1836-byte delay and a total interleaver-deinterleaver delay of 1224 symbols, etc. . .

3.5 Convolutional Coding

Convolutional encoding [37] is a form of FEC that is used in data transmission to counter random errors. Convolutional codes operate on serial data at up to a couple of bits at a time.

Convolutional codes are defined by their code rate, $\frac{k}{n}$, and constraint length, K , where k is the input length, n is the output length and K is the number of k -bit stages in the encoder. Code generator polynomials define the interconnections within the encoder.

Viterbi is the chosen form of decoding algorithm as it has a fixed decoding time, which makes it well suited for hardware implementation.

3.5.1 Convolutional Encoding

A simple convolutional coder is depicted in Figure 3.11. For each bit arriving at C two bits are transmitted, at X and Y , therefore $\frac{k}{n} = \frac{1}{2}$. Each input bit passes through three stages in consecutive clock cycles, C to B to A , therefore $K = 3$. Y is the sum of A and C and X is the sum of A , B and C , therefore the code generator polynomials are 101_2 and 111_2 .

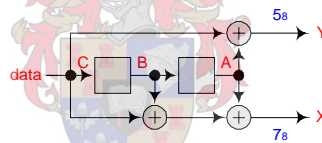


Fig. 3.11. Convolutional Encoder

The state diagram for this circuit is depicted in Figure 3.12.

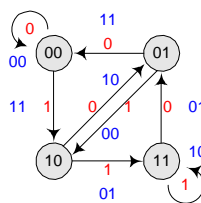


Fig. 3.12. State Diagram

The transitions from each state, BA , are depicted in the trellis diagrams in Figure 3.13.

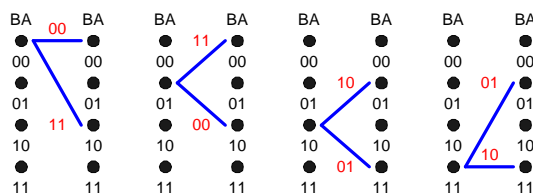


Fig. 3.13. Trellis Diagram of Possible States

Encoding Example

The convolutional encoder in Figure 3.11 encodes the data stream 100111011_2 . The circuit starts in the state $BA = 00_2$. When the first bit ('1') arrives $CBA = 100_2$ produce the symbol $XY = 11_2$. When the circuit is clocked the new state $BA = CB = 10_2$. Similarly when the second bit ('0') arrives $CBA = 010_2$ produce the symbol $XY = 10_2$. When the circuit is clocked the new state $BA = CB = 01_2$.

This process is repeated for each input data bit, with extra zeros appended to the data stream to flush the circuit back to the state $BA = 00_2$, as depicted in Figure 3.14. The output data stream carves a distinct path through the trellis from which the original data stream can be extracted.

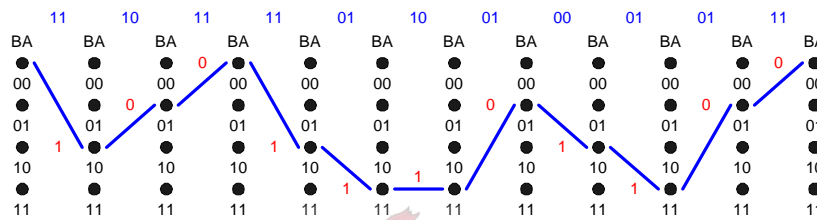


Fig. 3.14. Trellis Diagram

3.5.2 Convolutional Decoding

Upon reception the data is recovered by identifying the path through the trellis defined by the symbol sequence. If there are no errors in the incoming symbols, the process of decoding is readily accomplished.

Error-Free Decoding Example

Figure 3.13 is used to create a trellis diagram from the output stream of the encoding example in Section 3.5.1. For example the first symbol 11_2 represents either the data bit '1' on a path from state 00_2 to state 10_2 or the data bit '0' on a path from state 01_2 to state 00_2 . When the possibilities for all the received symbols are shown together, as depicted in Figure 3.15, a distinct path through the trellis is identified and gives the correct data stream, as depicted in Figure 3.14. This path is continuous as there are no errors in the incoming symbols.

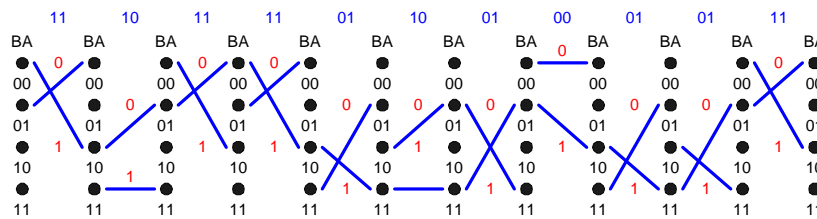


Fig. 3.15. Trellis Diagram of Possible Paths

If there are errors, a decoding technique is required to identify the path through the trellis. Viterbi decoding is the chosen method for hardware implementation.

The Viterbi decoder builds up a trellis stage-for-stage of all the possible paths. Each path has an error allocated to it, the difference between the path's symbol and the received symbol. When two paths arrive at the same state the one with the higher cumulative error is discarded.

After a finite amount of symbols the path that gives the lowest cumulative error is assumed to correspond to the initial data stream. The cumulative error is equal to the number of errors that were encountered and corrected along this path.

Error-Correcting Decoding Example

The data stream 1110100101100100010111_2 is received, which has two errors when compared to the output stream of the encoding example in Section 3.5.1.

Two paths are defined leading off from the initialised state $BA = 00_2$. The initial error is the difference between the received symbol, 11_2 , and the symbol associated with the new path segment, as depicted in Figure 3.16.

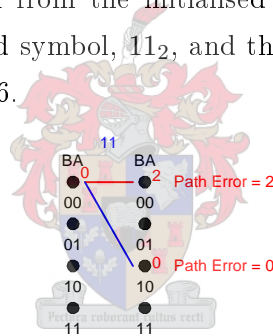


Fig. 3.16. Step One

Four paths are defined leading to the next stage, two from each state in Step One. The accumulated error is increased by the difference between the received symbol, 10_2 , and the symbol associated with the new path segment, as depicted in Figure 3.17.

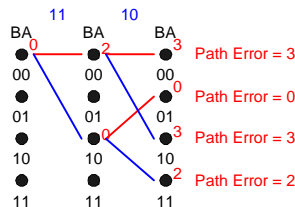


Fig. 3.17. Step Two

Eight paths are defined leading to the next stage, two from each state in Step Two. The accumulated error is increased by the difference between the received symbol, 10_2 , and the symbol associated with the new path segment. Where two paths arrive at the same state, the one with the lower cumulative error is retained and the other is discarded, as depicted in Figure 3.18.

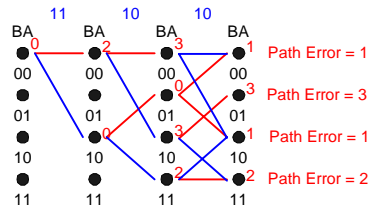


Fig. 3.18. Step Three

In the following steps the four best paths are retained and the four non-productive paths are discarded. Once the trellis has been traversed, the path that gives the lowest cumulative error is assumed to correspond to the correct data, as depicted in Figure 3.19. The cumulative error is equal to the number of errors that were encountered and corrected along this path.

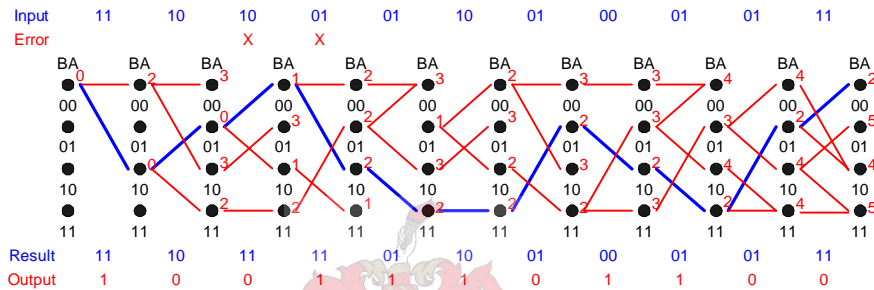


Fig. 3.19. Trellis Diagram of Viterbi Decoder

If two paths arrive at the same state with equal cumulative errors either one is retained, but if this ends up being the path with the lowest cumulative error it is possible that there is an error in the recovered data stream.

3.5.3 Implementation

A convolutional encoder with code rate $\frac{k}{n} = \frac{1}{2}$, constraint length $K = 7$ and code generator polynomials 133_8 and 171_8 , as depicted in Figure 3.20, encodes the interleaved frames, as depicted in Figure 3.9.

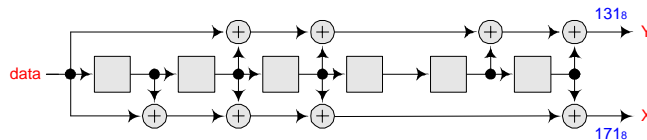


Fig. 3.20. Convolutional Encoder

A Viterbi decoder capable of automatically detecting the code rate is used. The Viterbi decoder's output is aligned into bytes by detecting the sync byte, 47_{16} , within the data stream, which is the first byte in a RS codeword. The data is inverted if a 180° phase shift is detected in the framed byte stream.

3.6 Puncturing

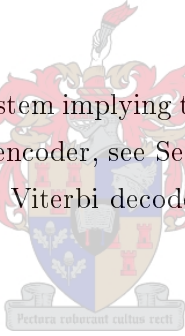
The convolutional encoded data can be punctured to increase the data rate, with the range for a $\frac{k}{n} = \frac{1}{2}$ code rate, $K = 7$ constraint length convolutional encoder defined in Table 3.1, where a ‘1’ indicates a transmitted bit and a ‘0’ indicates a non-transmitted bit.

TABLE 3.1
PUNCTURED CODE DEFINITION

Code Rate				
1/2	2/3	3/4	5/6	7/8
X:1 Y:1	X:10 Y:11	X:101 Y:110	X:10101 Y:11010	X:1000101 Y:1111010
$I = X_1$ $Q = Y_1$	$I = X_1Y_2Y_3$ $Q = Y_1X_3Y_4$	$I = X_1Y_2$ $Q = Y_1X_3$	$I = X_1Y_2Y_4$ $Q = Y_1X_3X_5$	$I = X_1Y_2Y_4Y_6$ $Q = Y_1Y_3X_5X_7$

3.6.1 Implementation

The $\frac{k}{n} = \frac{1}{2}$ code rate is used in this system implying that no puncture architecture is implemented and the output of the convolutional encoder, see Section 3.5, is directly mapped to the input of the modulator, see Section 3.8. The Viterbi decoder is capable of automatically detecting the puncturing pattern.



3.7 Baseband Shaping for Modulation

To decrease inter-symbol interference between channels and to comply with existing regulations concerning outer-band power transmission it is necessary to limit the transmitted power to the allocated frequency band. The baseband square root raised cosine filter [38, 39] is defined by

$$H(f) = \begin{cases} 1 & |f| < f_N(1 - \alpha) \\ \left(\frac{1}{2} + \frac{1}{2} \sin \frac{\pi}{2f_N} \left[\frac{f_N - |f|}{\alpha}\right]\right)^{\frac{1}{2}} & f_N(1 - \alpha) \leq |f| < f_N(1 + \alpha) \\ 0 & |f| > f_N(1 + \alpha) \end{cases} \quad (3.35)$$

where $f_N = \frac{1}{2T_S}$ is the Nyquist frequency and $\alpha = 0.35$ is the roll-off factor.

3.7.1 Implementation

No filtering is implemented in the transmitter, which leads to an imbalance between the transmitter and receiver as the receiver contains a complementary filter.

3.8 Phase Shift Keying Modulation

Phase Shift Keying (PSK) modulation [40, 41] varies the phase of a cosine carrier while keeping the amplitude and frequency constant and is defined as

$$s(t) = A \cos(\omega_c t + \phi_i) \quad (3.36)$$

where $\phi_i = (2i + 1) \frac{\pi}{M}$, $i = 1, 2, \dots, M - 1$, $M = 2^n$ and $n \geq 1$. The smaller the phase shifts, ϕ , the more bits, x , that are transmitted per phase change. Therefore PSK can increase the bit rate without increasing the bandwidth.

3.8.1 Implementation

The modulators [13] perform Quadrature Phase Shift Keying (QPSK) modulation, which has four possible phases ($\phi_i = 45^\circ, 135^\circ, 225^\circ, 315^\circ$), as depicted in Figure 3.21, each of which convey $n = 2$ bits of information.

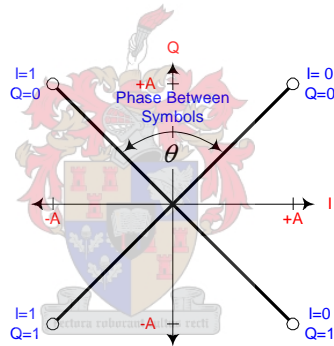


Fig. 3.21. QPSK Constellation

A QPSK demodulator is used that includes the following: Analogue-to-Digital Converters (ADC), imbalance correctors, notch, decimation and matched filters, equalisers, Automatic Gain Controls (AGC), automatic acquisition, fade recovery sequencers, carrier offset estimation and a soft-decision decoder.

3.9 Summary

This chapter discussed the DVB-S standard which is used to protect and modulate the MPEG-2 coded data and shape the transmitted signal. As the system is implemented on three separate boards, devices based on LVDS technology are used to transfer the data between the boards. LVDS technology and its implementation are discussed in the next chapter.

Chapter 4

LVDS

The Low Voltage Differential Signalling (LVDS) standard [9, 10] and the Digital Video Broadcasting - Synchronous Parallel Interface (DVB-SPI) standard [11] are explained in this chapter. This includes the technology fundamentals and how the standards are used to transfer data between the three boards.

4.1 LVDS

LVDS devices transmit high-speed data streams over copper wire. LVDS offers common mode rejection as equally-coupled noise on both signals is ignored. This gain in noise tolerance allows for smaller signal swings, and therefore faster data rates due to short rise-times and fall-times.

4.1.1 Configuration

The LVDS configuration requires a matched external resistor to terminate the receiver input signals to complete the current loop, as depicted in Figure 4.1. This prevents reflections, reduces unwanted electromagnetic emissions and provides the optimum signal quality.

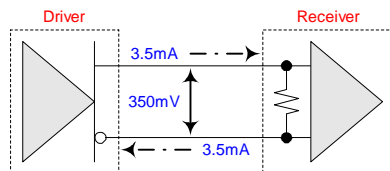


Fig. 4.1. LVDS Driver and Receiver Configuration

The receiver's high input impedance forces the driver's current to flow across the receiver's termination resistor. This will generate either a positive or negative voltage across the receiver's inputs depending on the direction of current flow.

4.1.2 Functional Table

The operation of the LVDS devices are defined in Table 4.1 [25] and Table 4.2 [26]. Failsafe mode places the outputs in a known state when the receiver inputs are under certain fault conditions.

TABLE 4.1
LVDS DRIVER FUNCTIONAL TABLE

Input	Enables		Output	
	EN ⁺	EN ⁻	Positive	Negative
Data	H/X	X/L	Data ⁺	Data ⁻
Open	H/X	X/L	L	H
X	L	H	Z	Z

TABLE 4.2
LVDS RECEIVER FUNCTIONAL TABLE

Input		Enables		Output
Positive	Negative	EN ⁺	EN ⁻	
Data ⁺	Data ⁻	H/X	X/L	Data
Failsafe	Failsafe	H/X	X/L	H
X	X	L	H	Z

4.1.3 Electrical Characteristics

LVDS only defines the driver output and receiver input characteristics, as defined in Table 4.3 and Table 4.4. This allows for its adoption into many applications that complete the interface by specifying the necessary functional specifications, protocols and cable characteristics.

TABLE 4.3
LVDS DRIVER OUTPUT CHARACTERISTICS

Parameter	Description	Min Value	Max Value
V _{OD}	Differential Mode Voltage	247mV	454mV
V _{OS}	Common Mode Voltage	1.125V	1.375V
Z ₀	Impedance		100Ω
t _r /t _f	Rise and Fall Times	0.26ns	1.5ns

TABLE 4.4
LVDS RECEIVER INPUT CHARACTERISTICS

Parameter	Description	Min Value	Max Value
V _{TH}	Threshold Voltage	-100mV	100mV
V _{IN}	Voltage Range	0V	2.4V
Z _{IN}	Input Impedance	90Ω	132Ω

4.1.4 Implementation

A PCB containing two LVDS drivers is mounted onto the FPGA DevBoard and a pair of PCB containing a LVDS receiver each is mounted onto Malan's Board. The layouts are given in Appendix B.2. These LVDS driver-receiver pairs are used to transfer the data between the transmitter's convolutional encoder and modulator via twisted-wire pair cables.

4.2 DVB-SPI

The DVB-SPI standard specifies the synchronous parallel interface for the interconnections of devices within the DVB-S/MPEG-2 format via LVDS devices, as depicted in Figure 4.2.

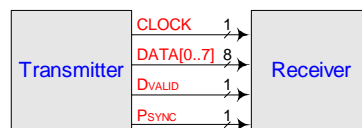


Fig. 4.2. Synchronous Parallel Interface

Both the head-end equipment's physical interface and the signal formats are defined.

4.2.1 Signal Format

The data, control and synchronisation signals are transmitted in parallel and are synchronous to the clock signal, as depicted in Figure 4.3.

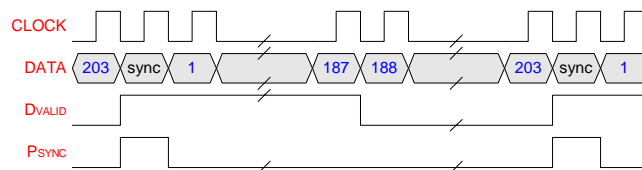


Fig. 4.3. Signal Format

CLOCK is a square wave signal that is active for all 204-bytes of the MPEG-2 TS-Packet.

DATA is the DVB-S decoded data for input to the MPEG-2 decoder.

DVALID is asserted to indicate valid data at the interface. This signal is asserted during the transmission of the sync byte, 47_{16} , and the 187-byte DVB-S payload, but cleared during the transmission of the parity data.

PSYNC is asserted to indicate the beginning of a DVB-S codeword. This signal is asserted during the transmission of the sync byte, 47_{16} .

4.2.2 Mechanical Details

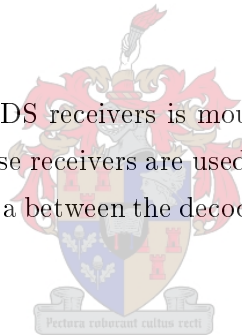
The interface's 25-pin Type-D Connector pin assignments are defined in Table 4.5.

TABLE 4.5
DVB-SPI PIN ASSIGNMENTS

Pin	Signal Line	Pin	Signal Line	Pin	Signal Line
1	Clock ⁺	3	Data ₇ ⁺	16	Data ₇ ⁻
14	Clock ⁻	4	Data ₆ ⁺	17	Data ₆ ⁻
11	D _{VALID} ⁺	5	Data ₅ ⁺	18	Data ₅ ⁻
24	D _{VALID} ⁻	6	Data ₄ ⁺	19	Data ₄ ⁻
12	P _{SYNC} ⁺	7	Data ₃ ⁺	20	Data ₃ ⁻
25	P _{SYNC} ⁻	8	Data ₂ ⁺	21	Data ₂ ⁻
2	Ground	9	Data ₁ ⁺	22	Data ₁ ⁻
13	Ground	10	Data ₀ ⁺	23	Data ₀ ⁻
15	Ground				

4.2.3 Implementation

A PCB board containing three LVDS receivers is mounted onto the FPGA DevBoard with the layout given in Appendix B.2. These receivers are used in conjunction with the DVB-S decoder's DVB-SPI output to transfer the data between the decoder's DVB-S decoder and MPEG-2 decoder via twisted-wire pair cables.



4.3 Summary

This chapter discussed the method of transferring the data between the three boards in the system and thereby completes the communication system. Some form of broadcast method of the transmitted data is required to test this system. The chosen data and broadcast methods are discussed in the next chapter.

Chapter 5

Audio and Video

The processes involved in displaying images and broadcasting audio are explained in this chapter. This includes explanations of the components used to display images and broadcast audio and how to control them. The images are used to show any signal degradation in the system caused by interference.

5.1 Video

The images are displayed on a Video Graphics Adapter (VGA) screen with 640-by-480 resolution, 16-bit colour quality and 60Hz refresh rate. A video Digital-to-Analogue Converter (DAC) [19] converts the digital image data into analogue signals, as depicted in Figure 5.1.

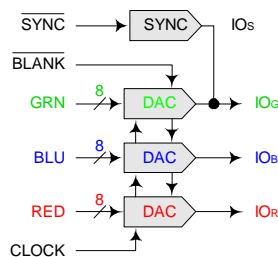


Fig. 5.1. Video DAC Block Diagram

5.1.1 VGA Screen

The VGA standard [42] uses separate wires to transmit the three colour signals and four synchronisation signals to co-ordinate the display. VGA video is a stream of frames made up of a series of horizontal lines, with each line made up of a series of pixels. The lines in each frame are transmitted from top to bottom, and the pixels in each line are transmitted from left to right. Each pixel is made up of the three colours of red, green and blue.

BLANK forces the outputs to their blanking level and **SYNC** forces the green output to a special synchronisation level below the normal blanking level. The position of the Cathode Ray Tube (CRT) is control by the synchronisation signals V_{SYNC} and H_{SYNC} , as depicted in Figure 5.2.

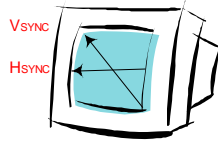


Fig. 5.2. CRT control

Horizontal Sweep

Each horizontal sweep takes $32\mu\text{s}$, as depicted in Figure 5.3. H_{SYNC} is cleared for $3.84\mu\text{s}$ to indicate a new line and asserted for the remainder of the line. H_{EN} is cleared for a further $1.92\mu\text{s}$. H_{EN} remains asserted for $25.6\mu\text{s}$, corresponding to 640 pixels, and then is cleared. H_{SYNC} is cleared $0.64\mu\text{s}$ later.

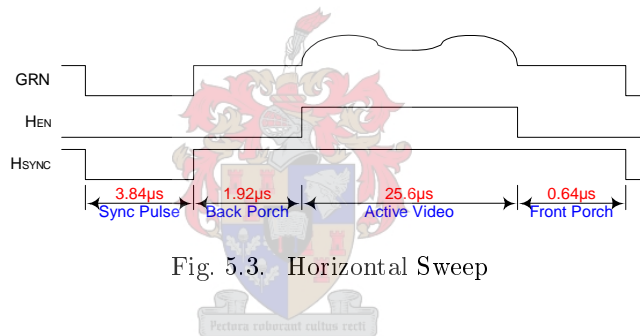


Fig. 5.3. Horizontal Sweep

Vertical Sweep

Each vertical sweep takes 16.8ms , corresponding to a 60Hz refresh rate, as depicted in Figure 5.4. V_{SYNC} is cleared for $64\mu\text{s}$ to indicate a new frame and asserted for the remainder of the frame. V_{EN} is cleared for a further 1.056ms . V_{EN} remains asserted for 15.36ms , corresponding to 480 lines, and then is cleared. V_{SYNC} is cleared $320\mu\text{s}$ later.

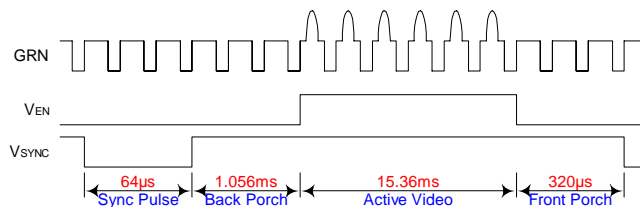


Fig. 5.4. Vertical Sweep

When both H_{EN} and V_{EN} are asserted the corresponding pixel is transmitted to the video DAC. The mechanism that controls the correct addressing of the pixels is explained in Section 6.3.3.

5.2 Audio

The audio is broadcast by a stereo audio coder-decoder (codec) [20] with 44.1kHz sample rate and 16-bit resolution. The codec contains a DAC to convert the digital audio data into analogue signals, as depicted in Figure 5.5.

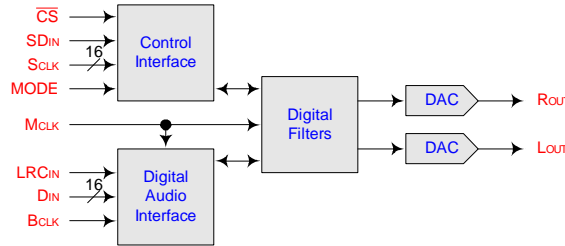


Fig. 5.5. Stereo Audio Codec Block Diagram

5.2.1 Control Interface

The codec registers are programmed using a serial peripheral interface (SPI). SD_{IN} carries the serial data, S_{CLK} is the serial clock and \overline{CS} latches the data word into the codec. Starting with the MSB the data bits are latched on the rising edge of S_{CLK} after which a rising edge on \overline{CS} latches the data word into the codec, as depicted in Figure 5.6.

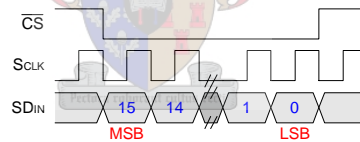


Fig. 5.6. SPI Timing

The register values are defined in Table 5.1.

TABLE 5.1
AUDIO CODEC REGISTER ASSIGNMENT

Address	Register	Assignment	Description
0_{16}	Input Volume Control	107_{16}	-24dB volume
2_{16}	Headphone Volume Control	$1E1_{16}$	-24dB volume
4_{16}	Analogue Audio Path Control	012_{16}	DAC enabled
5_{16}	Digital Audio Path Control	004_{16}	44.1kHz de-emphasis filter
6_{16}	Power Down Control	000_{16}	Power all functions
7_{16}	Digital Audio Interface Format	001_{16}	16-bit, left-aligned data
8_{16}	Sample Rate Control	020_{16}	44.1kHz sampling rate
9_{16}	Digital Interface Activation	001_{16}	Activates the interface
F_{16}	Reset Register	000_{16}	Triggers reset

5.2.2 Digital Audio Interface

The two DACs are controlled by a left-justified interface. D_{IN} is the DAC input, B_{CLK} is the serial clock and LRC_{IN} latches the data word into the DAC. Starting with the MSB the data bits are latched on the rising edge of B_{CLK} after which a falling edge on LRC_{IN} latches the left channel data word into the DAC and a rising edge latches the right channel data word into the DAC, as depicted in Figure 5.7.

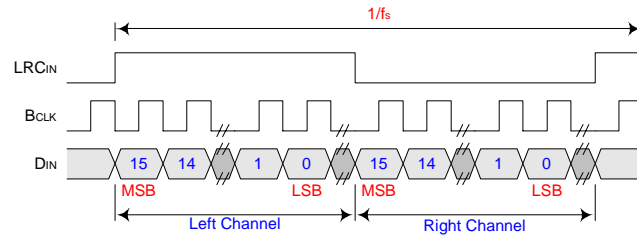


Fig. 5.7. Left-Justified Mode Timing

5.3 Summary

This chapter describes the chosen method to display images and broadcast audio. The images show the signal degradation by displaying the original images permanently stored in the FPGA's internal memory next to the decoded images being continually refreshed in the external memory. The process of storing the images and audio is discussed in the next chapter.



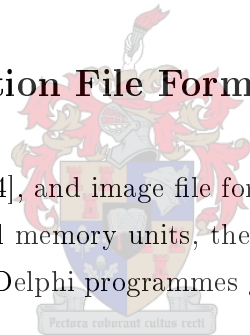
Chapter 6

Memory

The types of memory used within the system are discussed in this chapter. This includes the internal memory units within the FPGA and the external memory units connected to the FPGA. The mechanism for initialising audio and video files in the internal memory units is also discussed.

6.1 Memory Initialisation File Format

The audio file format, wave [43, 44], and image file format, bitmap [45, 46], are not compatible for initialisation of FPGA internal memory units, therefore they are converted to hexadecimal object file format [47] by the two Delphi programmes given on the attached CD.



6.1.1 Bitmap File Format

The bitmap file format consists of a 14-byte file header, a 40-byte information header and a 9600-byte payload of video data, as depicted in Figure 6.1.

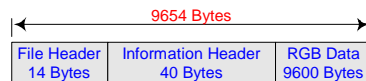


Fig. 6.1. Bitmap File Format

The file header contains information about the file, the information header describes the image and the Red-Green-Blue (RGB) data is the image data. The video data is stored in 24-bit blocks equally divided between red, green and blue data.

The video data is stored in consecutive equal-length scan-lines from left to right and from bottom to top, which is rearranged to top to bottom for usage within the FPGA internal memory units. The 24-bit blocks are reduced to 16-bit blocks for usage with the 16-bit VGA screen.

6.1.2 Wave File Format

The wave file format consists of a 12-byte RIFF chunk, a 24-byte format chunk and a 22920-byte payload of audio data, as depicted in Figure 6.2.

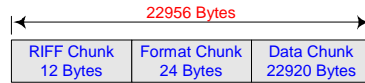


Fig. 6.2. Wave File Format

The RIFF chunk contains information about the file, the format chunk describes the audio format and the data chunk contains the audio data. The audio data is stored in 32-bit blocks equally divided between left and right channel data.

6.1.3 Hexadecimal Object File Format

A hexadecimal object file is blocked into records each containing the following fields, as depicted in Figure 6.3.

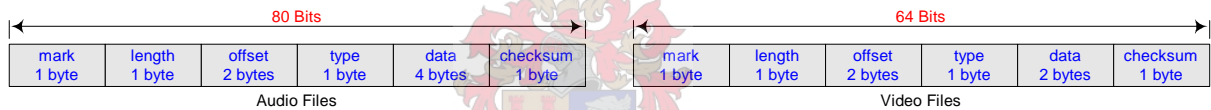


Fig. 6.3. Hexadecimal Object File Format

MARK is assigned the value $3A_{16}$, the ASCII code for the colon (':') character, to indicate the start of a record.

LENGTH specifies the number of info/data bytes in the record. **LENGTH** is assigned the value 02_{16} for video data records, the value 04_{16} for audio data records and the value 00_{16} for end of file records.

OFFSET specifies the 16-bit starting load offset of the data bytes. **OFFSET** is assigned the address of the audio or video data for Data Records and the value 00_{16} for End of File Records.

TYPE specifies the record type. **TYPE** is assigned the value 00_{16} for audio and video Data Records and the value 01_{16} for End of File Records.

INFO/DATA length and contents are determined by the record type. **INFO/DATA** contains a 16-bit image pixel for video Data Records, a 32-bit audio sample for audio Data Records and is non-existent for End of File Records.

CHECKSUM is assigned the two's complement of the result of the addition of the other bytes within the record excluding **MARK**.

6.2 Internal Memory

The FPGA has a 52.5kB embedded memory structure [48] that is implemented as dual-clock First-In, First-Out (FIFO) buffers [49] to transfer the video data between the external memory and the VGA screen and as Read Only Memory (ROM) [50] to store the video data.

6.2.1 ROM Mode

The ROM units are controlled as depicted in Figure 6.4. **Q** is updated to contain the data addressed by **ADDR** at each rising edge of **CLOCK**.

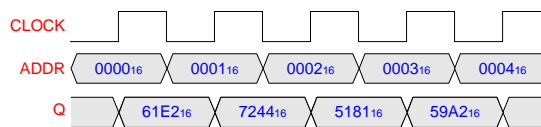


Fig. 6.4. Internal Memory ROM Control

Due to the limited size of the embedded memory structure a 16-bit 80-by-60 image is stored on the FPGA in ROM for each channel, resulting in a total of 28.125kB. Audio storage is impractical as the remaining embedded memory can only store a total of 0.3 seconds of 16-bit audio.

6.2.2 FIFO Buffer Mode

The FIFO buffers are controlled as depicted in Figure 6.5. **DATA** is read into the FIFO at the first rising edge of **WR_{CLK}** after **WR_{REQ}** is asserted. **Q** is the oldest entry in the FIFO at the first rising edge of **RD_{CLK}** after **RD_{REQ}** is asserted. **A_{CLR}** is used to clear the FIFO.

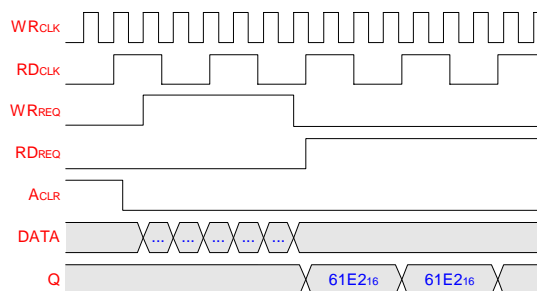


Fig. 6.5. Internal Memory FIFO Buffer Control

Due to the different clock rates of the VGA screen and external memory the FIFO buffers are used as a bridge between the two subsystems. For each channel one line of the 16-bit 80-by-60 image is stored in the FIFO buffer for each horizontal sweep of the VGA screen, resulting in a total of 480B of memory. The buffer is filled up during the display of the source image and emptied to display the final image.

6.3 External Memory

A 256MB Random Access Memory (RAM) module [18] is used to store the received data for later display. A DDR2 SDRAM Controller [51] translates read and write requests into the necessary commands and handles the other aspects of interfacing with the external memory module using the FPGA's dedicated external memory interface [52] at a data transfer rate of up to 334Mbps.

6.3.1 Write Process

The write process is depicted in Figure 6.6. \mathbf{WR}_{REQ} is asserted to request a write operation. \mathbf{SIZE} contains the burst size and \mathbf{ADDR} contains the start address. These signals remain constant until the controller asserts \mathbf{READY} , indicating that the request has been accepted. \mathbf{W}_{DATA} and the data masking control, \mathbf{BE} , are provided one clock cycle after the controller asserts \mathbf{WD}_{REQ} .

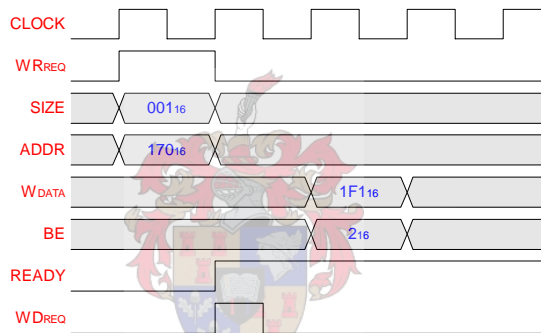


Fig. 6.6. Write Process

6.3.2 Read Process

The read process is depicted in Figure 6.7. \mathbf{RD}_{REQ} is asserted to request a read operation. \mathbf{SIZE} contains the burst size and \mathbf{ADDR} contains the start address. These signals remain constant until the controller asserts \mathbf{READY} , indicating that the request has been accepted. \mathbf{R}_{DATA} is valid for one clock cycle after the controller has asserted $\mathbf{RD}_{\text{VALID}}$.

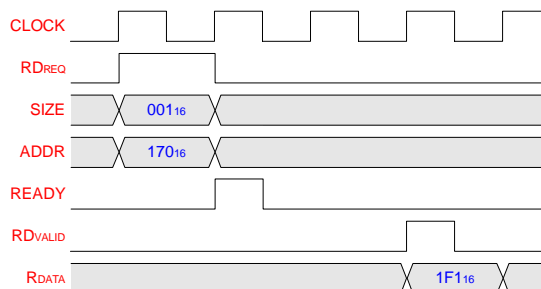


Fig. 6.7. Read Process

6.3.3 Synchronisation

The external memory, VGA screen and MPEG-2 decoder are clocked at different rates. Therefore some form of buffering process is required to facilitate the data flow between these components. The possibility of concurrent read and write commands to the external memory is also investigated.

FIFO Buffer Control

The FIFO Buffers are used to transfer the image data between the clock rates of the external memory and VGA screen, as depicted in Figure 6.8.

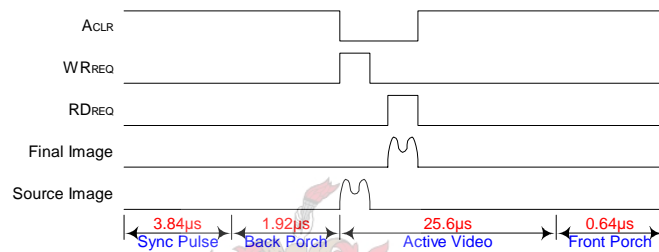


Fig. 6.8. FIFO Buffer Transfer

During the display of the internal memory's image the VGA controller supplies the starting address of an image's line, which the external memory controller writes pixel for pixel to the FIFO Buffers. During the display of the external memory's image the VGA controller reads the image's line pixel for pixel into the video DAC.

MPEG-2 Decoder Control

The external memory control is used to transfer the image data between the clock rates of the external memory and the MPEG-2 decoder, as depicted in Figure 6.9.

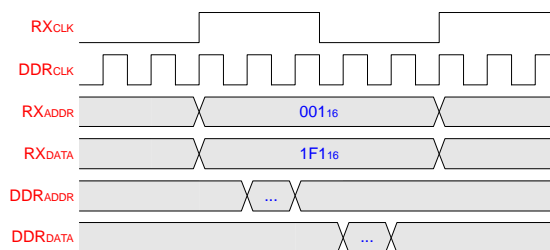


Fig. 6.9. MPEG-2 Decoder Output Transfer

The MPEG-2 decoder supplies $\mathbf{RX_DATA}$ to be stored at $\mathbf{RX_ADDR}$, which the external memory controller temporarily stores until the external memory is ready to be written to.

If concurrent read and write commands to the external memory are performed the possibility of data contention arises, as depicted in Figure 6.10. $\mathbf{RD_REQ}$ and $\mathbf{WR_REQ}$ are simultaneously asserted, with $\mathbf{RD_ADDR}$ and $\mathbf{WR_ADDR}$ to be written concurrently to the external memory's \mathbf{ADDR} . This leads to ambiguity surrounding the assertion of \mathbf{READY} .

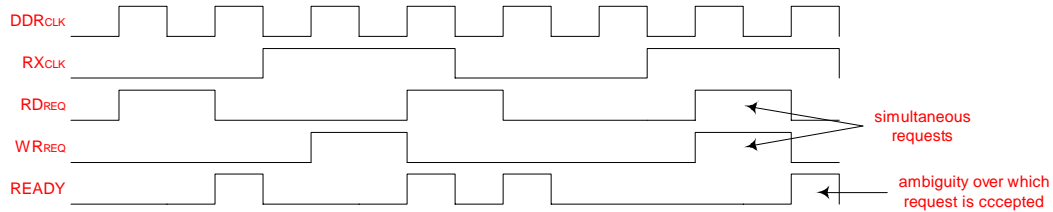


Fig. 6.10. Read and Write Contention

Figure 6.11 illustrates a possible repercussion of $\mathbf{RD_REQ}$ having precedence over $\mathbf{WR_REQ}$. Two images are transmitted and data contention occurs during the storage of the second image. As not all of the second image pixels are written to memory and are therefore lost, the displayed image contains sections of the first image.



Fig. 6.11. Read and Write Contention with Read Commands having Precedence

Figure 6.12 illustrates a possible repercussion of $\mathbf{WR_REQ}$ having precedence over $\mathbf{RD_REQ}$. One image is transmitted and data contention occurs during the display of the image. As $\mathbf{RD_DATA}$ is not updated for each $\mathbf{RD_REQ}$ yet is still written to the FIFO buffers, the displayed image will contain multiples of the same pixel.



Fig. 6.12. Read and Write Contention with Write Commands having Precedence

Therefore data transmission is suspended while displaying images and vice versa to prevent data contention.

6.4 Summary

This chapter discussed the various memory units used to store the image data. The synchronization of these units with the rest of the system is discussed in the next chapter.

Chapter 7

FPGA Implementation

This chapter is used to detail the parts of the FPGA code that are used to control the system and synchronise the various processes.

7.1 Controllers

7.1.1 System Timing Control

A Phase-Lock Loop (PLL) [53] sourced from the on-board 100MHz oscillator [54] has three outputs, as depicted in Figure 7.1. **USR_{CLK}** clocks the user input and output interface components, **SPD_{CLK}** clocks the transmitter components and **VGA_{CLK}** clocks the VGA components.

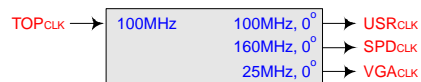


Fig. 7.1. System Timing Controls

7.1.2 External Memory Timing Control

A PLL, sourced from the on-board 100MHz oscillator, has three outputs, as depicted in Figure 7.2, which are the clock inputs for the external memory controller.

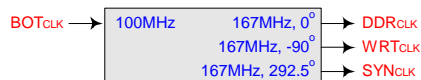


Fig. 7.2. External Memory Timing Control

DDR_{CLK} clocks the majority of the external memory's functions. **WRT_{CLK}** lags **DDR_{CLK}** by 270° to offset **W_{DATA}** and **DM** from **DQS** operations. **SYN_{CLK}** lags **DDR_{CLK}** by 292.5° to offset the resynchronisation processes from **DQS** operations.

7.1.3 User Input Control

The four push-buttons select the FPGA function, as defined in Table 7.1. Data transmission is suspended while the images are being displayed and the screen receives no data while transmission is active to prevent data conflict as the transmission and display components share the same memory space, as discussed in Section 6.3.3.

TABLE 7.1
FPGA PUSH-BUTTON COMMAND TABLE

PB	Command
0001 ₂	Transmit Source Image
0010 ₂	Display Source Image
0100 ₂	Display Final Image
1000 ₂	Display Source and Final Image

The octal Dual In-line Package (DIP) switch selects the transmission speed and channel, as defined in Table 7.2. The selected channel is decoded and stored for future display, while the other channels are ignored.

TABLE 7.2
FPGA DIP-SWITCH COMMAND TABLE

SW _{3..0}	Symbol Rate	SW _{7..5}	Channel
0001 ₂	40Mbaud	001 ₂	One
0010 ₂	20Mbaud	010 ₂	Two
0100 ₂	10Mbaud	100 ₂	Three
1000 ₂	5Mbaud		

7.1.4 User Output Control

The Liquid Crystal Display (LCD) segments display the chosen transmission speed and receiver channel settings and the Light Emitting Diodes (LED) indicate any error encountered by the MPEG-2 decoder, as defined in Table 7.3.

TABLE 7.3
MPEG-2 DECODER ERROR INDICATION

LED	Description	LED	Description
LED ₀	Missing TS-Packet	LED ₃	Missing Video ES
LED ₁	Missing PAT	LED ₄	Missing Video ES
LED ₂	Missing PMT	LED ₅	Missing Audio ES
		LED ₆	Missing Audio ES

7.2 MPEG-2/DVB-S Synchronisation

Synchronisation between the MPEG-2 and DVB-S encoders is implemented on multiple levels. The TS-Packet needs to be built up of its various components, the DVB-S synchronisation requirements need to be met and the various TS-Packets need to be combined to form the TS.

7.2.1 MPEG-2 Initialisation

The initialisation of the MPEG-2 processes to create a PAT, PMT and image is controlled by the signals depicted in Figure 7.3.

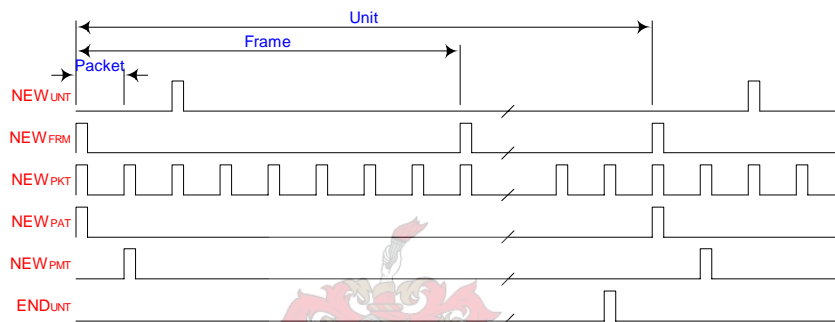


Fig. 7.3. MPEG-2 Initialisation Synchronisation

For each TS-Packet NEW_{PKT} is asserted to update the continuity counter. For every DVB-S frame NEW_{FRM} is asserted to bitwise invert the TS-Packet sync byte. For each unit NEW_{UNT} is asserted to indicate the start of a payload unit, NEW_{PAT} is asserted to indicate the start of a PAT-Section, NEW_{PMT} is asserted to indicate the start of a PMT-Section and END_{UNT} is asserted to indicate that the current TS-Packet contains an AF.

7.2.2 MPEG-2 Activation

The activation of the MPEG-2 processes to create a PAT, PMT and image is controlled by the signals depicted in Figure 7.4.

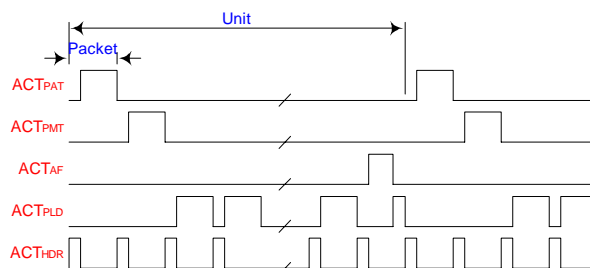


Fig. 7.4. MPEG-2 Activation Synchronisation

For each TS-Packet ACT_{HDR} is asserted to create a TS-Packet header and ACT_{PLD} is asserted to create a payload from the image data read from ROM. For each unit ACT_{PAT} is asserted to create a PAT-Section, ACT_{PMT} is asserted to create a PMT-Section and ACT_{AF} is asserted to create an AF.

7.2.3 DVB-S Initialisation

The initialisation of the DVB-S processes to encode the TS-Packets is controlled by the signals depicted in Figure 7.5.

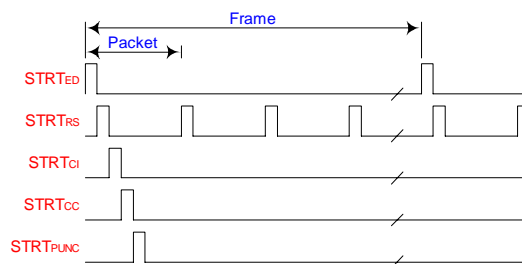
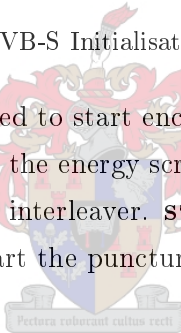


Fig. 7.5. DVB-S Initialisation Synchronisation

For each TS-Packet STRTR_{RS} is asserted to start encoding a new RS codeword. For each DVB-S frame STRTE_{ED} is asserted to initialise the energy scrambler. At the start of transmission STRTC_{CI} is asserted to start the convolutional interleaver. STRT_{CC} is asserted to start the convolutional encoder. $\text{STRTP}_{\text{PUNC}}$ is asserted to start the puncturer.



7.2.4 DVB-S Activation

The activation of the DVB-S processes to encode the TS-Packets is controlled by the signals depicted in Figure 7.6. For each TS-Packet ED_{EN} is asserted to enable the scrambler's output and ED_{OE} is asserted to scramble the scrambler's input.

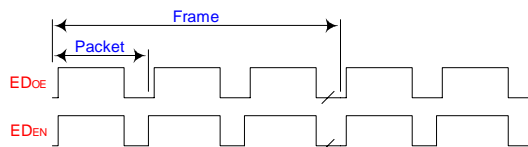


Fig. 7.6. DVB-S Activation Synchronisation

7.3 Summary

This chapter discussed the various control and synchronisation processes required to make the FPGA function. The requirements to get the system functional are discussed in the next chapter.

Chapter 8

Set-Up

The combination and operation of the system's components are discussed in this chapter. This includes the connections between components, the power requirements of the different components and the various settings of these components, as depicted in Figure 8.1.

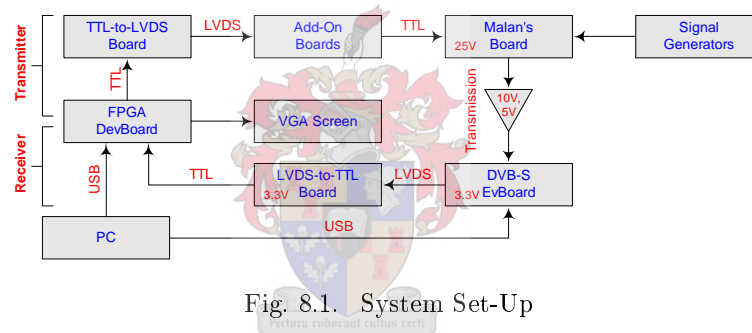


Fig. 8.1. System Set-Up

8.1 Frequency Selection

The frequency ranges of the various components are depicted in Figure 8.2, thereby setting the system's frequency range from 1000MHz to 2000MHz.

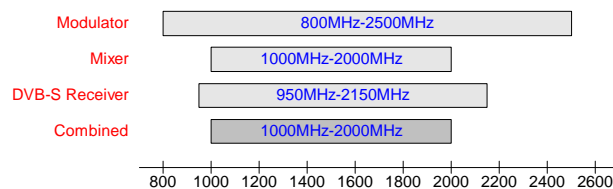


Fig. 8.2. Component Frequency Ranges

Signal generators [55, 56] provide the carrier frequencies for the channels. Channel One is set to the Marconi 2019A's maximum frequency of 1040MHz to fit the channel's maximum bandwidth of 40MHz within the system's frequency range. Channel Two is set to 1120MHz and Channel Three is set to 1200MHz to space the channels double the maximum bandwidth apart.

8.2 Power Requirements

The signal generators and the VGA screen are connected to a 240V, 60Hz power supply. The FPGA DevBoard [17] is connected to a 240V, 60Hz power supply via a power supply adapter. Malan's Board [22] is connected to a 25V DC power supply. The MAR6 amplifier [15] is connected to a 10V DC power supply. The ZX60 amplifier [16] is connected to a 5V DC power supply. The DVB-S EvBoard [21] and LVDS-to-TTL Board are connected to a 3.3V DC power supply.

8.3 Connections

The various components in the system are connected as depicted in Figure 8.3.

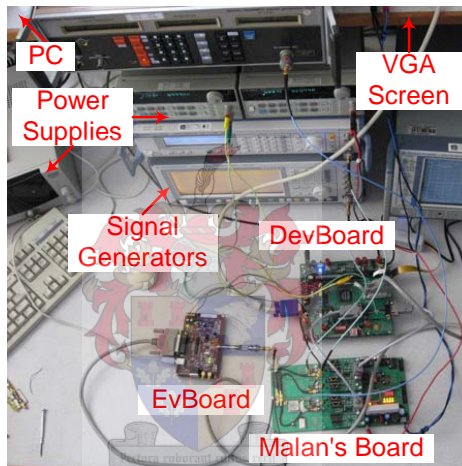


Fig. 8.3. System Connections

The FPGA DevBoard is connected to the VGA screen via connector J₂₁, the Personal Computer (PC) via connector J₉, the TTL-to-LVDS Board via connector J₂₂ and the LVDS-to-TTL Board via connector J₁₅, as depicted in Figure 8.4.

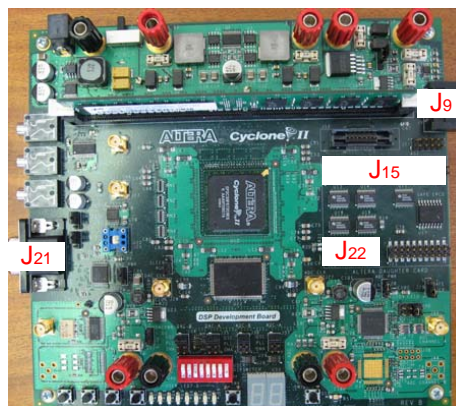


Fig. 8.4. FPGA DevBoard Connections

Malan's Board is connected to the signal generators via connectors LO₂, LO₃ and LO₄ and the Add-On Boards via connectors U₁₆-U₁₉ and U₂₀-U₂₃, as depicted in Figure 8.5.

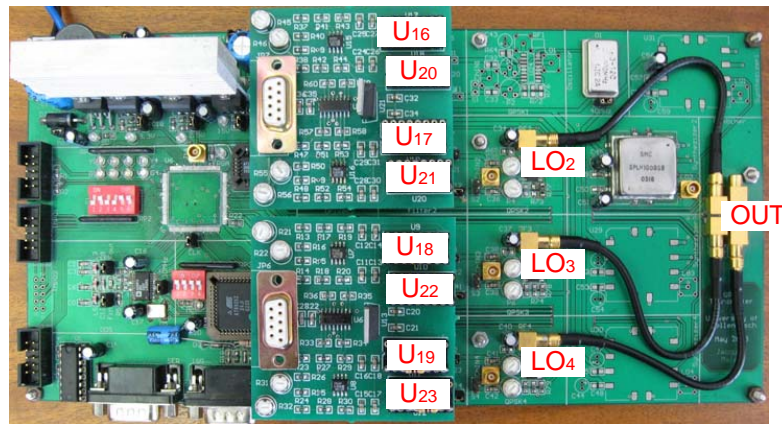


Fig. 8.5. Malan's Board Connections

The DVB-S EvBoard is connected to the PC connector J₇, as depicted in Figure 8.6.

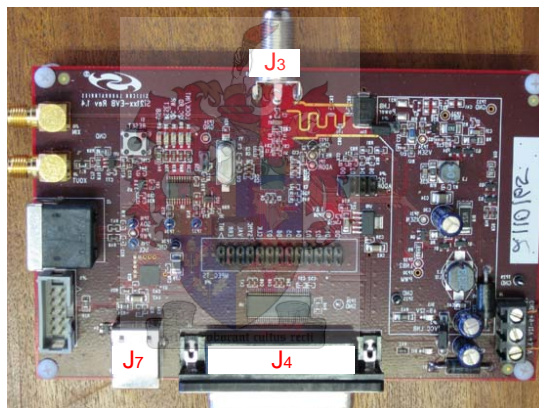


Fig. 8.6. DVB-S EvBoard Connections

The FPGA DevBoard and Malan's Board are connected together via twisted wire pair cables inserted in connectors JP₂ and JP₃ on the TTL-to-LVDS Board and JP₆ and JP₇ on the Add-On Boards, as depicted in Figure 8.7.

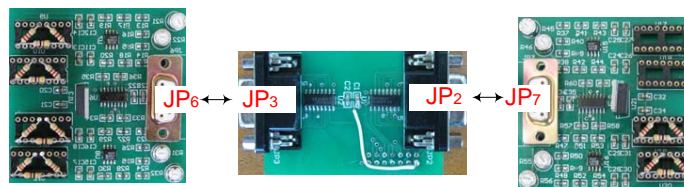


Fig. 8.7. FPGA DevBoard to Malan's Board Connections

Malan's Board and the DVB-S EvBoard are connected via two amplifiers, an attenuator and a DC block inserted between connector OUT on Malan's Board and connector J₃ on the DVB-S EvBoard, as depicted in Figure 8.8.

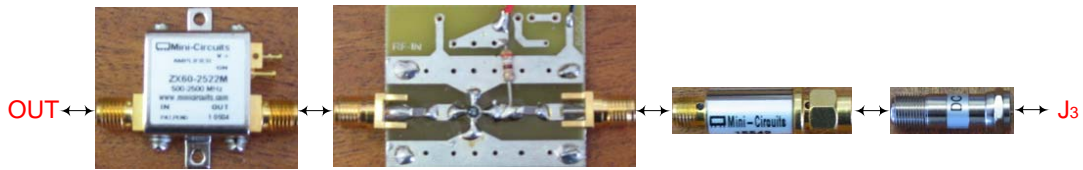


Fig. 8.8. Amplifier and Attenuator Connections

The DVB-S EvBoard and the FPGA DevBoard are connected via a twisted wire pair cable inserted between connector J_4 of the DVB-S EvBoard and connector JP_5 on the LVDS-to-TTL Board, as depicted in Figure 8.6 and Figure 8.9.

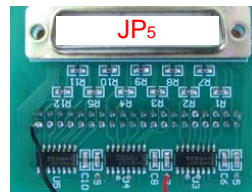


Fig. 8.9. DVB-S EvBoard to FPGA DevBoard Connections

8.4 Software

A PC is connected to the FPGA DevBoard to program the FPGA [12] with the Quartus .sof file and to the DVB-S EvBoard to control the DVB-S Receiver [14] with the Si2110-EVB software.

8.4.1 Quartus Software Set-Up

The following instructions are used to program the FPGA with the .sof file.

- Select **PROGRAMMER** in the **TOOLS** Menu
- Select **HARDWARE SETUP**
 - Select **HARDWARE SETTINGS** Panel
 - * Set **CURRENTLY SELECTED HARDWARE** to **USB-BLASTER**
 - * Select **CLOSE**
- Select **ADD FILE**
 - Select **SYS_TIME_LIMITED.SOF**
 - Select **OPEN**
- Select **PROGRAM/CONFIGURE**
- Select **START**

8.4.2 Si2110-EVB Software Set-Up

The following instructions are used to decode the received RF signal and transmit the decoded data to the FPGA DevBoard via the DVB-SPI Interface. Refer to [14] for more information concerning these parameters.

- Select **EASY START** Panel
 - Set **SYSTEM** to **DVB**
 - Set **MODULATION** to **QPSK**
- Select **BLIND SCAN** Panel
 - Set **SYMBOL RATE** to 5Mbaud-40Mbaud
 - Set **FREQUENCY** to 950MHz-1300MHz
 - Select **START BLIND SCAN**
 - Select the channel corresponding to the DIP-Switch setting defined in Table 7.2 in **CONFIRMED TRANSPONDERS**
- Select **EASY START** Panel
 - Select **TUNE**
- Select **REGISTER READ/WRITE** Panel
 - Set **TS VALID CLOCK EDGE** to **FALLING EDGE**
 - Set **TS CLOCK MODE** to **CONTINUOUS**
 - Set **TS SYNC POLARITY**, **TS VALID POLARITY** and **TS ERROR POLARITY** to **ACTIVE HIGH**
 - Set **TS PARITY GATE** to **ZERO LINES**
 - Set **TS PARALLEL CLOCK SMOOTHER** to **50% DUTY**
 - Set **TS DATA DELAY** and **TS CLOCK DELAY** to **NORMAL**
 - Select **UPDATE TS CONFIGURATION**

8.5 Summary

This chapter discussed how the various components were assembled and initialised in the system. The system needs to go through a series of tests to determine its performance level. These tests involve combining and amplifying the signals and reducing the spacing between channels. The test results are discussed in the next chapter.

Chapter 9

Measurements

The various measurements that are performed on the system are discussed in this chapter. This includes measurements in the time domain and the frequency domain, with an analysis of the system's possible breakdown points.

9.1 Time Domain Measurements

The time domain measurements are performed on an oscilloscope [57] at the points depicted in Figure 9.1, for a symbol rate of 40Mbaud. These measurements look at the distortion of individual signals and the imbalance between associated signals.

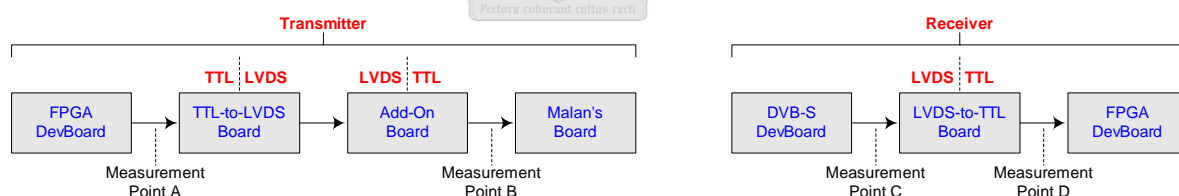


Fig. 9.1. Time Domain Measurement Points

Point A is the reference for the transmitter Q and I signals and is measured at the connections between the FPGA DevBoard and the TTL-to-LVDS Board. Point B is the destination for the transmitter Q and I signals and is measured at the modulator inputs. These measurements show the effect of the transmitter LVDS transmission, the line driver corrections to signal distortion and signal imbalance and the connections between the Add-On Boards and Malan's Board.

Point C is the reference for the DVB-SPI format signals and is measured as TTL signals at header JP_8 of the DVB-S EvBoard. Point D is the destination for the DVB-SPI format signals and is measured at the connections between the LVDS-to-TTL Board and the FPGA DevBoard. These measurements show the effect of the receiver LVDS transmission.

The sequences of $I = 1001_2$ and $Q = 1100_2$, as defined in Equation 9.1 and depicted in Figure 9.2, are used to show the effect of both distortion and imbalance on the modulated signal.

$$I(t) = \begin{cases} -A & |t| < T \\ +A & |t| > T \end{cases} \tag{9.1}$$

$$Q(t) = \begin{cases} -A & t > 0 \\ +A & t < 0 \end{cases}$$

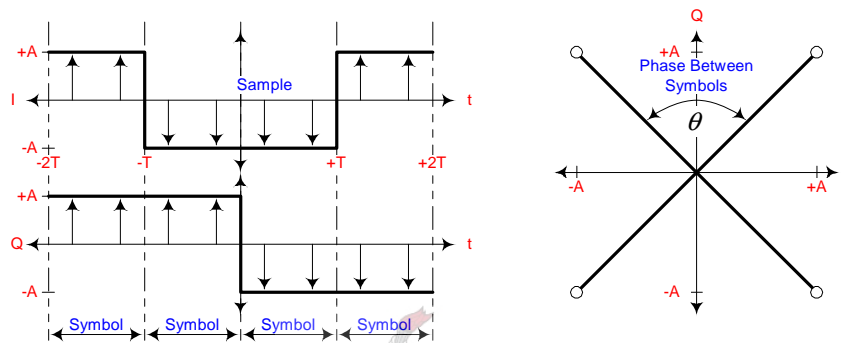


Fig. 9.2. Initial Sequence

9.1.1 Signal Distortion

The physical implementation of the various connections and components within the system cause the digital signals to become distorted from the ideal of a perfect square wave, as depicted in Figure 9.3.

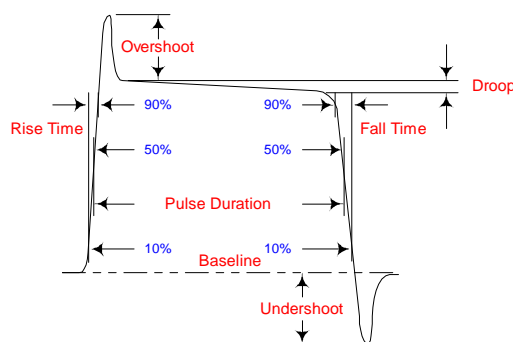


Fig. 9.3. Signal Distortion

Apart from droop, the signal's distortion is transient and only occurs at symbol changes. If the transient period falls between DVB-S Receiver samples the distortion will have no effect on the system. However the possibility exists that the transient period will be sampled by the DVB-S Receiver which would negatively effect the decoding process.

Rise-Time and Fall-Time

A perfect square wave switches instantaneously between symbols, while practical signals have finite rise-times and fall-times, τ , as defined in Equation 9.2. This results in gradual symbol changes.

$$I(t) = \begin{cases} 2Ae^{-\sigma|t+T|} - A & |t| < T \\ A - 2Ae^{-\sigma|t-T|} & |t| > T \end{cases} \quad (9.2)$$

$$Q(t) = \begin{cases} 2Ae^{-\sigma t} - A & t > 0 \\ A - 2Ae^{-\sigma(t+2T)} & t < 0 \end{cases}$$

with $\sigma \approx \frac{\ln(9)}{\tau}$.

The phase difference between samples, ϕ , decreases as the length of the transient, τ , increases, as depicted in Figure 9.4. As ϕ approaches $\phi_{\min} = 0^\circ$ the symbol changes become too gradual for the original data stream to be recovered.

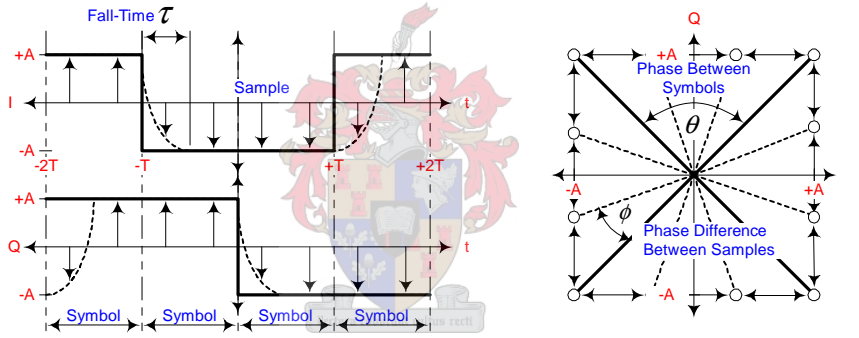


Fig. 9.4. Affects of Rise-Time and Fall-Time Distortion

Overshoot and Undershoot

A perfect square wave settles instantaneously at the desired level, while practical signals have finite overshoot and undershoot, ρ , as defined in Equation 9.3. This results in transient symbol positions for τ -length periods.

$$I(t) = \begin{cases} 2Ae^{-\sigma|t+T|} \left[\cos(\omega_d|t+T|) + \frac{\sigma}{\omega_d} \sin(\omega_d|t+T|) \right] - A & |t| < T \\ A - 2Ae^{-\sigma|t-T|} \left[\cos(\omega_d|t-T|) + \frac{\sigma}{\omega_d} \sin(\omega_d|t-T|) \right] & |t| > T \end{cases} \quad (9.3)$$

$$Q(t) = \begin{cases} 2Ae^{-\sigma t} \left[\cos(\omega_d t) + \frac{\sigma}{\omega_d} \sin(\omega_d t) \right] - A & t > 0 \\ A - 2Ae^{-\sigma(t+2T)} \left[\cos(\omega_d(t+2T)) + \frac{\sigma}{\omega_d} \sin(\omega_d(t+2T)) \right] & t < 0 \end{cases}$$

with $\ln(\rho) = -\frac{\pi\zeta}{\sqrt{1-\zeta^2}}$, $\omega_n = \frac{1.8}{\tau}$, $\sigma = \zeta\omega_n$ and $\omega_d = \omega_n\sqrt{1-\zeta^2}$.

The phase deviation, ϕ , increases as the magnitude of the overshoot and undershoot, ρ , increases, as depicted in Figure 9.5. As ϕ approaches $\phi_{\max} = \pm 45^\circ$, the phase difference between symbols, θ , approaches $\theta = 0^\circ$ and $\theta = 180^\circ$ and the point is reached where the original data stream is irrecoverable.

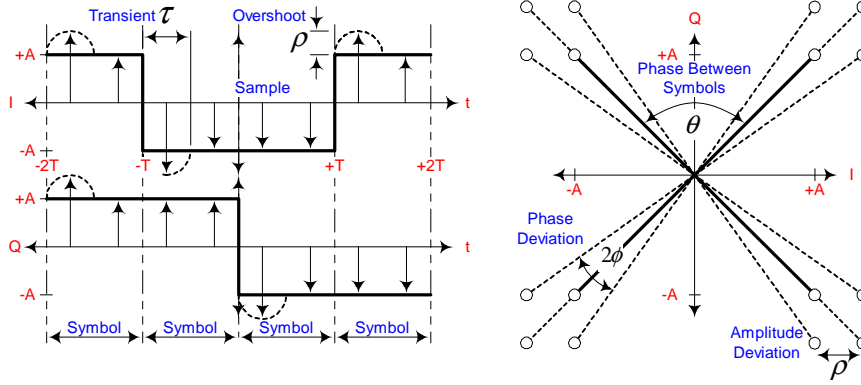


Fig. 9.5. Affects of Overshoot and Undershoot Distortion

Droop

A perfect square wave remains at the desired level for the whole symbol, while real signals have a finite droop, ρ , as defined in Equation 9.4. This results in shifting symbol positions for T -length periods.

$$I(t) = \begin{cases} A - \rho [e^{\sigma(t+2T)} - 1] & t < -T \\ \rho [e^{\sigma(t+T)} - 1] - A & t < 0 \\ \rho [e^{\sigma t} - 1] - A & t < T \\ A - \rho [e^{\sigma(t-T)} - 1] & t < 2T \end{cases} \quad (9.4)$$

$$Q(t) = \begin{cases} A - \rho [e^{\sigma(t+2T)} - 1] & t < -T \\ A - \rho [e^{\sigma(t+T)} - 1] & t < 0 \\ \rho [e^{\sigma t} - 1] - A & t < T \\ \rho [e^{\sigma(t-T)} - 1] - A & t < 2T \end{cases}$$

with $\sigma = \frac{\ln(2)}{T}$.

The phase deviation, ϕ , increases as the magnitude of the droop, ρ , increases, as depicted in Figure 9.6. As ϕ approaches $\phi_{\max} = \pm 45^\circ$, the phase difference between symbols, θ , approaches $\theta = 0^\circ$ and $\theta = 180^\circ$ and the point is reached where the original data stream is irrecoverable.

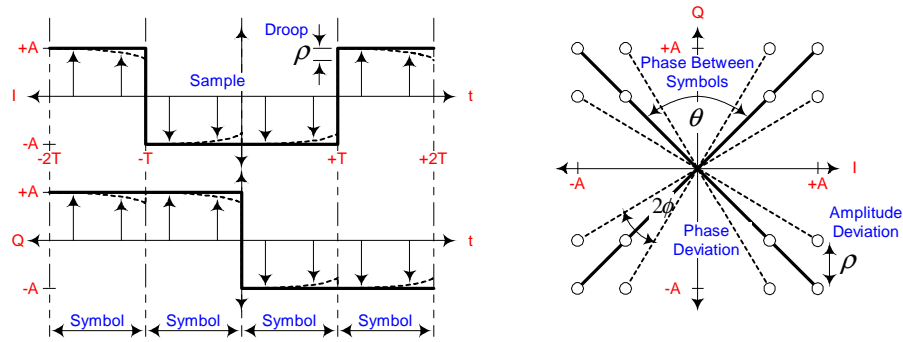


Fig. 9.6. Affects of Droop Distortion

Measurements

In the transmitter the LVDS components maintain the signals' shape and the line driver potentiometers are adjusted to reduce the overshoot and undershoot, as depicted in Figure 9.7.

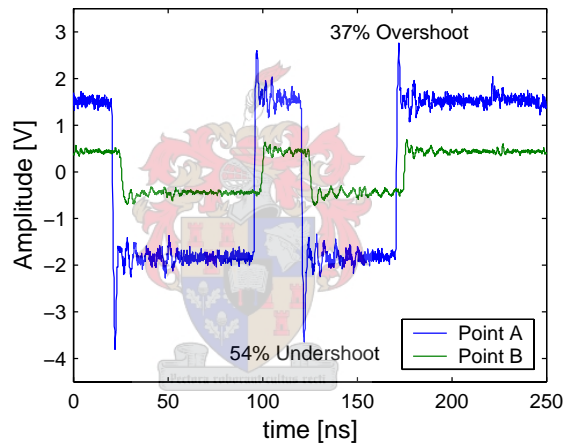


Fig. 9.7. Comparison between Transmitter Distortion Measurements

In the receiver the LVDS components maintain the signal's shape, as depicted in Figure 9.8.

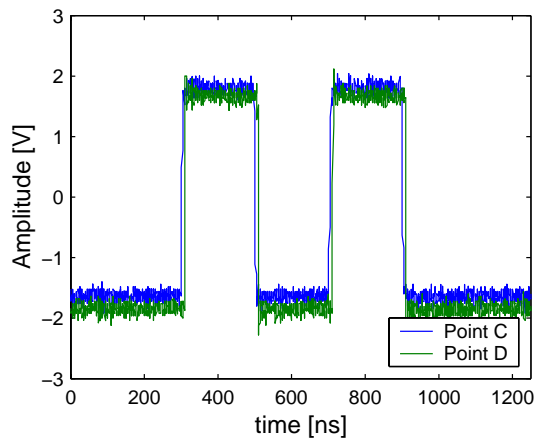


Fig. 9.8. Comparison between Receiver Distortion Measurements

9.1.2 Signal Imbalance

The physical implementation of the various connections and components within the system cause the paired signals to become imbalanced from the ideal of perfectly matched signals, as depicted in Figure 9.9.

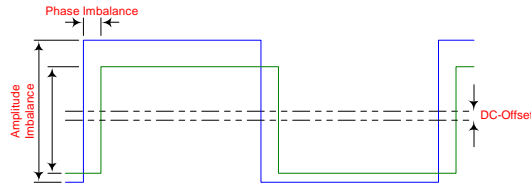


Fig. 9.9. Signal Imbalance

The effects of the three kinds of signal imbalances are investigated below.

Amplitude Imbalance

Perfectly matched signals have equal amplitudes, while in practice there is an amplitude imbalance between the signals, ρ , as defined in Equation 9.5. This results in expanded symbol positions for $\rho > 0$ and contracted symbol position for $\rho < 0$.

$$\begin{aligned}
 I(t) &= \begin{cases} -(A + \rho) & |t| < T \\ +(A + \rho) & |t| > T \end{cases} \\
 Q(t) &= \begin{cases} -A & t > 0 \\ +A & t < 0 \end{cases}
 \end{aligned} \tag{9.5}$$

The phase deviation, ϕ , increases as the magnitude of the amplitude imbalance, ρ , increases, as depicted in Figure 9.10. As ϕ approaches $\phi_{\max} = \pm 45^\circ$, the phase difference between symbols, θ , approaches $\theta = 0^\circ$ and $\theta = 180^\circ$ and the point is reached where the original data stream is irrecoverable.

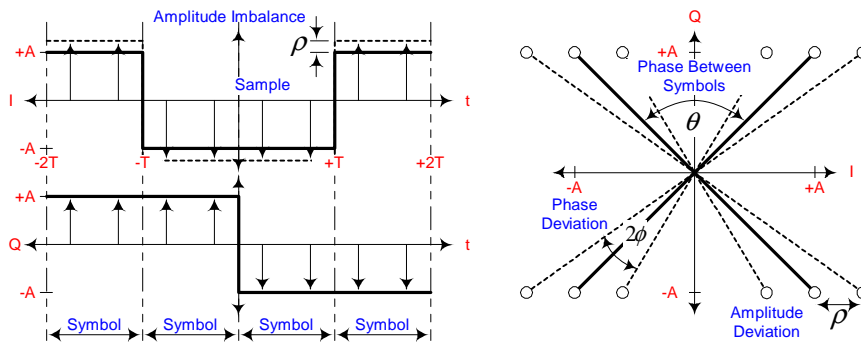


Fig. 9.10. Affects of Amplitude Imbalance

Phase Imbalance

Perfectly matched signals have equal phase, while in practice there is a phase imbalance between the signals, κ , as defined in Equation 9.6. This results in misaligned Q and I sequences.

$$I(t) = \begin{cases} -A & |t| + \kappa < T \\ +A & |t| + \kappa > T \end{cases} \tag{9.6}$$

$$Q(t) = \begin{cases} -A & t > 0 \\ +A & t < 0 \end{cases}$$

The phase imbalance, κ , spreads the Q and I sequences out from one another resulting in incorrect symbol being decoded, as depicted in Figure 9.11, with the possibility of decoding more symbols than are transmitted.

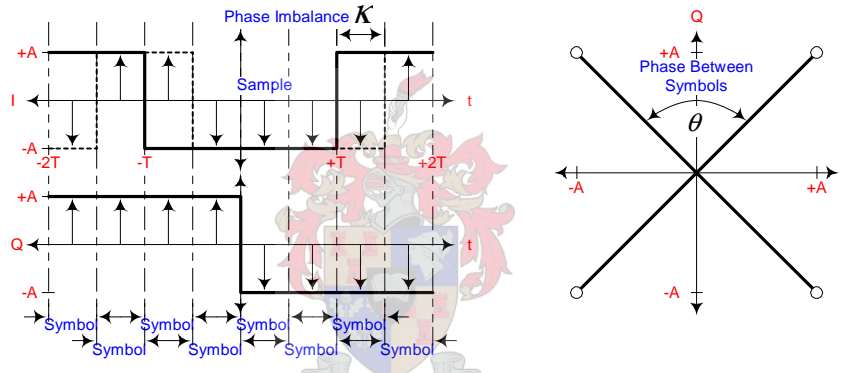


Fig. 9.11. Affects of Phase Imbalance

DC-Offset

Perfectly match signals have equal mean values, while in practice there is a DC-Offset between the signals, ρ , as defined in Equation 9.7. This results in right-shifted symbol positions for $\rho > 0$ and left-shifted symbol position for $\rho < 0$.

$$I(t) = \begin{cases} -A + \rho & |t| < T \\ +A + \rho & |t| > T \end{cases} \tag{9.7}$$

$$Q(t) = \begin{cases} -A & t > 0 \\ +A & t < 0 \end{cases}$$

The phase deviation, ϕ , increases as the magnitude of the DC-Offset, ρ , increases, as depicted in Figure 9.12. As ϕ approaches $\phi_{\max} = \pm 45^\circ$, the phase difference between symbols, θ , approaches $\theta = 0^\circ$ and $\theta = 180^\circ$ and the point is reached where the original data stream is irrecoverable.

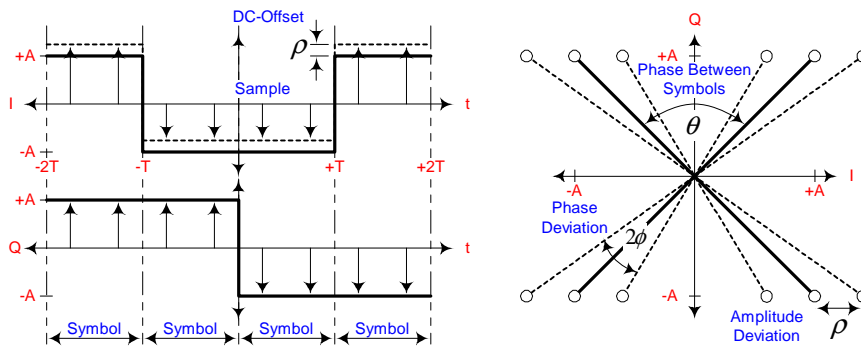


Fig. 9.12. Affects of DC-Offset

Measurements

In the transmitter the line driver potentiometers are adjusted to reduce the amplitude imbalance and the modulator potentiometers are adjusted to reduce the DC-Offset, as depicted in Figure 9.13.

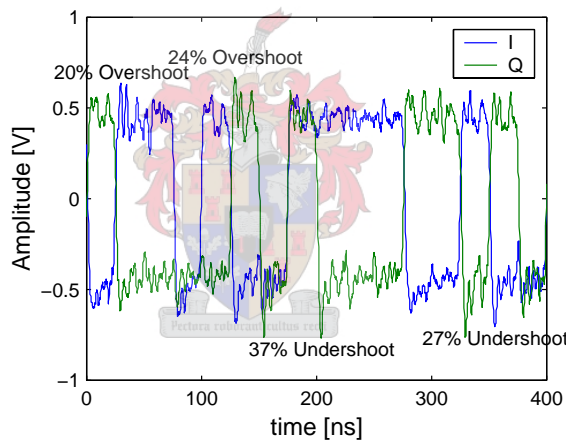


Fig. 9.13. Comparison between Transmitter Imbalance Measurements

9.1.3 Distortion and Imbalance Correction Measures

Apart from the already mentioned potentiometers, the system contains other components that reduce the various forms of distortion and imbalance. The LVDS receiver amplifiers and the line driver amplifiers force the signals to the rail voltages to minimise droop distortion and the DVB-S Receiver contains imbalance correctors, filters, equalisers and gain controls to correct droop distortion, amplitude imbalance and DC-Offset.

However the system does not contain any components that correct rise-time distortion, fall-time distortion and phase imbalance. This is because the DVB-S Receiver is only able to correct errors present in multiple samples and not errors present in singular samples.

9.2 Frequency Domain Measurements

The frequency domain measurements are performed on a spectrum analyser [58] at the points depicted in Figure 9.14, for a symbol rate of 40Mbaud.

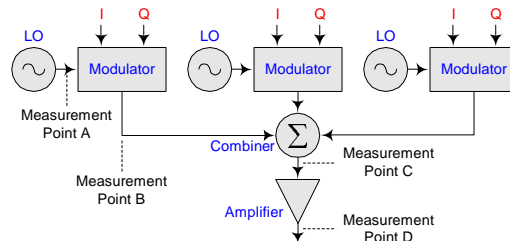
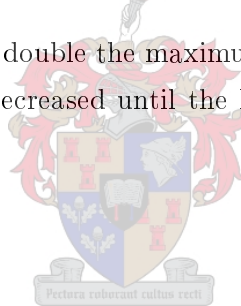


Fig. 9.14. Frequency Domain Measurement Points

The Local Oscillators (LO) are measured at Point A to look at the purity of the carrier and to adjust their power level settings to maximise and balance the modulator outputs. The signal flow of the separate channels and their combination are measured at Point B, Point C and Point D to look at the effect of combining and amplifying the channels and overlapping their bandwidths.

Starting from a channel spacing of double the maximum bandwidth, as discussed in Section 8.1, the channel spacing is gradually decreased until the DVB-S Receiver is no longer able to lock onto and decode the channels.

9.2.1 Local Oscillators



A comparison is made between the on-board synthesizer [23] and the three external signal generators to decide which to use as the LO for the three channels, as depicted in Figure 9.15. The sources are set to a frequency of 1GHz and a power level of +0.6dBm. The on-board synthesizer is rejected as it is the only source with sidebands above the noise floor.

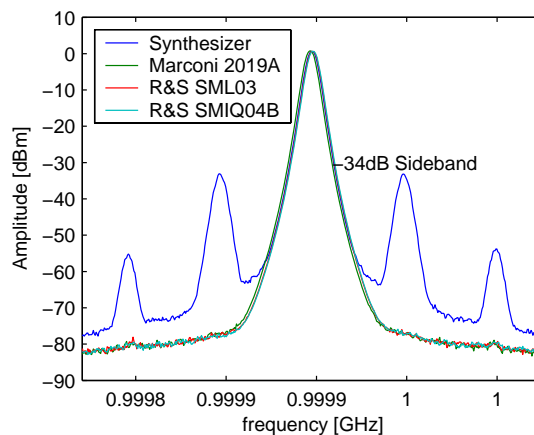


Fig. 9.15. Comparison of Signal Generators

9.2.2 Signal Flow

A combined, unamplified signal with a channel spacing of 80MHz is input to the DVB-S Receiver. This signal contains no intermodulation products and has separated channel bandwidths.

DVB-S Receiver

The Si2110-EVB Software performs a blind scan of the frequency spectrum, as depicted in Figure 9.16. All three are detected with similar power levels and Carrier-to-Noise Ratios (CNR).

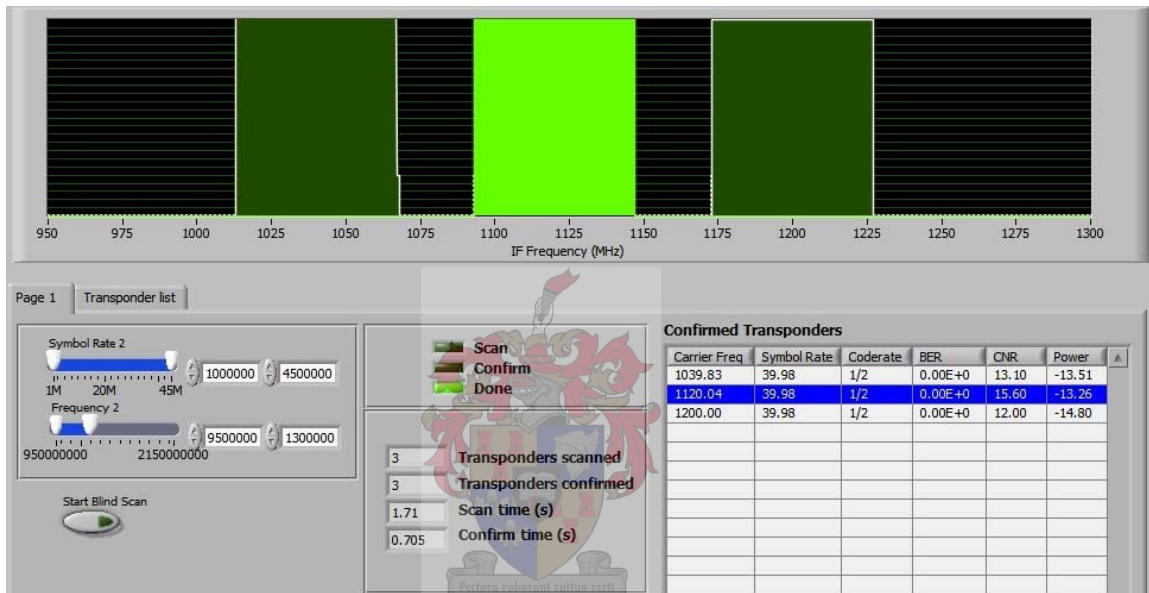


Fig. 9.16. Blind Scan Panel Screenshot for Unamplified 80MHz Channel Spacing

The DVB-S Receiver is able to output an error-free data stream for Channel Two even though the QPSK Bit Error Rate (BER) is high, as depicted in Figure 9.17.

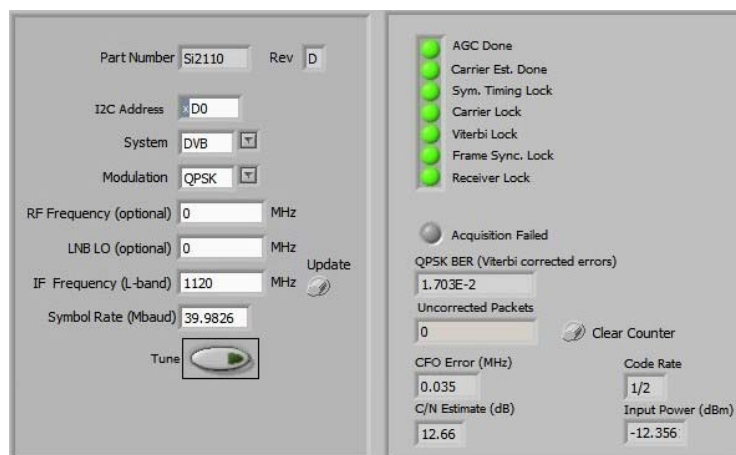


Fig. 9.17. Easy Start Panel Screenshot for Unamplified 80MHz Channel Spacing

Modulators

The modulator outputs are depicted in Figure 9.18.

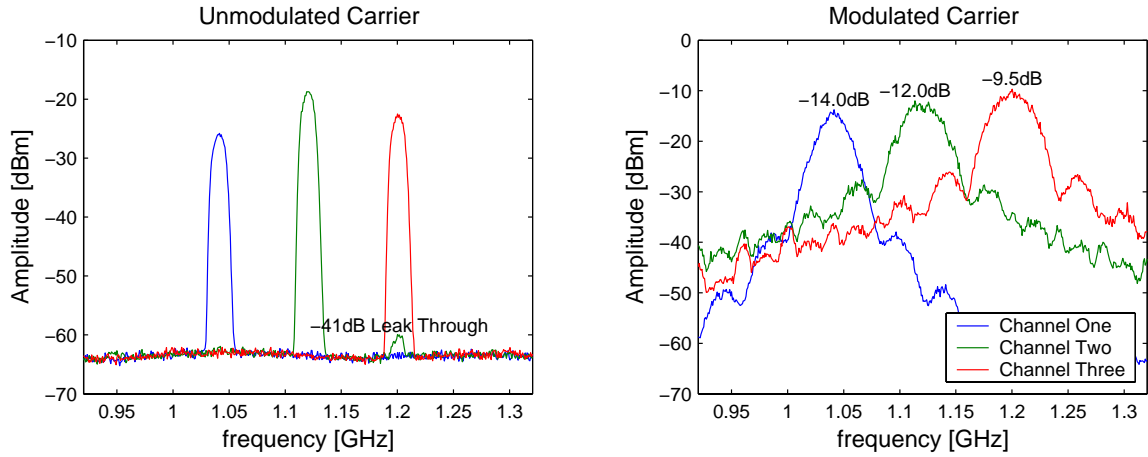
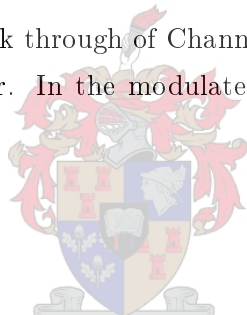


Fig. 9.18. Modulator Outputs for 80MHz Channel Spacing

In the unmodulated graph, the leak through of Channel Three into the input of Channel Two’s modulator is above the noise floor. In the modulated graph, the outputs have an amplitude mismatch of 4.5dB.



Combiner

The combiner’s output for each channel is depicted in Figure 9.19.

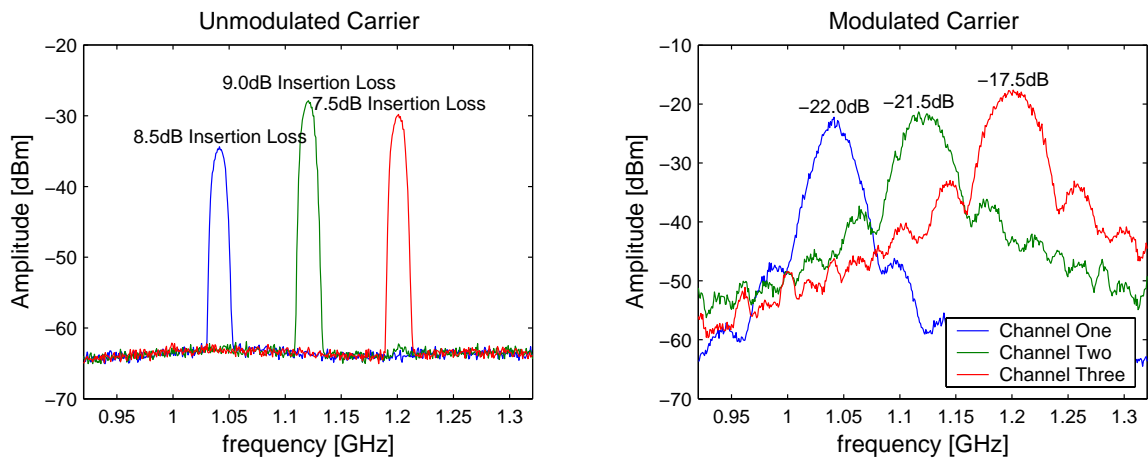


Fig. 9.19. Combiner Outputs for 80MHz Channel Spacing

In the unmodulated graph, the combiner has an insertion loss of between 7.5dB and 9.0dB. In the modulated graph, the input mismatch of the combiner maintains the amplitude mismatch between the channels at 4.5dB.

The combiner's output for all three channels is depicted in Figure 9.20.

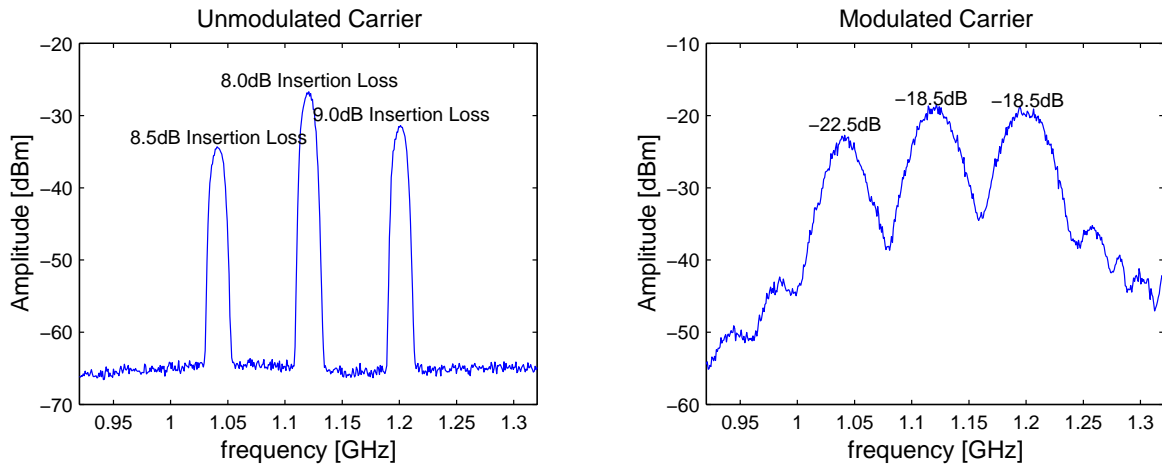
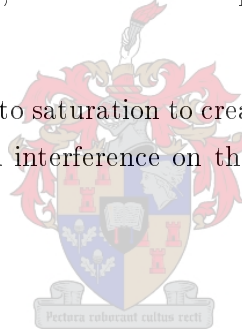


Fig. 9.20. Combiner Output for 80MHz Channel Spacing

In the unmodulated graph, the combination of channels alters the insertion loss to between 8.0dB and 9.0dB. In the modulated graph, the channels are spaced far enough apart to be distinguished from each other.

The combined signal is amplified into saturation to create intermodulation products and spurious signals. The effect of this injected interference on the DVB-S Receiver's ability to decode the signal is investigated.



DVB-S Receiver

The DVB-S Receiver is able to output an error-free data stream for Channel Two even though the QPSK BER is high, as depicted in Figure 9.21.

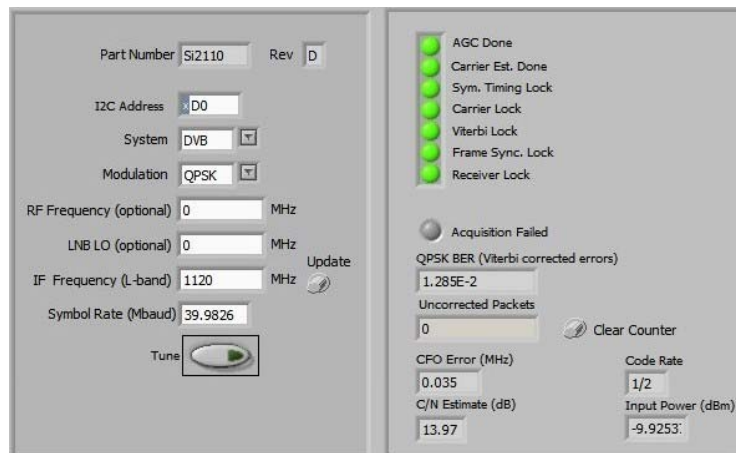


Fig. 9.21. Easy Start Panel Screenshot for Amplified 80MHz Channel Spacing

Amplifier

The amplifier's output for each channel is depicted in Figure 9.22.

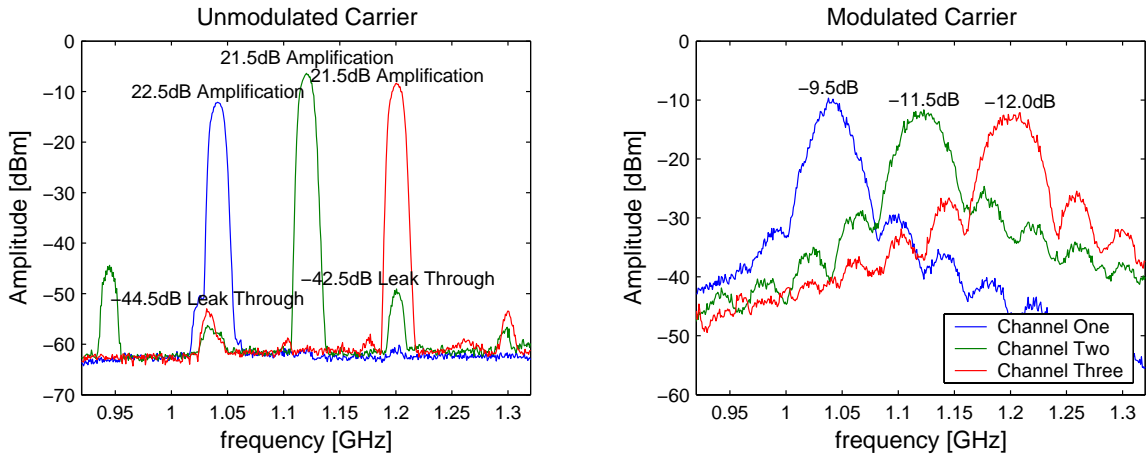


Fig. 9.22. Amplifier Outputs for 80MHz Channel Spacing

In the unmodulated graph, the amplifier raises any leak through above the noise floor and creates spurious signals within the multi-channel band. In the modulated graph, the amplification decreases the amplitude mismatch between the signals to 2.5dB.

The amplifier's output for all three channels is depicted in Figure 9.23.

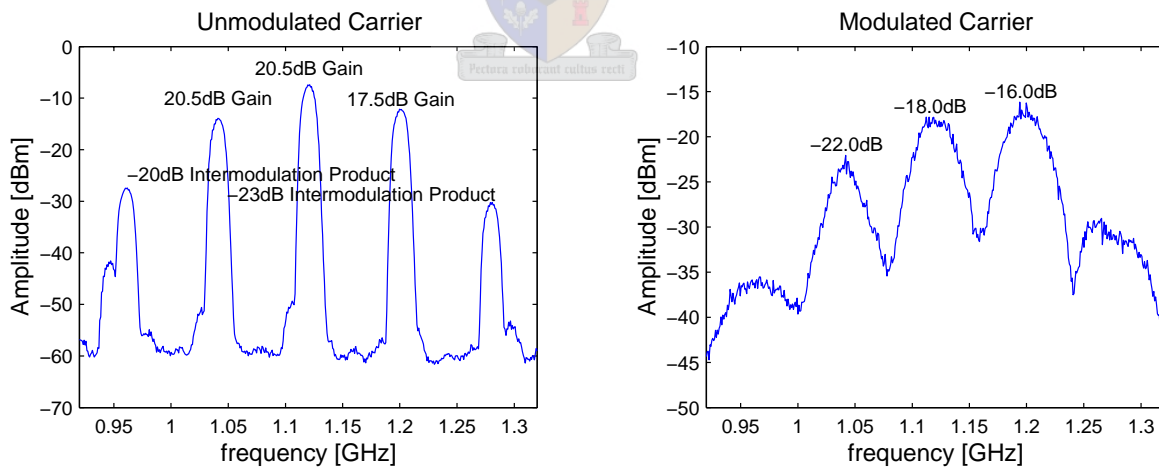


Fig. 9.23. Combined Amplifier Output for 80MHz Channel Spacing

In the unmodulated graph, the amplification of the combined channels creates spurious signals and large intermodulation products within the multi-channel band. In the modulated graph, the intermodulation products and spurious signals have increased the out of band power levels and the combination of the channels increases the amplitude mismatch between the signals to 6.0dB.

The channel spacing is reduced to investigate the effect of overlapping bands. The DVB-S Receiver continues to output an error-free data stream until the channel spacing decreases below 45MHz, where the DVB-S Receiver loses lock of the signal. Measurements are performed with the channel spacing set to 45MHz.

DVB-S Receiver

The DVB-S Receiver is able to output an error-free data stream for Channel Two even though the QPSK BER is high, as depicted in Figure 9.24.

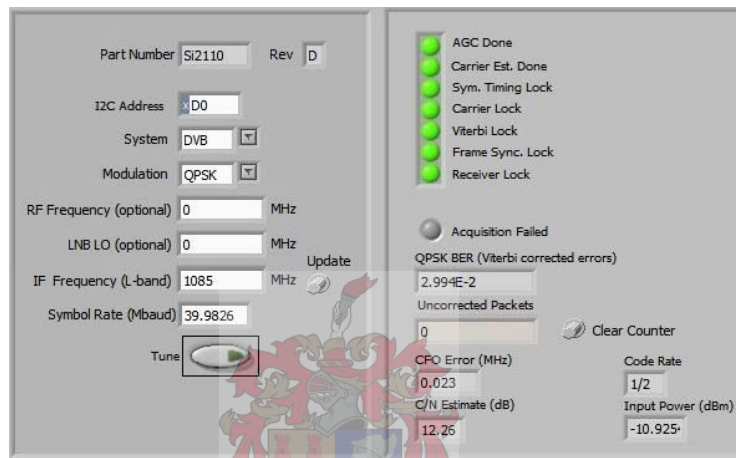


Fig. 9.24. Easy Start Panel Screenshot for Unamplified 45MHz Channel Spacing

Modulators

The modulator outputs are depicted in Figure 9.25.

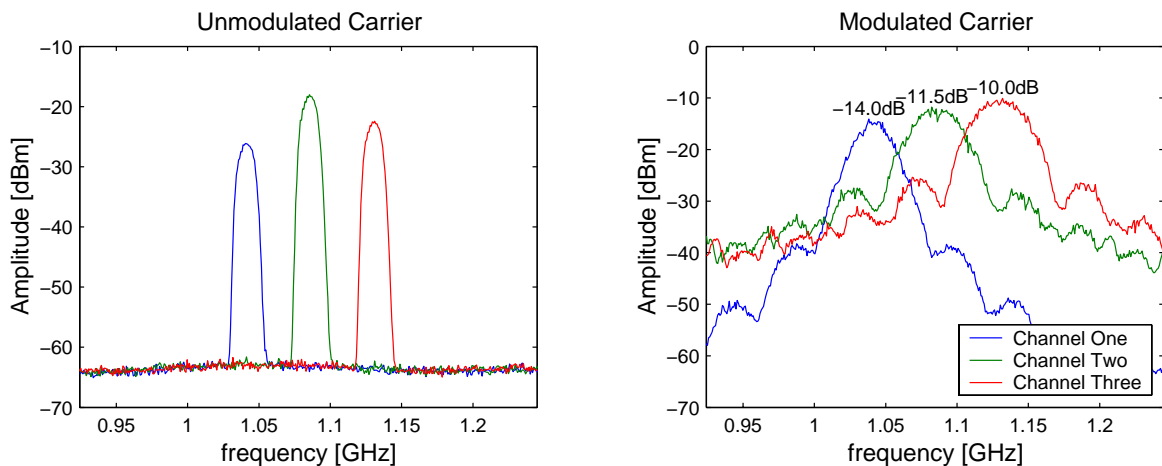


Fig. 9.25. Modulator Outputs for 45MHz Channel Spacing

In the modulated graph, the outputs have an amplitude mismatch of 4.0dB.

Combiner

The combiner's output for each channel is depicted in Figure 9.26.

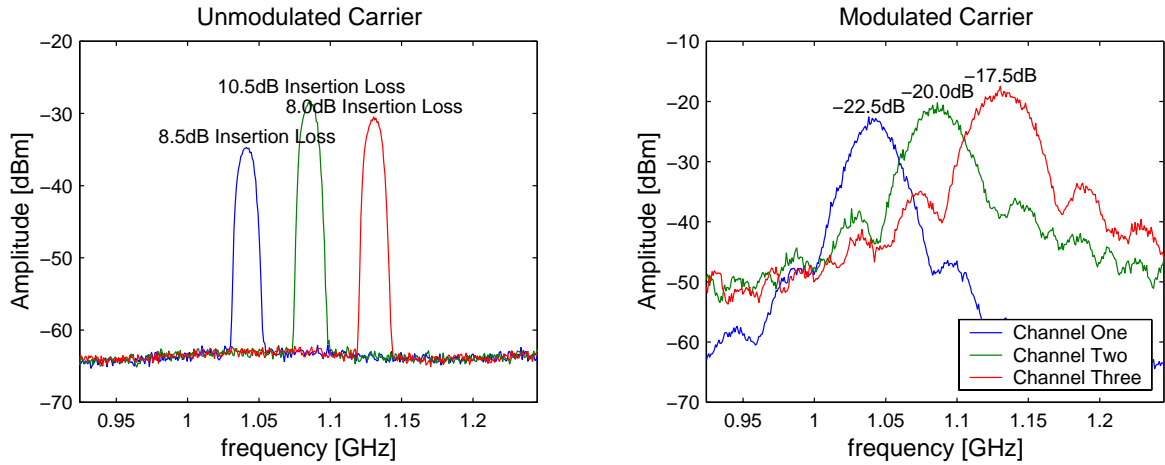


Fig. 9.26. Combiner Outputs for 45MHz Channel Spacing

In the unmodulated graph, the combiner has an insertion loss of between 8.0dB and 10.5dB. In the modulated graph, the input mismatch of the combiner increases the amplitude mismatch between the channels to 5.0dB.

The combiner's output for all three channels is depicted in Figure 9.27.

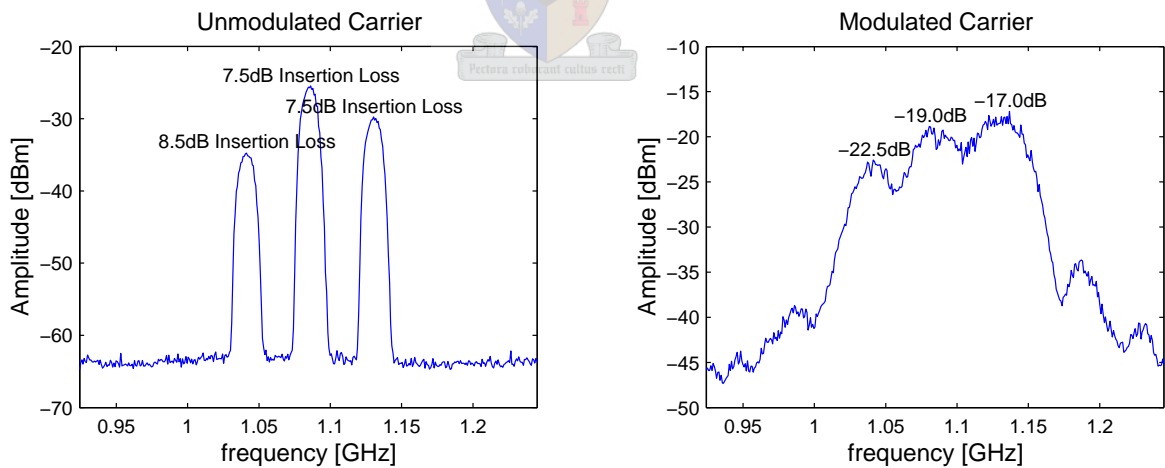


Fig. 9.27. Combiner Output for 45MHz Channel Spacing

In the unmodulated graph, the combination of channels alters the insertion loss to between 7.5dB and 8.5dB. In the modulated graph, the channels are spaced to close together to be distinguished from each other.

The combined signal is amplified to create intermodulation products and spurious signals to investigate the effect of interference on the DVB-S Receiver's ability to decode the signal.

DVB-S Receiver

The DVB-S Receiver is able to output an error-free data stream for Channel Two even though the QPSK BER is high, as depicted in Figure 9.28.

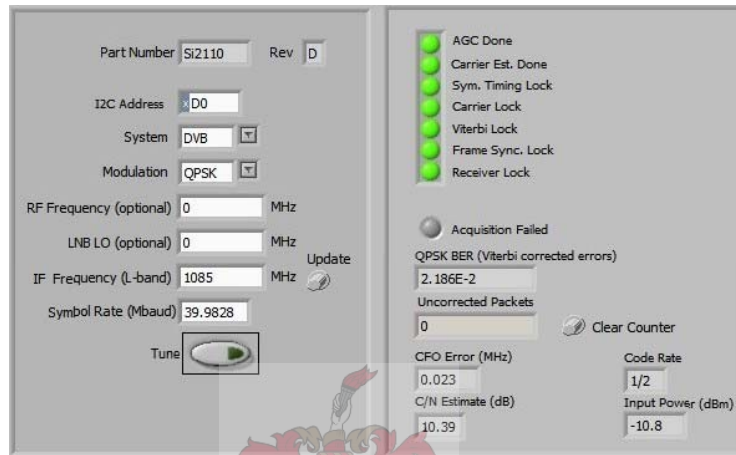


Fig. 9.28. Easy Start Panel Screenshot for Amplified 45MHz Channel Spacing

Amplifier

The amplifier's output for each channel is depicted in Figure 9.29.

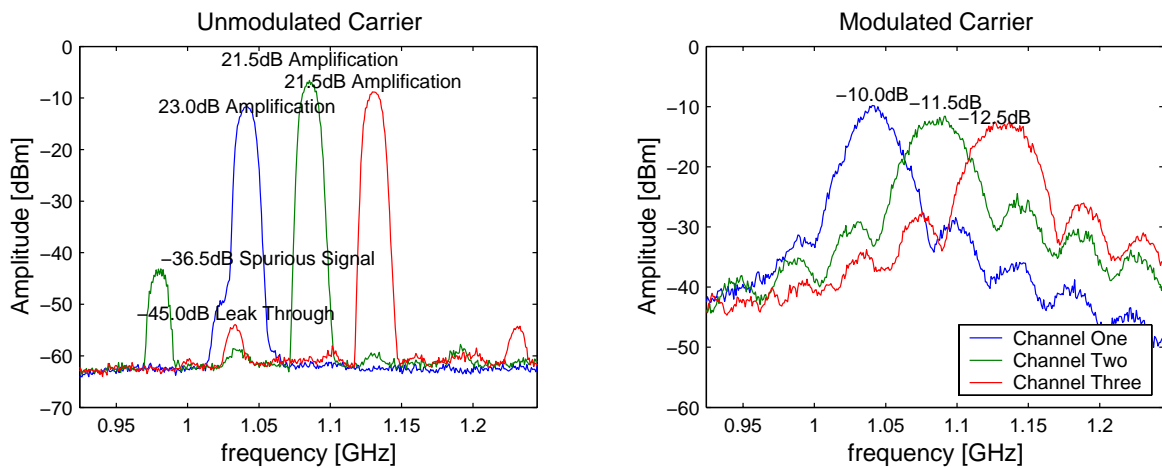


Fig. 9.29. Amplifier Outputs for 45MHz Channel Spacing

In the unmodulated graph, the amplifier raises any leak through above the noise floor and creates spurious signals within the multi-channel band. In the modulated graph, the amplification decreases the amplitude mismatch between the signals to 2.5dB.

The amplifier's output for all three channels is depicted in Figure 9.30.

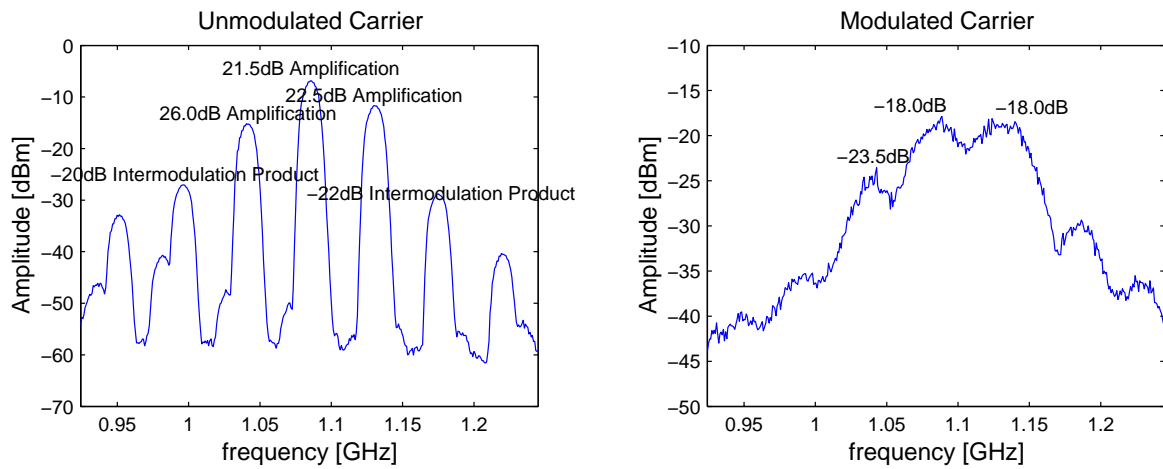


Fig. 9.30. Combined Amplifier Output for 45MHz Channel Spacing

In the unmodulated graph, the amplification of the combined channels creates spurious signals and large intermodulation products within the multi-channel band. In the modulated graph, the intermodulation products and spurious signals have increased the out of band power levels and the combination of the channels increases the amplitude mismatch between the signals to 5.5dB.

Observations

The close proximity of the channels and the amplification of the signals results in a frequency domain littered with spurious signals, intermodulation products and overlapping bandwidths. This noisy environment is indicated by the high QPSK BER.

Yet the DVB-S Receiver is able to output an error-free data stream even though 87.5% of Channel Two's bandwidth is filled with either the main band of Channel One or Channel Three when the channel spacing is set to 45MHz.

The frequency domain contains less spurious signals and no intermodulation products when the channels are amplified separately, thereby implying that the signal quality would be higher if the power amplifiers were part of the parallel configuration in the system.

9.3 Summary

This chapter discussed the measurements performed on the signals to show the DVB-S Receiver's ability to decode an error-free data stream from a signal in a noisy environment. The conclusion and recommendations of the system are discussed in the next chapter.

Chapter 10

Conclusion and Recommendations

10.1 Summary

In this thesis a simplified multi-channel transmission system was developed with off-the-shelf components and software based on existing standards implemented on a FPGA. The system demonstrates that a parallel channel configuration can bypass the technological limitations that impede an increase in the data rate of satellite systems.

The system consists of a packaging mechanism and coding algorithms, with the transfer of data between the components handled by devices based on the LVDS standard. The packaging mechanism is a subset of the Systems part of the MPEG-2 standard and is implemented on a FPGA. The coding algorithms are based on the DVB-S standard, with the digital sections of the encoder implemented on a FPGA, the analogue sections of the encoder performed on Malan's Board and the decoding sections performed on a DVB-S EvBoard.

The system is tested by displaying transmitted images alongside the originals to show any quality degeneration caused by interference encountered in the system and the transmission link.

10.2 Results

The system was tested under various conditions to gauge the implemented standards' capabilities to protect and correct the data stream from the non-idealities inherent in the system and the interference injected into the system.

The tests were performed and documented to demonstrate that the system does work and that it is capable of performing well in non-ideal situations.

10.3 Recommendations

The implemented system serves two purposes.

Firstly, it is a working model of the proposed solution. This proves that the aim of this thesis was met: to design a configuration that bypasses the limit existing devices and regulations place on the system's data rate.

Secondly, it served as the first iteration in the development of a more complex system. The components are not optimised or customised for any specific usage, but are building blocks based on existing standards. Therefore, each section of the system can be tested, modified and improved upon to meet a developer's unique needs.

The system was designed as a measurement and development tool and is not a commercial product. The components in the system that are only useful for testing purposes can be removed. Also, the transmitter's correction of signal distortion and signal imbalance can be automated.

As the system was designed from off-the-shelf components, it is not customised for use within any specific framework. No time was spent on matching networks for specific frequency ranges and no filtering or punctured code were implemented in the transmitter. Incorporation of these components would lead to better signal quality and would result in less interference from surrounding channels.

The parallel configuration can be used to improve the sustainability and efficiency of satellite systems. The incorporation of channel selection would give the option of only activating the required amount of channels to meet a specific download's bandwidth, thereby reducing the power consumption. If any channel malfunctions, its data could be rerouted to the other channels, thereby resulting in a functional system albeit one with a slower data rate.

10.4 Conclusion

The aim of this thesis was to find an affordable and upgradeable method of increasing the data rate of current satellite systems. The compression, modulation and amplification techniques and telecommunication regulatory standards restrict the data rate, which are bypassed by designing a parallel channel configuration. This solution can be implemented with cheaper components and can easily be extended and upgraded to meet the ever increasing need for fast and accurate information.

Bibliography

- [1] R. E. Zeimer and W. H. Tranter, *Principles of Communication: Systems, Modulation & Noise*, chapter 7 - Binary Data Transmission, pp. 329–395, John Wiley & Sons, fifth edition, 2002.
- [2] ICASA, “South African Table of Frequency Allocations,” Tech. Rep., ICASA, 16 August 2005.
- [3] ETSI, “Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for 11/12 GHz Satellite Services,” Tech. Rep. EN 300 421, ETSI, August 1997.
- [4] H. Matsumura and H. Katoh, “Technologies and Services of Digital Broadcasting - Satellite Digital Broadcasting Systems,” *Broadcast Technology*, vol. 18, pp. 12–17, Spring 2004.
- [5] ETSI, “Digital Video Broadcasting (DVB); Implementation Guidelines for the Use of MPEG-2 Systems, Video and Audio in Satellite, Cable and Terrestrial Broadcasting Applications,” Tech. Rep. 154, ETSI, September 1997.
- [6] ISO/IEC, “Generic Coding of Moving Pictures and Associated Audio: Systems,” Tech. Rep. 13818-1, ISO/IEC, 13 November 1994.
- [7] P.A. Sarginson, “MPEG-2: Overview of the Systems Layer,” Tech. Rep. 1996/2, Research & Development Department, BBC, 1996.
- [8] G. Fairhurst, “Electronics Research Group,” Department of Engineering, University of Aberdeen, January 2001, <http://erg.abdn.ac.uk/research.html>.
- [9] Fairchild Semiconductor, “LVDS Fundamentals,” Appl. Note 5017, June 2005.
- [10] National Semiconductor, *LVDS Owner’s Manual*, National Semiconductor, third edition, Spring 2004.
- [11] CENELEC, “Interfaces for CATV/SMATV Headends and Similar Professional Equipment for DVB/MPEG-2 Transport Streams,” Tech. Rep. EN 50083-9, CENELEC, March 1997.

- [12] Altera Corporation, *Cyclone II Device Handbook*, vol. 1, Altera Corporation, November 2005.
- [13] RFMD, “2.5GHz Direct Quadrature Modulator,” Tech. Rep. RF2422, RFMD, 6 February 2003.
- [14] Silicon Laboratories, “Satellite Receiver for DVB-S with QuickLock and QuickScan,” Tech. Rep. Si2110, Silicon Laboratories, June 2006.
- [15] Mini-Circuits, “Broadband Monolithic Amplifiers,” Tech. Rep. MAR-6, Mini-Circuits, 17 August 2004.
- [16] Mini-Circuits, “Broadband, Linear High Isolation Amplifiers,” Tech. Rep. ZX60, Mini-Circuits, 25 May 2006.
- [17] Altera Corporation, *Cyclone II EP2C35 DSP Development Board Reference Manual*, Altera Corporation, May 2005.
- [18] Micron, “DDR2 SDRAM Unbuffered DIMM,” Tech. Rep. MT8HTF3264A, Micron, 2005.
- [19] Fairchild Semiconductor, “Triple Video DAC,” Tech. Rep. FMS3818, Fairchild Semiconductor, December 2005.
- [20] Texas Instruments, “Sereo Audio Codec,” Tech. Rep. TLV320AIC23, Texas Instruments, May 2002.
- [21] Silicon Laboratories, “Evaluation Board for the Si2110 SiRXTM DVB Receiver IC with USB Interface,” Tech. Rep. Si2110USB-EVB, Silicon Laboratories, June 2006.
- [22] J. de Villiers Malan, “Microwave data transmission by means of an optimal bandwidth multichannel qpsk system,” M.S. thesis, University of Stellenbosch, December 2003.
- [23] Synergy, “Integrated Frequency Synthesizer,” Tech. Rep. SPLH1000SB, Synergy, 15 March 2000.
- [24] Mini-Circuits, “Power Splitter/Combiners,” Tech. Rep. SCA-4-20, Mini-Circuits, 15 January 2003.
- [25] Texas Instruments, “High-Speed Differential Line Drivers,” Tech. Rep. SN65LVDS31, Texas Instruments, March 2004.
- [26] Texas Instruments, “High-Speed Differential Line Receivers,” Tech. Rep. SN65LVDS32, Texas Instruments, March 2004.
- [27] Altera Corporation, *Cyclone II Device Handbook*, vol. 1, chapter 11 - High-Speed Differential Interfaces in Cyclone II Devices, pp. 11–1 to 11–18, Altera Corporation, November 2005.

- [28] Altera Corporation, *altlvs Megafunction User Guide*, Altera Corporation, August 2005.
- [29] P. Tudor, "Progressive Transmission of Interlaced Pictures," Tech. Rep. T-1231, Research & Development Department, BBC, August 1993.
- [30] M. Kilgore and J. Wu, "MPEG, the MP3 Standard, and Audio Compression," student project for Mathematics of the Information Age, Stanford University, 16 September 2003.
- [31] D. Pan, "A Tutorial on MPEG/Audio Compression," *IEEE Multimedia Journal*, pp. 60–74, Summer 1995.
- [32] ITU, "Carrier Energy Dispersal for Systems Employing Angle Modulation by Analogue Signals or Digital Modulation in the Fixed-Satellite Service," Tech. Rep. S.446-4, ITU, 1993.
- [33] B. Sklar, *Digital Communications: Fundamentals and Applications*, Prentice-Hall, second edition, 2001.
- [34] S. S. Shah, S. Yaqub, and F. Suleman, "Self-Correcting Codes Conquer Noise Part 2: Reed-Solomon Codecs," *EDN*, pp. 107–120, 15 March 2001.
- [35] Altera Corporation, *Reed-Solomon Compiler User Guide*, Altera Corporation, July 2004.
- [36] S.A. Hanna, "Convolutional Interleaving for Digital Radio Communications," Ottawa, Canada, October 1993, IEEE ICUPC, pp. 443–447.
- [37] R. Tervo, "Error Control Coding," class notes for EE4253: Digital Communications, University of New Brunswick, 8 September 2005.
- [38] N. Sheikholeslami and P. Kabal, "A Family of Nyquist Filters Based on Generalized Raised-Cosine Spectra," Ontario, Canada, June 1998, Biennial Symposium on Communications, pp. 131–135.
- [39] K. Gentile, "The Care and Feeding of Digital, Pulse-Shaping Filters," *RF Design*, pp. 50–61, April 2002.
- [40] S.N. Bhatti, "Digital Modulation, ASK, FSK and PSK," class notes for D51: Basic Communications and Networks, University College London, March 1995.
- [41] D. M. Pozar, *Microwave and RF Design of Wireless Systems*, chapter 9 - Modulation Techniques, pp. 288–335, John Wiley & Sons, 2001.
- [42] N. Ickes, "VGA Video," class notes for 6.111: Introductory Digital Systems Laboratory, Massachusetts Institute of Technology, April 29 2004.

- [43] M. Csele, "WAV File Format," class notes for TEC1631: Digital Signal Processing, Niagara College, 2005.
- [44] S. Wilson, "Microsoft WAVE Soundfile Format," class notes for Music 422: Perceptual Audio Coding, Stanford University, 20 January 2003.
- [45] W. L. Bahn, "Windows BMP Bitmap File Format," class notes for ECE-1021: Engineering Modeling and Methods in C, University of Colorado, 8 August 2004.
- [46] M. Reddy, "MS-Windows Bitmap Format," 27 June 1997, <http://www.martinreddy.net/gfx/2d/BMP.txt>.
- [47] Intel, "Hexadecimal Object File Format Specification," Tech. Rep., Intel, 6 January 1988.
- [48] Altera Corporation, *Cyclone II Device Handbook*, vol. 1, chapter 8 - Cyclone II Memory Blocks, pp. 8-1 to 8-29, Altera Corporation, November 2005.
- [49] Altera Corporation, *Single- & Dual-Clock FIFO Megafunction User Guide*, Altera Corporation, September 2005.
- [50] Altera Corporation, *lpm_rom Megafunction User Guide*, Altera Corporation, March 2005.
- [51] Altera Corporation, *DDR & DDR2 SDRAM Controller Compiler User Guide*, Altera Corporation, April 2005.
- [52] Altera Corporation, *Cyclone II Device Handbook*, vol. 1, chapter 9 - External Memory Interfaces, pp. 9-1 to 9-24, Altera Corporation, July 2005.
- [53] Altera Corporation, *altPLL Megafunction User Guide*, Altera Corporation, May 2006.
- [54] ECS, "Series SMD Clock Oscillators," Tech. Rep. ECS-3953M, ECS.
- [55] Rohde & Schwarz, *Signal Generator R&S[®] SML*, Rohde & Schwarz.
- [56] Rohde & Schwarz, *Vector Signal Generator R&S SMIQ*, Rohde & Schwarz, July 2002.
- [57] Tektronix, "Series Digital Phosphor Oscilloscopes," Tech. Rep. TDS3000B, Tektronix.
- [58] Rohde & Schwarz, *Spectrum Analyzers FSEx*, Rohde & Schwarz.
- [59] Altera Corporation, *Cyclone II Device Handbook*, vol. 1, chapter 10 - Selectable I/O Standards in Cyclone II Devices, pp. 10-1 to 10-34, Altera Corporation, November 2005.
- [60] Altera Corporation, *Cyclone II Device Handbook*, vol. 1, chapter 5 - DC Characteristics & Timing Specifications, pp. 5-23 and 5-24, Altera Corporation, March 2006.
- [61] W.H. Circuit, "Manufacturing Dimensions," Tech. Rep., W.H. Circuit, 6 May 2004.

- [62] Analog Devices, “Dual High Output Current, High Speed Amplifier,” Tech. Rep. AD8017, Analog Devices, 2002.
- [63] National Semiconductor, “400mA and 500mA Voltage Regulators,” Tech. Rep. LM2937-3.3, National Semiconductor, 2005.
- [64] Mini-Circuits, “Low Pass Filters,” Tech. Rep. SLP-50, Mini-Circuits, 3 March 2004.
- [65] Altera Corporation, *USB-Blaster Download Cable User Guide*, Altera Corporation, July 2006.
- [66] Altera Corporation, “Cyclone II DSP Board,” Tech. Rep. 150-0310202-B1, Altera Corporation, 24 March 2005.
- [67] Altera Corporation, *DSP Development Kit, Cyclone II Edition Getting Started User Guide*, Altera Corporation, May 2005.
- [68] Altera Corporation, *Quartus[®] II Installation & Licensing for PCs*, Altera Corporation, May 2006.
- [69] Altera Corporation, *Configuration Handbook*, vol. 2, chapter 4 - Serial Configuration Devices Features, pp. 4-1 to 4-40, Altera Corporation, August 2005.
- [70] Texas Instruments, “12-Bit, 125Msps Analog-To-Digital Converter,” Tech. Rep. ADS5520, Texas Instruments, 2006.
- [71] Texas Instruments, “14-Bit, 165Msps Digital-to-Analog Converter,” Tech. Rep. DAC904, Texas Instruments, May 2002.
- [72] Cypress Semiconductor, “9-Mbit Pipelined SRAM,” Tech. Rep. CY7C1360B, Cypress Semiconductor, 9 April 2004.
- [73] Silicon Laboratories, “Full Speed USB, 16k ISP Flash MCU Family,” Tech. Rep. C8051F321, Silicon Laboratories, December 2003.
- [74] Texas Instruments, “High-Speed Differential Line Driver,” Tech. Rep. SN75LVDS387, Texas Instruments, May 2001.

Appendix A

FPGA Pin Assignments

The FPGA pin assignments are detailed in this appendix. This includes the pin locations [17], I/O standards [59] and load capacitances [60].

A.1 User Interface

The Push-Buttons and DIP-Switch pins have an I/O standard of 1.8V and the clock, LED and LCD pins have a I/O standard of LVTTTL. The LED and LCD pins have an output load of 0pF.



TABLE A.1
TIMING CONTROL PIN ASSIGNMENTS

Device	Location	Device	Location
Top Clock	N2	Bottom Clock	N25

TABLE A.2
PUSH-BUTTONS PIN ASSIGNMENTS

Device	Location	Device	Location
Push-Button ₀	AC18	Push-Button ₁	AE16
Push-Button ₂	AE22	Push-Button ₃	AE14

TABLE A.3
DIP-SWITCH PIN ASSIGNMENTS

Device	Location	Device	Location
DIP-Switch ₀	AC13	DIP-Switch ₁	A19
DIP-Switch ₂	C21	DIP-Switch ₃	C23
DIP-Switch ₄	AF4	DIP-Switch ₅	AC20
DIP-Switch ₆	AE18	DIP-Switch ₇	AE19

TABLE A.4
LED PIN ASSIGNMENTS

Device	Location	Device	Location
LED ₀	E5	LED ₁	B3
LED ₂	F20	LED ₃	E22
LED ₄	AC3	LED ₅	AB4
LED ₆	AA6	LED ₇	AA7

TABLE A.5
LCD PIN ASSIGNMENTS

Device	Location	Device	Location
U32 _A	Y21	U33 _A	K2
U32 _B	T7	U33 _B	U25
U32 _C	AB23	U33 _C	AA3
U32 _D	Y50	U33 _D	V1
U32 _E	E1	U33 _E	V7
U32 _F	U1	U33 _F	U23
U32 _G	W21	U33 _G	AC2
U32 _{DP}	V3	U33 _{DP}	P7

A.2 VGA Interface

The pins have an I/O standard of LVTTTL and an output load of 0pF.

TABLE A.6
VGA CONTROL PIN ASSIGNMENTS

Device	Location	Device	Location
Blank	U6	Sync	AE2
Horizontal Sync	H21	Vertical Sync	F21
Clock	T4		

TABLE A.7
VGA COLOUR PIN ASSIGNMENTS

Device	Location	Device	Location	Device	Location
Red ₀	T20	Green ₀	R7	Blue ₀	W2
Red ₁	AC23	Green ₁	U5	Blue ₁	H2
Red ₂	U21	Green ₂	R6	Blue ₂	W1
Red ₃	P4	Green ₃	AA4	Blue ₃	U4
Red ₄	Y25	Green ₄	T6	Blue ₄	U2
		Green ₅	V4		

A.3 LVDS Interface

The pins have an I/O standard of LVTTTL, the output pins have an output load of 10pF and the enable pins have a current strength of 4mA.

TABLE A.8
LVDS DEVICE CONTROL PIN ASSIGNMENTS

Device	Location	Device	Location
Enable ₅	L2	Enable ₄	G1
Enable ₃	L6	Enable ₁	L4
Enable ₀	J6		

TABLE A.9
LVDS DRIVER PIN ASSIGNMENTS

Device	Location	Device	Location
I ₁	K5	Q ₁	L3
I ₂	K8	Q ₂	H4
I ₃	K4	Q ₃	K6

TABLE A.10
LVDS RECEIVER PIN ASSIGNMENTS

Device	Location	Device	Location	Device	Location
Data ₀	J8	Data ₁	F4	D _{VALID}	G5
Data ₂	F1	Data ₃	J1	P _{SYNC}	V2
Data ₄	J3	Data ₅	J2	Clock	E2
Data ₆	M3	Data ₇	M2		

A.4 External RAM Interface

The pins have an I/O standard of SSTL-18 CLASS I, the address and control pins have an output load of 24pF, the clock pins have an output load of 10pF and the mask, strobe and data pins have an output load of 4pF.

TABLE A.11
EXTERNAL RAM CLOCK PIN ASSIGNMENTS

Device	Location	Device	Location
Clock ₀ ⁺	AC21	Clock ₀ ⁻	AD19
Clock ₁ ⁺	AB20	Clock ₁ ⁻	AD21
Clock ₂ ⁺	AD22	Clock ₂ ⁻	AA20

TABLE A.12
EXTERNAL RAM ADDRESS PIN ASSIGNMENTS

Device	Location	Device	Location
Bank Address ₀	Y18	Bank Address ₁	AF23
Address ₀	AE4	Address ₁	AC8
Address ₂	AD6	Address ₃	Y10
Address ₄	AF5	Address ₅	AD7
Address ₆	AC6	Address ₇	AB8
Address ₈	AD5	Address ₉	AE11
Address ₁₀	AE5	Address ₁₁	AD4
Address ₁₂	Y12		

TABLE A.13
EXTERNAL RAM CONTROL PIN ASSIGNMENTS

Device	Location	Device	Location
Column Address Strobe	AC22	Write Enable	AA17
Row Address Strobe	AE20	Clock Enable ₀	AE21
On-Die Termination ₀	AE23	Chip Select ₀	AF22

TABLE A.14
EXTERNAL RAM DATA MASK PIN ASSIGNMENTS

Device	Location	Device	Location	Device	Location
Data Mask ₀	AC15	Data Mask ₁	AA12	Data Mask ₂	AC9

TABLE A.15
EXTERNAL RAM DATA STROBE PIN ASSIGNMENTS

Device	Location	Device	Location	Device	Location
Data Strobe ₀	AF19	Data Strobe ₁	AE15	Data Strobe ₂	AE13

TABLE A.16
EXTERNAL RAM DATA BUS PIN ASSIGNMENTS

Device	Location	Device	Location
Data Bus ₀	AD16	Data Bus ₁	AF17
Data Bus ₂	AE17	Data Bus ₃	AC17
Data Bus ₄	AD17	Data Bus ₅	AA16
Data Bus ₆	Y16	Data Bus ₇	AF18
Data Bus ₈	AD12	Data Bus ₉	AE12
Data Bus ₁₀	AC14	Data Bus ₁₁	AA13
Data Bus ₁₂	Y13	Data Bus ₁₃	Y14
Data Bus ₁₄	Y15	Data Bus ₁₅	AA15
Data Bus ₁₆	AE9	Data Bus ₁₇	AF94
Data Bus ₁₈	AD10	Data Bus ₁₉	AC11
Data Bus ₂₀	AE10	Data Bus ₂₁	AF10
Data Bus ₂₂	AB12	Data Bus ₂₃	AD11

Appendix B

Additional Boards

The additional Printed Circuit Boards (PCB) designed for this system are detailed in this appendix. This includes the schematics, layouts, bill of materials and modifications.

B.1 Modifications

The following modifications were made to rectify the additional PCB.

B.1.1 Power Supply Modifications

The FPGA logics pins should have sufficient power capabilities to supply the LVDS devices [59], but this was not found to be the case and the power routing on both of the additional PCB connected to the FPGA had to be modified.

TTL-to-LVDS PCB Modifications

A wire is soldered on between pin₁ of capacitor C₁ and pin₁ of header JP₂ to connect to the 3.3V supply at pin₂ of J₂₂ of the FPGA DevBoard. The track connected between pin₁ of capacitor C₁ and pin₇ of header JP₂ is broken.

LVDS-to-TTL PCB Modifications

A wire is soldered on pin₁ of capacitor C₉ to connect to an external power supply delivering 3.3V. The track between pin₁ of capacitor C₅ and pin₂ of JP₄ is broken. The track between pin₁ of capacitor C₇ and pin₁₃ of JP₄ is broken. The track between pin₁ of capacitor C₉ and pin₃₀ of JP₄ is broken.

B.1.2 Data Signal Modifications

The usage of incorrect footprints for connector JP₅, unit U₃ and unit U₄ resulted in the pin switching of the DVB-SPI depicted in Figure B.1.2.

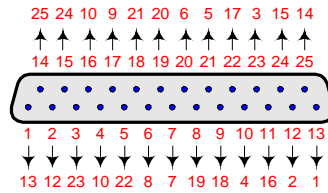


Fig. B.1. Corrected DVB-SPI Pin Assignments

This was corrected by inserting an adapter at connector JP₅ that reroutes the incoming data signals. This adapter is made up of a male 25-pin D-Type connector, a female 25-pin D-Type connector and interconnecting wires.

B.2 Schematics

The PCB that converts the DVB-S Receiver’s DVB-SPI signals to TTL signals for use in the FPGA is depicted in Figure B.2. This PCB is inserted in connector J₁₅ of the FPGA DevBoard.

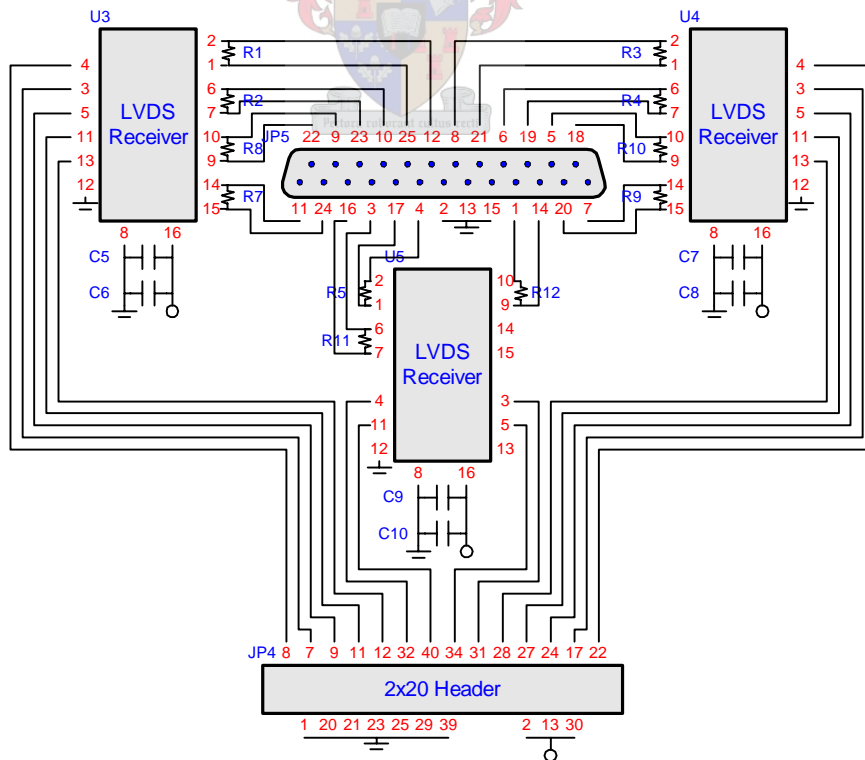


Fig. B.2. LVDS-to-TTL Board Schematic

The PCB that converts the FPGA's TTL signals to LVDS signals for transfer to Malan's board is depicted in Figure B.2. This PCB is inserted in connector J₂₂ of the DevBoard.

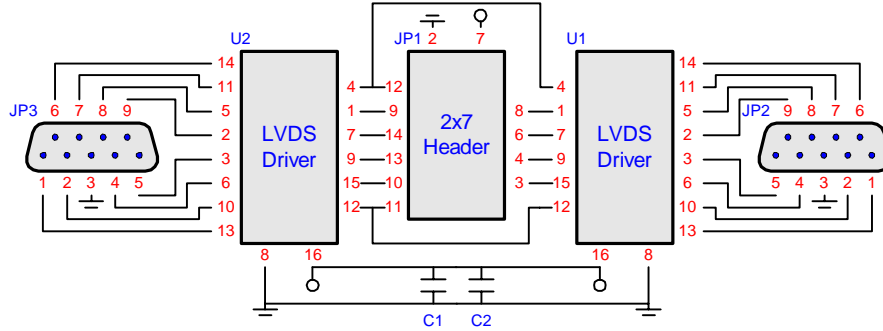


Fig. B.3. TTL-to-LVDS Board Schematic

The PCB that convert the LVDS signals to TTL signals for use on Malan's board are depicted in Figure B.2 and Figure B.2. These PCB are inserted in connectors U₁₆ to U₂₃ of Malan's board.

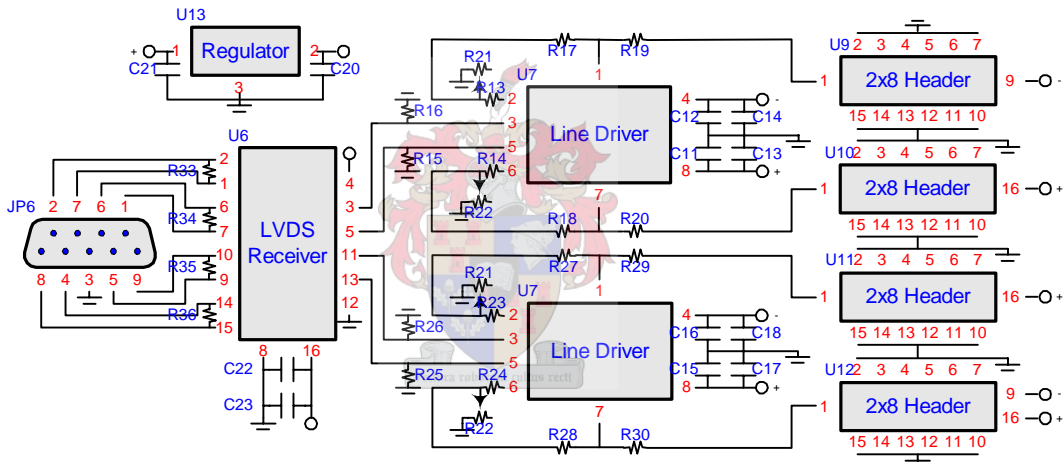


Fig. B.4. Add-On Board Schematic

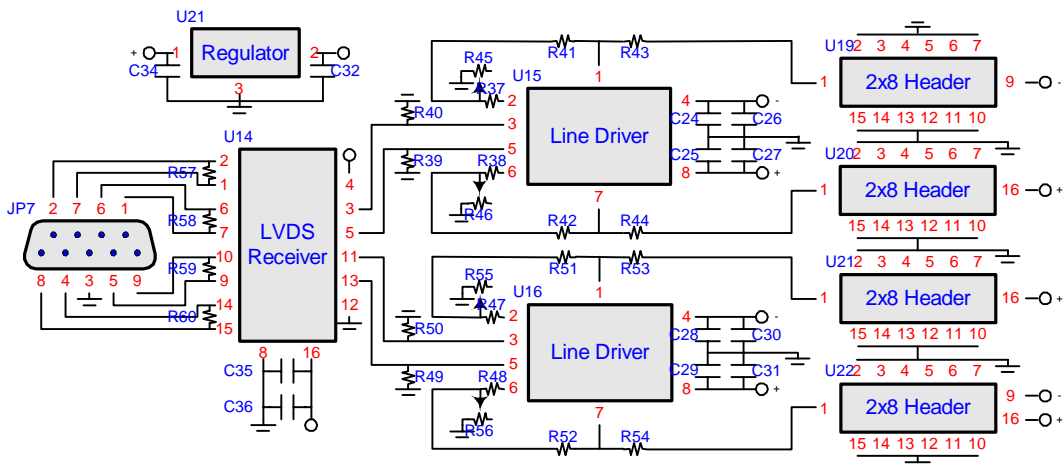


Fig. B.5. Duplicate Add-On Board Schematic

B.3 Bill of Materials

The substrate is a 1.6mm thick double-sided RF4 board with $35\mu\text{m}$ copper cladding manufactured to IPC standards Class 2 by W.H. Circuit [61].

TABLE B.1
DEVICES

Board Reference	Part Number	Description
U ₁ -U ₂	SN65LVDS31D [25]	LVDS Driver
U ₃ -U ₆ ,U ₁₄	SN65LVDS32D [26]	LVDS Receiver
U ₇ -U ₈ ,U ₁₅ -U ₁₆	AD8017AR [62]	Line Driver
U ₉ -U ₁₂ ,U ₁₇ -U ₂₀	2x8 Socket	Voltage Divider
U ₁₃ ,U ₂₁	LM2937-3.3 [63]	3.3V Voltage Regulator

TABLE B.2
CONNECTORS

Board Reference	Part Number	Description
JP ₁	2x7 Header	DevBoard Connector
JP ₂ -JP ₃ ,JP ₆ -JP ₇	9-Pin D-Type Connector	LVDS Connectors
JP ₄	2x20 Header	DevBoard Connector
JP ₅	25-Pin D-Type Connector	DVB-SPI Connector
U ₉ -U ₁₂ ,U ₁₇ -U ₂₀	2x8 Socket	Voltage Divider Connector
U ₉ -U ₁₂ ,U ₁₇ -U ₂₀	1x8 Socket	Malan's Board Connector

TABLE B.3
CAPACITORS

Board Reference	Part Number	Description
C ₁₁ -C ₁₂ ,C ₁₅ -C ₁₆ ,C ₂₄ -C ₂₅ ,C ₂₈ -C ₂₉	SMD Capacitor	1 μF Capacitor
C ₂₀ ,C ₃₂	SMD Capacitor	10 μF Capacitor
C ₁ ,C ₅ ,C ₇ ,C ₉ ,C ₁₃ -C ₁₄ ,C ₁₇ -C ₁₈	SMD Capacitor	100nF Capacitor
C ₂₁ ,C ₂₃ ,C ₂₆ -C ₂₇ ,C ₃₀ -C ₃₁ ,C ₃₄ ,C ₃₆	SMD Capacitor	1nF Capacitor
C ₂ ,C ₆ ,C ₈ ,C ₁₀ ,C ₂₂ ,C ₃₅	SMD Capacitor	1nF Capacitor

TABLE B.4
RESISTORS

Board Reference	Part Number	Description
R ₁ -R ₅ ,R ₇ -R ₁₂ ,R ₁₉ -R ₂₀ ,R ₂₉ -R ₃₀	SMD Resistor	100 Ω Resistor
R ₃₃ -R ₃₆ ,R ₄₃ -R ₄₄ ,R ₅₃ -R ₅₄ ,R ₅₇ -R ₆₀	SMD Resistor	1k Ω Resistor
R ₁₅ -R ₁₆ ,R ₂₅ -R ₂₆ ,R ₃₉ -R ₄₀ ,R ₄₉ -R ₅₀	SMD Resistor	560 Ω Resistor
R ₁₃ -R ₁₄ ,R ₁₇ -R ₁₈ ,R ₂₃ -R ₂₄ ,R ₂₇ -R ₂₈	SMD Resistor	20k Ω Potentiometer
R ₃₇ -R ₃₈ ,R ₄₁ -R ₄₂ ,R ₄₇ -R ₄₈ ,R ₅₁ -R ₅₂	Potentiometer	20k Ω Potentiometer
R ₂₁ -R ₂₂ ,R ₃₁ -R ₃₂ ,R ₄₅ -R ₄₆ ,R ₅₅ -R ₅₆	Potentiometer	20k Ω Potentiometer

B.4 Board Layouts

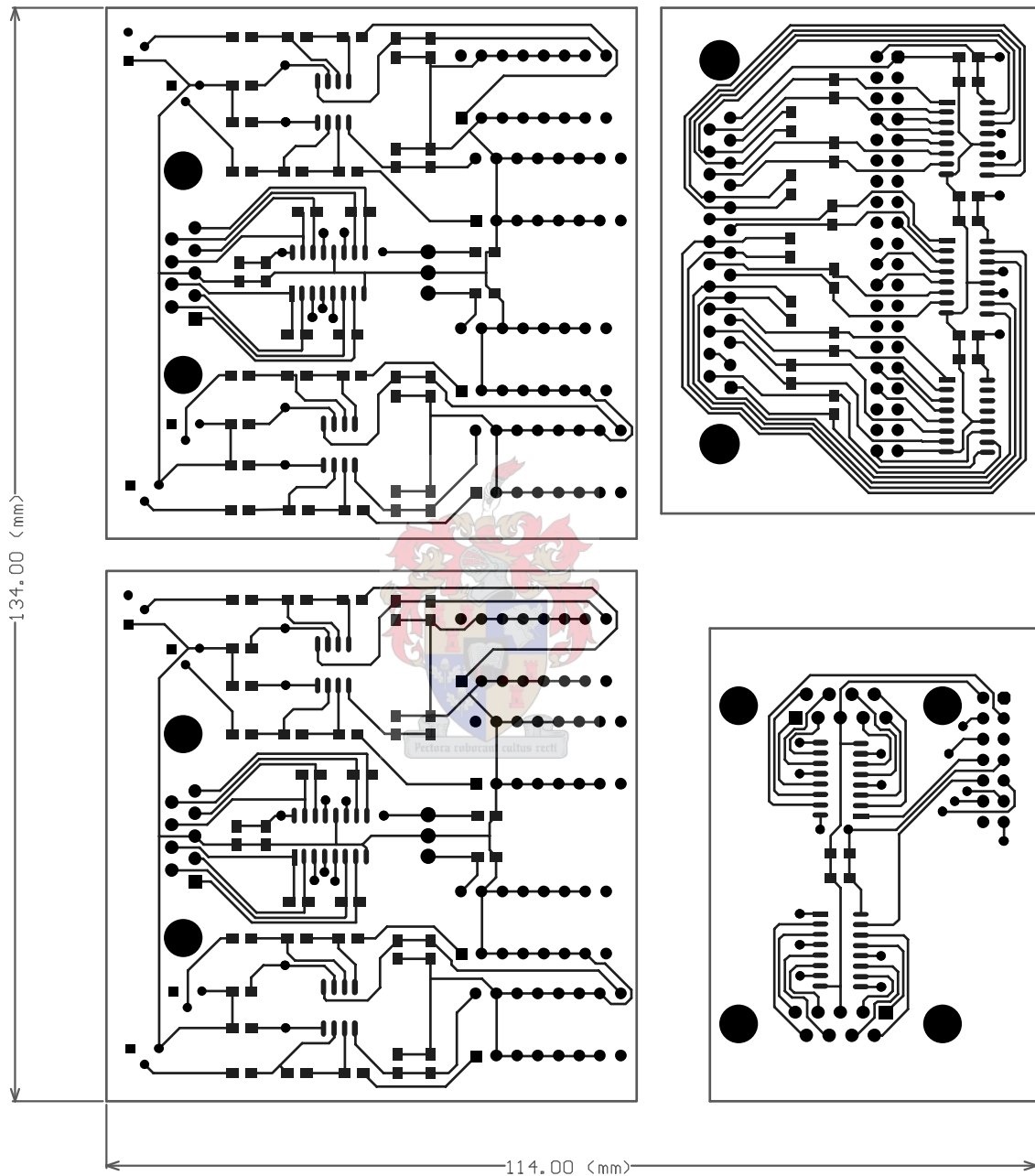


Fig. B.6. Top Layer

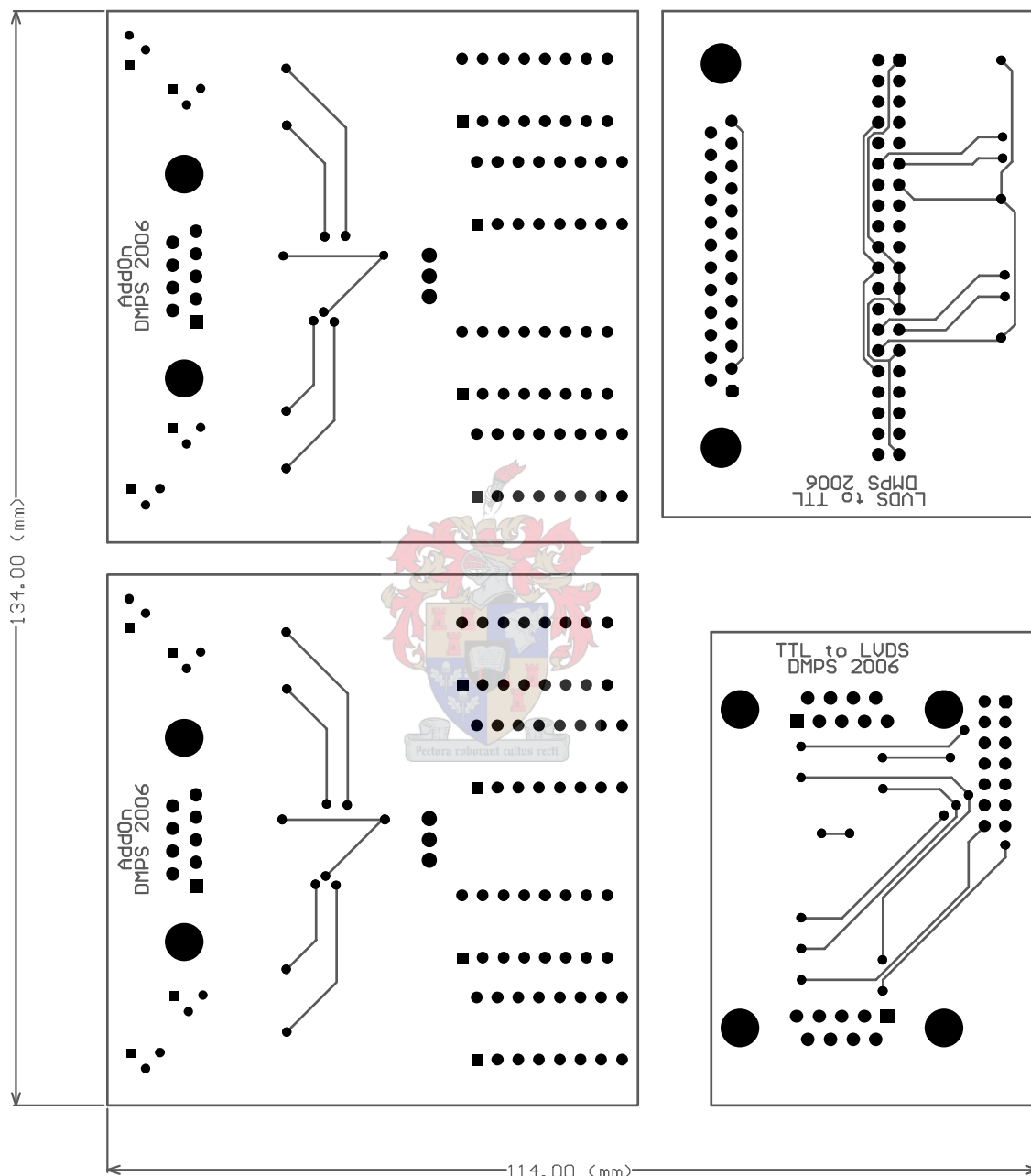


Fig. B.7. Bottom Layer

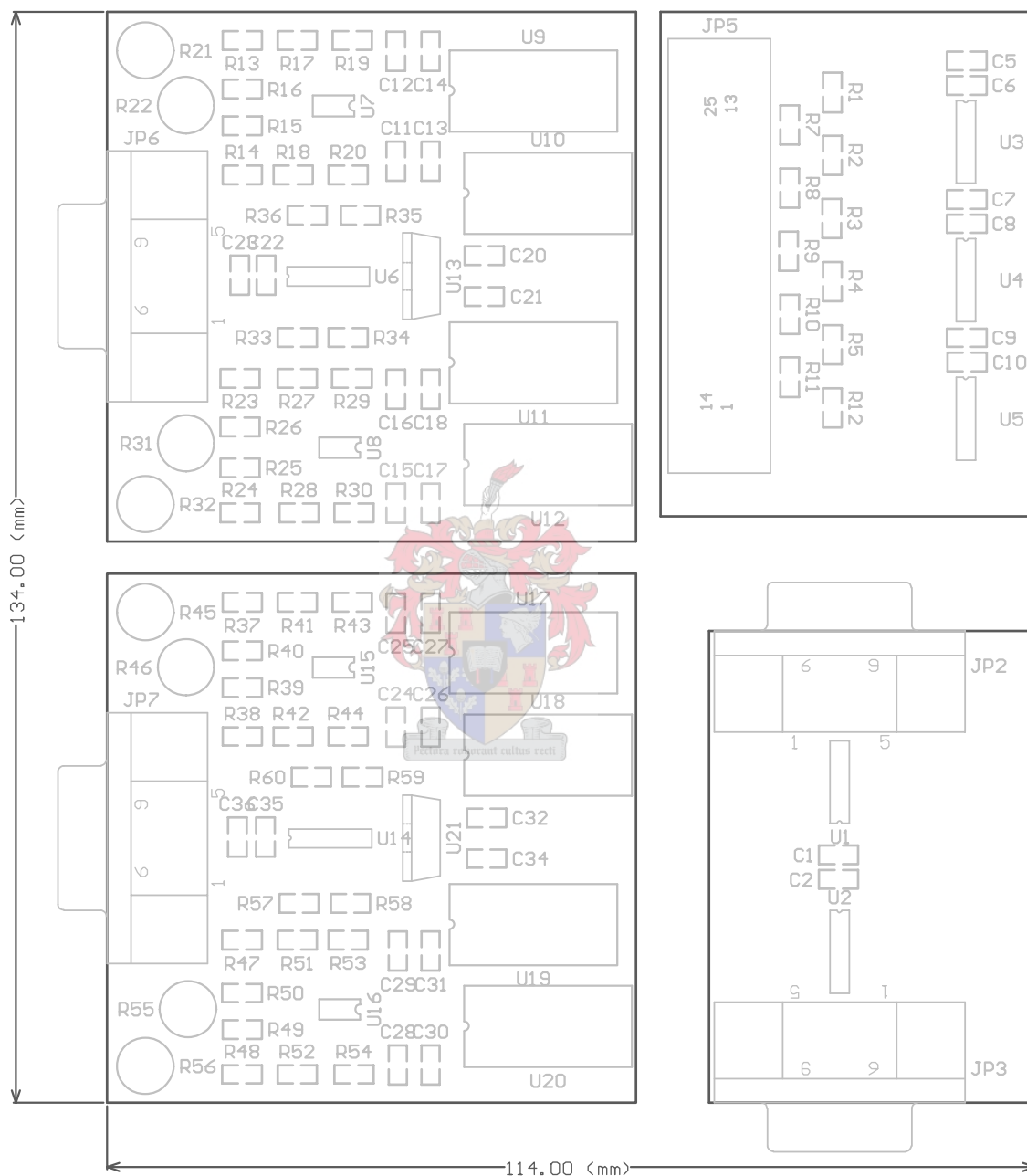


Fig. B.8. Top Silkscreen Overlay

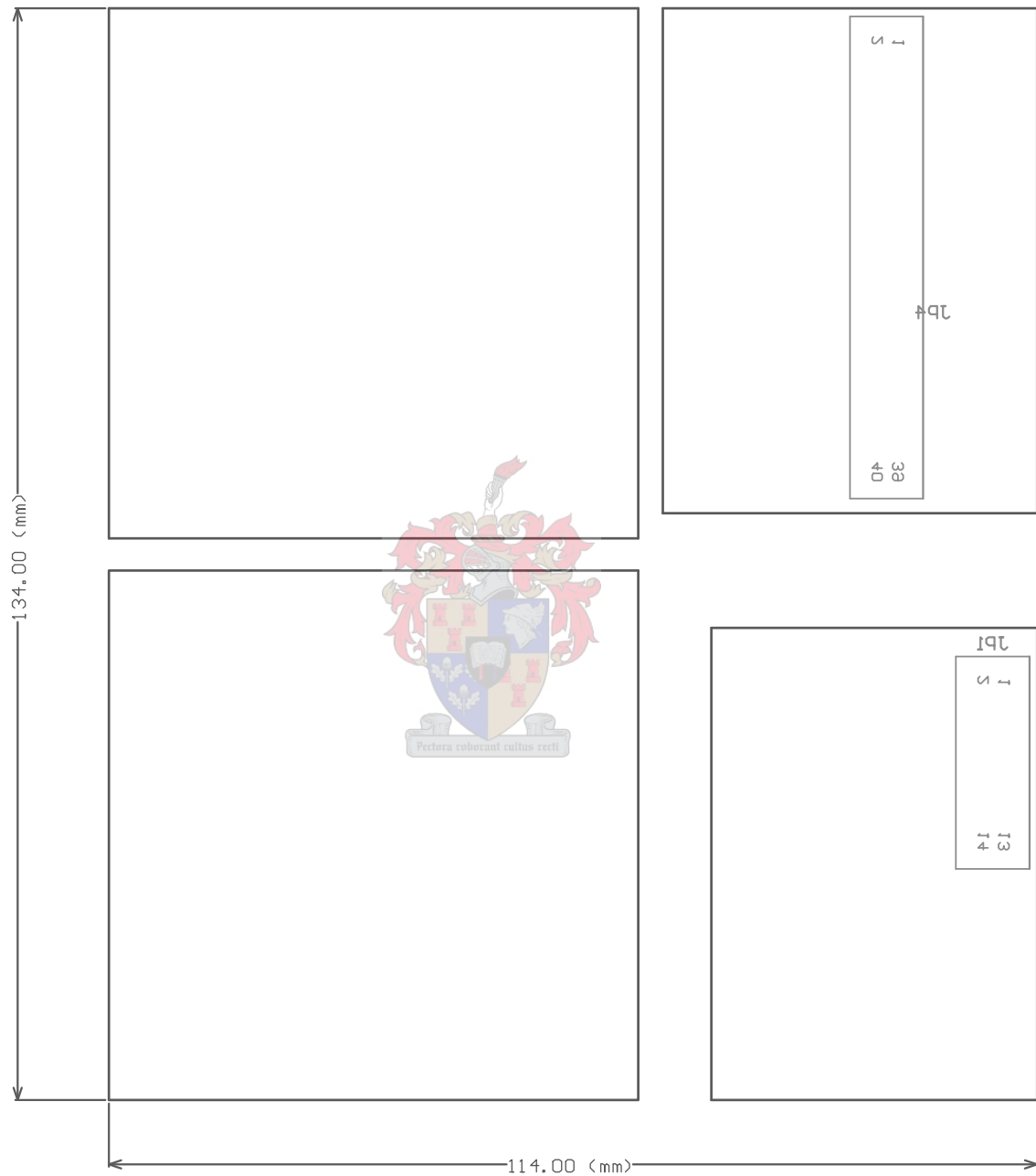


Fig. B.9. Bottom Silkscreen Overlay

Appendix C

FPGA Development Kit

The FPGA Development Kit is detailed in this appendix. This includes the kit's hardware and software contents, the board's components and modifications made to the board.

C.1 FPGA Development Kit Contents

- Hardware Contents
 - Cyclone[®] II EP2C35 DSP DevBoard [17]
 - Anti-Aliasing Filter (Mini-Circuits SLP-50) [64]
 - SMA Cable
 - USB-Blaster[™] Download Cable (Altera PL-USB-BLASTER-RB) [65]
 - USB Cable
 - Power Supply Cables (UK, US and Europe)
 - Power Supply Adapter (GlobTek TR9KT3750LCP-Y)
- Software Contents
 - DSP Development Kit, Cyclone II Edition v1.0.0
 - DSP Builder v5.0.0
 - MATLAB v7.0.4
 - Simulink v6.2
 - Quartus[®] II DKE v5.0
 - Nios[®] II Embedded Processor Evaluation Edition v5.0
 - MegaCore[®] IP Library v5.0

- Documentation
 - Cyclone II EP2C35 DSP Development Board Reference Manual [17]
 - Cyclone II EP2C35 DSP Development Board Schematic [66]
 - DSP Development Kit, Cyclone II Edition Getting Started User Guide [67]
 - Quartus[®] II Installation & Licensing for PCs [68]

C.2 FPGA DevBoard Modifications

The choice of voltage level at pin₂ of header J₂₂ is determined by the presence of a resistor at either R₂₁, for 5V, or R₂₀, for 3.3V. As the FPGA DevBoard is produced with the resistor at R₂₁ and as the LVDS devices require a 3.3V supply, this resistor is moved to R₂₀.

C.3 FPGA DevBoard Components

The FPGA DevBoard, depicted in Figure C.3, contains the component listed below:

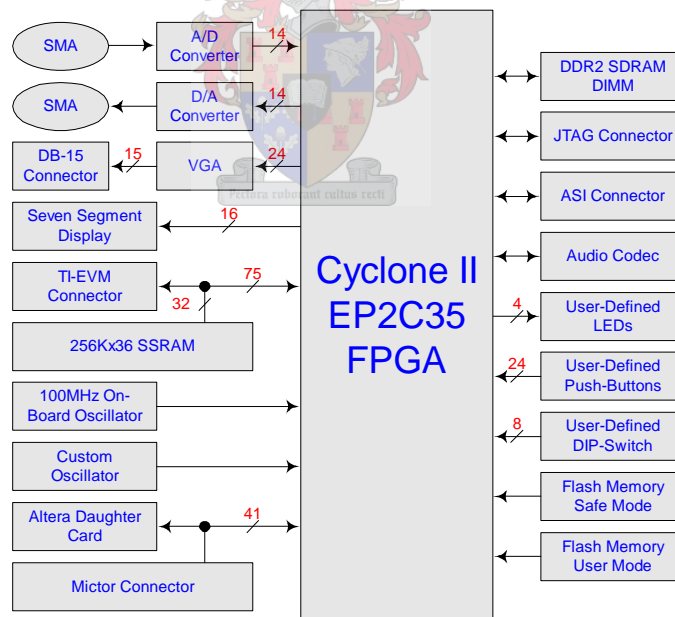
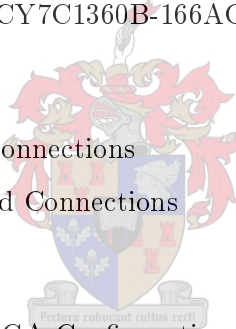


Fig. C.1. FPGA DevBoard Functional Diagram

- Programmable Devices
 - Cyclone[™] II FPGA (Altera EP2C35F672-C6) [12]
 - 8MB Flash (Altera EPCS64SI16N) [69]

- Analogue I/O
 - 12-bit, 70dB SNR, 125Msps ADC (Texas ADS5520) [70]
 - 14-bit, 70dB SNR, 165Msps DAC (Texas DAC904E) [71]
 - 24-bit, 180Msps, RGB VGA DAC (Fairchild FMS3818) [19]
 - 8-96kHz Stereo Audio Codec (Texas TLV320AIC23) [20]
- User I/O
 - Eight User-Defined LEDs
 - Two User-Defined Seven-Segment LCD Displays
 - Octal User-Defined DIP-Switch
 - Four User-Defined Push-Buttons
- Memory Components
 - 256MB DDR2 SDRAM DIMM (Micron MT8HTF3264AY-40E) [18]
 - 1MB SSRAM (Cypress CY7C1360B-166AC) [72]
- Expansion Connectors
 - Altera Daughter Card Connections
 - TMS320C6416 DevBoard Connections
- General Connectors
 - JTAG Connector for FPGA Configuration
 - Mictor Connector for Hardware and Software Debugging
 - ASI Connector for EPCS64 Programming
 - Line-In Audio Jack
 - Line-Out Audio Jack
 - Headphone Jack
- Indicators
 - Power Indicator LED
 - Configuration Indication LED
- Switches
 - Power Switch
 - DevBoard Reset Push-Button
 - Hardware Reset Push-Button



Appendix D

DVB-S Receiver Evaluation Kit

The DVB-S Receiver Evaluation Kit is detailed in this appendix. This includes the kit's contents and the board's components.

D.1 DVB-S Receiver Evaluation Kit Contents

The DVB-S Receiver Evaluation Kit contains the following items:

- Hardware Contents
 - Si2110USB-EVB EvBoard
 - USB Cable
- Software Contents
 - Si2110 EVB Device Driver Set
 - Si2110 EVB v0.81
- Documentation
 - Si2110 - Satellite Receiver for DVB-S with *QuickLock* and *QuickScan* [14]
 - Si2110USB-EVB - EvBoard for Si2110 SiRX™ DVB Receiver IC [21]



D.2 DVB-S Receiver EvBoard Components

The DVB-S Receiver EvBoard, depicted in Figure D.2, contains the component listed below:

- DVB-S Satellite Receiver (Silicon Si2110-D-FM)

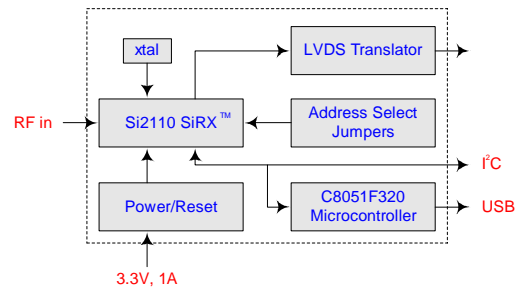


Fig. D.1. DVB-S EvBoard Functional Diagram

- User Interface
 - RF Input Connector
 - Microcontroller (Silicon C8051F321) [73]
 - LVDS Line Driver (Texas SN75LVDS387DGG) [74]
- General Connectors
 - DVB-SPI Connector
 - MPEG-2 TS Header
 - I²C Connector
 - USB Type-B Connector
 - Balun
- Indicators
 - DVB-S Signal Lock Indicator
 - Reset Indicator
 - I²C Write Activity Indicator
 - I²C Read Activity Indicator
 - USB Interface power Indicator
 - 3.3V Power Indicator
 - LNB Power Indicator
- Switches
 - Reset Button

