



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Research Commons

<http://waikato.researchgateway.ac.nz/>

Research Commons at the University of Waikato

Copyright Statement:

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

The thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of the thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from the thesis.

The Design of a CD Transport for Audio Applications

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Masters of Engineering in Physics
and Electronic Engineering
at the
University of Waikato

By

Carl Dennis Benton



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

2006

ABSTRACT

The project to design a CD transport (CD player) in conjunction with Perreux Industries came about from the need for a source component in their Silhouette series of products. This project describes the design a high quality CD player, at a low price, to compliment Perreux's Silhouette series.

A CD drive is selected over a proprietary optical pickup due to the former's low cost and the standardisation of the interface. The control circuitry includes a micro controller and discrete logic to provide the correct data and clock signals to the SPDIF transmitter and DAC circuits. These two circuits provided a high quality analogue output, and facilitate an upgrade path by connecting the SPDIF output to an external DAC.

After three board iterations, a final production ready revision was achieved. The design includes a high quality toroidal transformer, low jitter crystal oscillator, and a very high quality SPDIF pulse transformer output. The design also allows a remote input to control the player, and an optional digital cable via an RJ45 connector to provide synchronisation with a future design of the SXD2 DAC module, or to transmit SPDIF to a remote location.

The specifications of the final design were higher than expectations. The digital output boasts equal or superior performance to competitive products in the same price range, with the analogue output attaining exceptionally high performance.

ACKNOWLEDGEMENTS

There are several parties whom I wish to thank for the opportunity to work on such a quality and educational project. Firstly, my supervisor Dale Carnegie, who has been a mentor to me at the University of Waikato for several years throughout the course of my degree, and who set up both my second work placement and this masters project with Perreux Industries Ltd. Also despite Dale's relocation to Wellington, he remained my supervisor and worked hard to give me a great deal of attention. I hope Dale and his family enjoy their new life in the capital.

I would also like to thank Martin Van Rooyen, the Managing Director at Perreux Industries Ltd, for the opportunity to work on the project. Also Clinton Jenson (Design Engineer) for working with me during my time at Perreux. I wish both of them the best for the future.

I would also like to thank my partner (fiancé at the time of writing this report) Christy Brett for her support and understanding during the time I was working in Dunedin with Perreux. For the duration of the project Christy remained in Hamilton also working on a master's project, and despite having over 1200 km between us for 8 months, we finished on a high note when we got engaged. I would also like to thank Christy for helping proof read the thesis, and for simply being there. Christy, I love you.

DEDICATION

I would like to dedicate this thesis to both my parents for the excellent job they have done over the past 23 years. My Mother is the most understanding woman I know and has always shown an interest in my passion, electronics. My Father introduced me to kit-set construction, which was most likely to stop me from dismantling my electronic toys. However, it is what started the ball rolling, and my Father has always encouraged me to use my full potential.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS	vii
LIST OF TABLES	xi
LIST OF FIGURES	xiii
1 INTRODUCTION	1
1.1 SCOPE.....	1
1.2 COMPANY OVERVIEW	1
1.3 THE SILHOUETTE SERIES	2
1.4 PROJECT REQUIREMENTS	3
1.5 THESIS STRUCTURE	4
2 BACKGROUND	7
2.1 COMPETITIVE PRODUCTS	7
2.1.1 <i>Cyrus Audio CD 6s</i>	7
2.1.2 <i>Marantz CD5400</i>	8
2.1.3 <i>Cambridge Audio Azur 640C</i>	8
2.1.4 <i>NAD C521BEE</i>	9
2.1.5 <i>Arcam Diva CD73</i>	9
2.1.6 <i>Denon DCD-685</i>	10
2.1.7 <i>Rotel RCD-1072</i>	10
2.1.8 <i>Summary</i>	11
2.2 A NOTE ON HI-FI.....	12
2.3 DESIGN CONSTRAINTS.....	12
2.3.1 <i>Costs</i>	12
2.3.2 <i>Component sourcing</i>	13
2.3.3 <i>System Clock</i>	13
2.4 THE OPTICAL MECHANISM.....	13
2.5 OTHER BACKGROUND INFORMATION	14
2.5.1 <i>The Compact Disc</i>	14
2.5.2 <i>Stream Is Accurate</i>	15

2.5.3 Jitter.....	16
2.5.4 Digital Audio Transmission.....	17
3 HARDWARE DESCRIPTION	19
3.1 OVERVIEW	19
3.2 THE ATAPI INTERFACE.....	20
3.3 THE CONTROLLER	20
3.4 DIGITAL OUTPUT	22
3.4.1 SPDIF Chip	22
3.4.2 Output Circuit.....	22
3.5 ANALOGUE OUTPUT	25
3.5.1 Digital to Analogue Converter	25
3.5.2 Output Filter.....	26
3.5.3 Analogue Power Supply	29
3.6 INTER CIRCUIT DIGITAL AUDIO TRANSMISSION	30
3.6.1 Serial Formats	30
3.6.2 Parallel to Serial Conversion.....	30
3.7 CLOCK GENERATION	32
3.8 MASTER CLOCK.....	34
3.9 USER INTERFACE	35
3.10 POWER SUPPLY	37
3.10.1 Circuit Overview	37
3.10.2 Transformer Rating	38
3.10.3 Voltage Regulators	39
3.10.4 Filtering.....	40
3.11 REMOTE INPUT	41
3.12 RJ45 CONNECTOR	42
3.13 PCB LAYOUT	43
3.14 INTERCONNECTS	45
3.15 DIMENSIONS	45
3.16 FRONT PANEL	46
4 FIRMWARE DESCRIPTION	47
4.1 OVERVIEW	47
4.1.1 Restrictions	48

4.1.2 <i>Interrupt Structure</i>	48
4.1.3 <i>Nested Interrupts</i>	49
4.1.4 <i>Code Definitions</i>	50
4.2 ATAPI INTERFACE	50
4.2.1 <i>Register Access</i>	51
4.2.2 <i>Packet Commands</i>	53
4.3 SYSTEM CONFIGURATION	55
4.4 KEY PROCESSING	56
4.4.1 <i>Play Key</i>	57
4.4.2 <i>Stop Key</i>	58
4.4.3 <i>Skip Keys</i>	58
4.4.4 <i>Scan Keys</i>	60
4.4.5 <i>Eject Key</i>	62
4.5 READING THE TABLE OF CONTENTS	62
4.6 AUDIO PLAYBACK	64
4.6.1 <i>Audio Buffer</i>	65
4.6.2 <i>Data Setup</i>	67
4.6.3 <i>Timing Restrictions</i>	68
4.6.4 <i>Actual Timing</i>	70
4.6.5 <i>Current Audio Position</i>	71
4.6.6 <i>Emphasis Control</i>	72
4.7 USER INTERFACE	73
4.8 REMOTE INPUT	75
4.9 THE LED	77
5 FINAL DESIGN PERFORMANCE	79
5.1 PHYSICAL CONSTRUCTION	79
5.2 SXCD OPERATION	82
5.2.1 <i>CD formats</i>	83
5.3 METHOD OF MEASUREMENT	85
5.4 ANALOGUE OUTPUT	88
5.4.1 <i>Frequency Response</i>	88
5.4.2 <i>Total Harmonic Distortion</i>	89
5.4.3 <i>Signal to Noise Ratio</i>	90
5.4.4 <i>Master Clock Jitter</i>	91

5.5 DIGITAL OUTPUT	93
5.5.1 Total Harmonic Distortion	93
5.5.2 Signal to Noise Ratio	94
5.5.3 Digital Signal Quality.....	95
5.5.4 Digital Signal Jitter	96
6 CONCLUSION	101
6.1 REVIEW	101
6.2 ANALYSIS OF RESULTS	101
6.2.1 Signal to Noise Ratio	102
6.2.2 Total Harmonic Distortion	102
6.2.3 Other Digital Parameters.....	103
6.3 CIRCUIT OPERATION	104
6.4 FUTURE IMPROVEMENTS.....	105
6.5 SUMMARY.....	106
7 REFERENCES	109
8 ACRONYMS.....	113
9 APPENDICES.....	117
9.1 APPENDIX A - PHILIPS L1210 QUOTATION:	117
9.2 APPENDIX B - SCHEMATICS	119
9.2.1 Main Schematic	119
9.2.2 Power Supply.....	120
9.2.3 Outputs	121
9.3 APPENDIX C – PCB LAYOUT	123
9.4 APPENDIX D – LETTER OF RECOMMENDATION	125
9.5 APPENDIX E – PROJECT CD CONTENTS	127

LIST OF TABLES

TABLE 3.1: SUMMARY OF CONTROLLER PIN CONNECTIONS	21
TABLE 3.2: SUMMARY OF REQUIRED CLOCK SIGNALS.....	32
TABLE 3.3: CD DRIVE CURRENT REQUIREMENTS.....	38
TABLE 4.1: LIST OF INTERRUPTS	49
TABLE 4.2: COMMAND BLOCK REGISTER LIST	51
TABLE 4.3: PACKET COMMANDS USED BY THE CONTROLLER	54
TABLE 4.4: LEAD IN AREA (TOC), SUB CHANNEL Q FORMATS.....	63
TABLE 4.5: LEAD IN AREA (TOC) POINT VALUE DESCRIPTIONS	63
TABLE 4.6: LED STATUS SUMMARY	77
TABLE 5.1: TOC FOR EXAMPLE MIXED MODE CD.....	84
TABLE 5.2: <i>TRACKINFO</i> ARRAY CONTENTS.....	84
TABLE 5.3: <i>LEADOUTTRACK</i> ARRAY CONTENTS.....	84
TABLE 5.4: EXTRACT FROM COLDPLAY X&Y CD TOC	85
TABLE 6.1: COMPARISON FOR THE CD PLAYERS DISCUSSED IN SECTION 2.1	104

LIST OF FIGURES

FIGURE 1.1: PERREAUX R200I INTEGRATED AMPLIFIER.....	1
FIGURE 1.2: THE SILHOUETTE SERIES SX25I.....	2
FIGURE 2.1: CYRUS AUDIO CD6 CD PLAYER.....	7
FIGURE 2.2: MARANTZ CD5400 CD PLAYER.....	8
FIGURE 2.3: CAMBRIDGE AUDIO AZUR 640C CD PLAYER.....	9
FIGURE 2.4: NAD C521BEE CD PLAYER.....	9
FIGURE 2.5: ARCAM DIVA CD73 CD PLAYER.....	10
FIGURE 2.6: DENON DCD-685 CD PLAYER.....	10
FIGURE 2.7: ROTEL RCD-1072 CD PLAYER.....	11
FIGURE 2.8: A JITTERED CLOCK COMPARED WITH AN IDEAL CLOCK.....	16
FIGURE 2.9: THE COMPONENTS OF JITTER.....	17
FIGURE 3.1: SYSTEM BLOCK DIAGRAM.....	19
FIGURE 3.2: THE CONTROLLER.....	21
FIGURE 3.3: SPDIF OUTPUT CIRCUIT.....	23
FIGURE 3.4: THEVENIN EQUIVALENT SPDIF OUTPUT.....	23
FIGURE 3.5: SIMULATED SPDIF SIGNAL (LOADED AND UNLOADED).....	24
FIGURE 3.6: DAC MFB FILTER.....	26
FIGURE 3.7: FILTER RESPONSE TO A STEP INPUT.....	27
FIGURE 3.8: BUTTERWORTH, BESSEL AND CHEBYSHEV RESPONSE CURVES.....	27
FIGURE 3.9: SIMULATED FREQUENCY RESPONSE.....	28
FIGURE 3.10: ANALOGUE POWER SUPPLY.....	29
FIGURE 3.11: I2S FORMAT.....	30
FIGURE 3.12: SERIAL DATA GENERATION CIRCUIT.....	31
FIGURE 3.13: TIMING OF THE SHIFT REGISTER SIGNALS.....	32
FIGURE 3.14: CLOCK GENERATION CIRCUIT.....	33
FIGURE 3.15: <i>BITCLK</i> AND <i>BITCLK_INV</i> PHASE RELATIONSHIP.....	34
FIGURE 3.16: PHASE NOISE FOR A 10 MHz CFPS-73.....	35
FIGURE 3.17: IMAGE OF USER INTERFACE BUTTONS.....	36
FIGURE 3.18: USER INTERFACE CIRCUIT.....	36
FIGURE 3.19: POWER SUPPLY CIRCUIT.....	37
FIGURE 3.20: POWER SUPPLY REGULATOR HEATSINKS.....	40
FIGURE 3.21: LOCAL L-C FILTERS.....	40

FIGURE 3.22: A SIMPLE MODULATED SIGNAL.....	41
FIGURE 3.23: REMOTE INPUT CIRCUIT.....	41
FIGURE 3.24: SIMULATED REMOTE IN FOR 5 V AND 12 V SIGNALS	42
FIGURE 3.25: RJ45 CONNECTOR CIRCUIT.....	43
FIGURE 3.26: PCB CIRCUIT OVERLAY	44
FIGURE 3.27: MAIN SUB-CIRCUITS ON THE PCB	44
FIGURE 4.1: FIRMWARE OVERVIEW STATE DIAGRAM.....	47
FIGURE 4.2: REGISTER READ AND REGISTER WRITE FUNCTIONS	52
FIGURE 4.3: ROUTINES TO WAIT FOR A GIVEN STATUS BIT	52
FIGURE 4.4: READ WORD FUNCTION.....	53
FIGURE 4.5: SENDPACKETCOMMAND FUNCTION	54
FIGURE 4.6: PLAY KEY PROCESSING FLOW DIAGRAM	58
FIGURE 4.7: FF AND RW KEY FUNCTIONALITY DURING AUDIO PLAYBACK.....	59
FIGURE 4.8: SCAN FUNCTION FLOW DIAGRAM	61
FIGURE 4.9: PLAY OPERATION STATE DIAGRAM	65
FIGURE 4.10: BUFFER OVERVIEW.....	66
FIGURE 4.11: CONTROLLER MEMORY ORGANISATION	66
FIGURE 4.12: LOAD SAMPLE CODE IN ASSEMBLY LANGUAGE	69
FIGURE 4.13: SR_EN INTERRUPT WORST CASE TIMING	70
FIGURE 4.14: UPDATE EMPHASIS FUNCTION	72
FIGURE 4.15: USER INTERFACE INTERRUPT ROUTINE STATE DIAGRAM.....	73
FIGURE 4.16: PUSH KEY AND POP KEY FUNCTIONS	75
FIGURE 4.17: REMOTE INPUT INTERRUPT ROUTINE STATE DIAGRAM.	76
FIGURE 4.18: SAMPLE LOCATION FOR REMOTE INPUT	77
FIGURE 5.1: PROTOTYPE RUNNING WITH TEMPORARY INTERFACE	79
FIGURE 5.2: PCB OUTLINE AND CRITICAL DIMENSIONS	80
FIGURE 5.3: CD DRIVE INTERCONNECTS.....	81
FIGURE 5.4: COMPLETED POWER SUPPLY	81
FIGURE 5.5: SXCD MAIN OPERATION STATE DIAGRAM.....	82
FIGURE 5.6: SXCD STANDBY STATE DIAGRAM.....	82
FIGURE 5.7: THE SXCD UNDERGOING TESTS	86
FIGURE 5.8: TYPICAL ATS-2 TEST SETUP	87
FIGURE 5.9: SXCD ATS-2 TEST SETUP	87
FIGURE 5.10: ANALOGUE FREQUENCY RESPONSE.....	88

FIGURE 5.11: ANALOGUE TOTAL HARMONIC DISTORTION	89
FIGURE 5.12: ANALOGUE SNR WITH CD-DA AND MACHINE LIMITS	90
FIGURE 5.13: ANALOGUE SNR WITH 70 PS JITTER LIMIT.....	92
FIGURE 5.14: DIGITAL TOTAL HARMONIC DISTORTION	94
FIGURE 5.15: DIGITAL SIGNAL TO NOISE RATIO	95
FIGURE 5.16: EYE DIAGRAM FOR THE DIGITAL OUTPUT SIGNAL.....	96
FIGURE 5.17: FFT OF DIGITAL SIGNAL JITTER, HIGH BANDWIDTH.....	97
FIGURE 5.18: MAXIMUM INAUDIBLE JITTER AMPLITUDES.....	98
FIGURE 5.19: FFT OF DIGITAL SIGNAL JITTER, LOW BANDWIDTH	99
FIGURE 5.20: PROBABILITY CURVE FOR THE DIGITAL SIGNAL JITTER	99
FIGURE 6.1: A COMPARISON OF COMPETITIVE PRODUCTS ANALOGUE THD.....	103
FIGURE 6.2: COMPLETED PROTOTYPE FROM REAR.....	105

The Design of a CD Transport for Audio Applications

A thesis
submitted in partial fulfilment
of the requirements for the Degree
of
Masters of Engineering in Physics
and Electronic Engineering
at the
University of Waikato
By

Carl Dennis Benton



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

2006

ABSTRACT

The project to design a CD transport (CD player) in conjunction with Perreaux Industries came about from the need for a source component in their Silhouette series of products. This project describes the design a high quality CD player, at a low price, to compliment Perreaux's Silhouette series.

A CD drive is selected over a proprietary optical pickup due to the former's low cost and the standardisation of the interface. The control circuitry includes a micro controller and discrete logic to provide the correct data and clock signals to the SPDIF transmitter and DAC circuits. These two circuits provided a high quality analogue output, and facilitate an upgrade path by connecting the SPDIF output to an external DAC.

After three board iterations, a final production ready revision was achieved. The design includes a high quality toroidal transformer, low jitter crystal oscillator, and a very high quality SPDIF pulse transformer output. The design also allows a remote input to control the player, and an optional digital cable via an RJ45 connector to provide synchronisation with a future design of the SXD2 DAC module, or to transmit SPDIF to a remote location.

The specifications of the final design were higher than expectations. The digital output boasts equal or superior performance to competitive products in the same price range, with the analogue output attaining exceptionally high performance.

ACKNOWLEDGEMENTS

There are several parties whom I wish to thank for the opportunity to work on such a quality and educational project. Firstly, my supervisor Dale Carnegie, who has been a mentor to me at the University of Waikato for several years throughout the course of my degree, and who set up both my second work placement and this masters project with Perreux Industries Ltd. Also despite Dale's relocation to Wellington, he remained my supervisor and worked hard to give me a great deal of attention. I hope Dale and his family enjoy their new life in the capital.

I would also like to thank Martin Van Rooyen, the Managing Director at Perreux Industries Ltd, for the opportunity to work on the project. Also Clinton Jenson (Design Engineer) for working with me during my time at Perreux. I wish both of them the best for the future.

I would also like to thank my partner (fiancé at the time of writing this report) Christy Brett for her support and understanding during the time I was working in Dunedin with Perreux. For the duration of the project Christy remained in Hamilton also working on a master's project, and despite having over 1200 km between us for 8 months, we finished on a high note when we got engaged. I would also like to thank Christy for helping proof read the thesis, and for simply being there. Christy, I love you.

DEDICATION

I would like to dedicate this thesis to both my parents for the excellent job they have done over the past 23 years. My Mother is the most understanding woman I know and has always shown an interest in my passion, electronics. My Father introduced me to kit-set construction, which was most likely to stop me from dismantling my electronic toys. However, it is what started the ball rolling, and my Father has always encouraged me to use my full potential.

TABLE OF CONTENTS

ABSTRACT	iii
ACKNOWLEDGEMENTS	v
TABLE OF CONTENTS	vii
LIST OF TABLES	xi
LIST OF FIGURES	xiii
1 INTRODUCTION	1
1.1 SCOPE.....	1
1.2 COMPANY OVERVIEW	1
1.3 THE SILHOUETTE SERIES	2
1.4 PROJECT REQUIREMENTS	3
1.5 THESIS STRUCTURE	4
2 BACKGROUND	7
2.1 COMPETITIVE PRODUCTS	7
2.1.1 <i>Cyrus Audio CD 6s</i>	7
2.1.2 <i>Marantz CD5400</i>	8
2.1.3 <i>Cambridge Audio Azur 640C</i>	8
2.1.4 <i>NAD C521BEE</i>	9
2.1.5 <i>Arcam Diva CD73</i>	9
2.1.6 <i>Denon DCD-685</i>	10
2.1.7 <i>Rotel RCD-1072</i>	10
2.1.8 <i>Summary</i>	11
2.2 A NOTE ON HI-FI.....	12
2.3 DESIGN CONSTRAINTS.....	12
2.3.1 <i>Costs</i>	12
2.3.2 <i>Component sourcing</i>	13
2.3.3 <i>System Clock</i>	13
2.4 THE OPTICAL MECHANISM.....	13
2.5 OTHER BACKGROUND INFORMATION	14
2.5.1 <i>The Compact Disc</i>	14
2.5.2 <i>Stream Is Accurate</i>	15

2.5.3 <i>Jitter</i>	16
2.5.4 <i>Digital Audio Transmission</i>	17
3 HARDWARE DESCRIPTION	19
3.1 OVERVIEW	19
3.2 THE ATAPI INTERFACE.....	20
3.3 THE CONTROLLER	20
3.4 DIGITAL OUTPUT	22
3.4.1 <i>SPDIF Chip</i>	22
3.4.2 <i>Output Circuit</i>	22
3.5 ANALOGUE OUTPUT	25
3.5.1 <i>Digital to Analogue Converter</i>	25
3.5.2 <i>Output Filter</i>	26
3.5.3 <i>Analogue Power Supply</i>	29
3.6 INTER CIRCUIT DIGITAL AUDIO TRANSMISSION	30
3.6.1 <i>Serial Formats</i>	30
3.6.2 <i>Parallel to Serial Conversion</i>	30
3.7 CLOCK GENERATION	32
3.8 MASTER CLOCK.....	34
3.9 USER INTERFACE	35
3.10 POWER SUPPLY	37
3.10.1 <i>Circuit Overview</i>	37
3.10.2 <i>Transformer Rating</i>	38
3.10.3 <i>Voltage Regulators</i>	39
3.10.4 <i>Filtering</i>	40
3.11 REMOTE INPUT	41
3.12 RJ45 CONNECTOR	42
3.13 PCB LAYOUT	43
3.14 INTERCONNECTS	45
3.15 DIMENSIONS	45
3.16 FRONT PANEL	46
4 FIRMWARE DESCRIPTION	47
4.1 OVERVIEW.....	47
4.1.1 <i>Restrictions</i>	48

4.1.2 <i>Interrupt Structure</i>	48
4.1.3 <i>Nested Interrupts</i>	49
4.1.4 <i>Code Definitions</i>	50
4.2 ATAPI INTERFACE	50
4.2.1 <i>Register Access</i>	51
4.2.2 <i>Packet Commands</i>	53
4.3 SYSTEM CONFIGURATION	55
4.4 KEY PROCESSING	56
4.4.1 <i>Play Key</i>	57
4.4.2 <i>Stop Key</i>	58
4.4.3 <i>Skip Keys</i>	58
4.4.4 <i>Scan Keys</i>	60
4.4.5 <i>Eject Key</i>	62
4.5 READING THE TABLE OF CONTENTS	62
4.6 AUDIO PLAYBACK	64
4.6.1 <i>Audio Buffer</i>	65
4.6.2 <i>Data Setup</i>	67
4.6.3 <i>Timing Restrictions</i>	68
4.6.4 <i>Actual Timing</i>	70
4.6.5 <i>Current Audio Position</i>	71
4.6.6 <i>Emphasis Control</i>	72
4.7 USER INTERFACE	73
4.8 REMOTE INPUT	75
4.9 THE LED	77
5 FINAL DESIGN PERFORMANCE	79
5.1 PHYSICAL CONSTRUCTION	79
5.2 SXCD OPERATION	82
5.2.1 <i>CD formats</i>	83
5.3 METHOD OF MEASUREMENT	85
5.4 ANALOGUE OUTPUT	88
5.4.1 <i>Frequency Response</i>	88
5.4.2 <i>Total Harmonic Distortion</i>	89
5.4.3 <i>Signal to Noise Ratio</i>	90
5.4.4 <i>Master Clock Jitter</i>	91

5.5 DIGITAL OUTPUT	93
5.5.1 Total Harmonic Distortion	93
5.5.2 Signal to Noise Ratio	94
5.5.3 Digital Signal Quality.....	95
5.5.4 Digital Signal Jitter	96
6 CONCLUSION	101
6.1 REVIEW	101
6.2 ANALYSIS OF RESULTS	101
6.2.1 Signal to Noise Ratio	102
6.2.2 Total Harmonic Distortion	102
6.2.3 Other Digital Parameters.....	103
6.3 CIRCUIT OPERATION	104
6.4 FUTURE IMPROVEMENTS.....	105
6.5 SUMMARY.....	106
7 REFERENCES	109
8 ACRONYMS.....	113
9 APPENDICES.....	117
9.1 APPENDIX A - PHILIPS L1210 QUOTATION:	117
9.2 APPENDIX B - SCHEMATICS	119
9.2.1 Main Schematic	119
9.2.2 Power Supply.....	120
9.2.3 Outputs	121
9.3 APPENDIX C – PCB LAYOUT	123
9.4 APPENDIX D – LETTER OF RECOMMENDATION	125
9.5 APPENDIX E – PROJECT CD CONTENTS	127

LIST OF TABLES

TABLE 3.1: SUMMARY OF CONTROLLER PIN CONNECTIONS	21
TABLE 3.2: SUMMARY OF REQUIRED CLOCK SIGNALS.....	32
TABLE 3.3: CD DRIVE CURRENT REQUIREMENTS.....	38
TABLE 4.1: LIST OF INTERRUPTS	49
TABLE 4.2: COMMAND BLOCK REGISTER LIST	51
TABLE 4.3: PACKET COMMANDS USED BY THE CONTROLLER	54
TABLE 4.4: LEAD IN AREA (TOC), SUB CHANNEL Q FORMATS.....	63
TABLE 4.5: LEAD IN AREA (TOC) POINT VALUE DESCRIPTIONS	63
TABLE 4.6: LED STATUS SUMMARY	77
TABLE 5.1: TOC FOR EXAMPLE MIXED MODE CD.....	84
TABLE 5.2: <i>TRACKINFO</i> ARRAY CONTENTS.....	84
TABLE 5.3: <i>LEADOUTTRACK</i> ARRAY CONTENTS.....	84
TABLE 5.4: EXTRACT FROM COLDPLAY X&Y CD TOC	85
TABLE 6.1: COMPARISON FOR THE CD PLAYERS DISCUSSED IN SECTION 2.1	104

LIST OF FIGURES

FIGURE 1.1: PERREAUX R200i INTEGRATED AMPLIFIER.....	1
FIGURE 1.2: THE SILHOUETTE SERIES SX25i	2
FIGURE 2.1: CYRUS AUDIO CD6 CD PLAYER.....	7
FIGURE 2.2: MARANTZ CD5400 CD PLAYER	8
FIGURE 2.3: CAMBRIDGE AUDIO AZUR 640C CD PLAYER	9
FIGURE 2.4: NAD C521BEE CD PLAYER	9
FIGURE 2.5: ARCAM DIVA CD73 CD PLAYER	10
FIGURE 2.6: DENON DCD-685 CD PLAYER.....	10
FIGURE 2.7: ROTEL RCD-1072 CD PLAYER.....	11
FIGURE 2.8: A JITTERED CLOCK COMPARED WITH AN IDEAL CLOCK.	16
FIGURE 2.9: THE COMPONENTS OF JITTER	17
FIGURE 3.1: SYSTEM BLOCK DIAGRAM.....	19
FIGURE 3.2: THE CONTROLLER	21
FIGURE 3.3: SPDIF OUTPUT CIRCUIT	23
FIGURE 3.4: THEVENIN EQUIVALENT SPDIF OUTPUT	23
FIGURE 3.5: SIMULATED SPDIF SIGNAL (LOADED AND UNLOADED).....	24
FIGURE 3.6: DAC MFB FILTER	26
FIGURE 3.7: FILTER RESPONSE TO A STEP INPUT	27
FIGURE 3.8: BUTTERWORTH, BESSEL AND CHEBYSHEV RESPONSE CURVES.....	27
FIGURE 3.9: SIMULATED FREQUENCY RESPONSE	28
FIGURE 3.10: ANALOGUE POWER SUPPLY.....	29
FIGURE 3.11: I2S FORMAT	30
FIGURE 3.12: SERIAL DATA GENERATION CIRCUIT.....	31
FIGURE 3.13: TIMING OF THE SHIFT REGISTER SIGNALS	32
FIGURE 3.14: CLOCK GENERATION CIRCUIT	33
FIGURE 3.15: <i>BITCLK</i> AND <i>BITCLK_INV</i> PHASE RELATIONSHIP	34
FIGURE 3.16: PHASE NOISE FOR A 10 MHz CFPS-73	35
FIGURE 3.17: IMAGE OF USER INTERFACE BUTTONS.....	36
FIGURE 3.18: USER INTERFACE CIRCUIT	36
FIGURE 3.19: POWER SUPPLY CIRCUIT	37
FIGURE 3.20: POWER SUPPLY REGULATOR HEATSINKS	40
FIGURE 3.21: LOCAL L-C FILTERS	40

FIGURE 3.22: A SIMPLE MODULATED SIGNAL.....	41
FIGURE 3.23: REMOTE INPUT CIRCUIT.....	41
FIGURE 3.24: SIMULATED REMOTE IN FOR 5 V AND 12 V SIGNALS	42
FIGURE 3.25: RJ45 CONNECTOR CIRCUIT.....	43
FIGURE 3.26: PCB CIRCUIT OVERLAY	44
FIGURE 3.27: MAIN SUB-CIRCUITS ON THE PCB	44
FIGURE 4.1: FIRMWARE OVERVIEW STATE DIAGRAM.....	47
FIGURE 4.2: REGISTER READ AND REGISTER WRITE FUNCTIONS.....	52
FIGURE 4.3: ROUTINES TO WAIT FOR A GIVEN STATUS BIT.....	52
FIGURE 4.4: READ WORD FUNCTION.....	53
FIGURE 4.5: SENDPACKETCOMMAND FUNCTION	54
FIGURE 4.6: PLAY KEY PROCESSING FLOW DIAGRAM	58
FIGURE 4.7: FF AND RW KEY FUNCTIONALITY DURING AUDIO PLAYBACK.....	59
FIGURE 4.8: SCAN FUNCTION FLOW DIAGRAM	61
FIGURE 4.9: PLAY OPERATION STATE DIAGRAM	65
FIGURE 4.10: BUFFER OVERVIEW.....	66
FIGURE 4.11: CONTROLLER MEMORY ORGANISATION	66
FIGURE 4.12: LOAD SAMPLE CODE IN ASSEMBLY LANGUAGE	69
FIGURE 4.13: SR_EN INTERRUPT WORST CASE TIMING.....	70
FIGURE 4.14: UPDATE EMPHASIS FUNCTION	72
FIGURE 4.15: USER INTERFACE INTERRUPT ROUTINE STATE DIAGRAM.....	73
FIGURE 4.16: PUSH KEY AND POP KEY FUNCTIONS	75
FIGURE 4.17: REMOTE INPUT INTERRUPT ROUTINE STATE DIAGRAM.....	76
FIGURE 4.18: SAMPLE LOCATION FOR REMOTE INPUT	77
FIGURE 5.1: PROTOTYPE RUNNING WITH TEMPORARY INTERFACE	79
FIGURE 5.2: PCB OUTLINE AND CRITICAL DIMENSIONS	80
FIGURE 5.3: CD DRIVE INTERCONNECTS.....	81
FIGURE 5.4: COMPLETED POWER SUPPLY	81
FIGURE 5.5: SXCD MAIN OPERATION STATE DIAGRAM.....	82
FIGURE 5.6: SXCD STANDBY STATE DIAGRAM.....	82
FIGURE 5.7: THE SXCD UNDERGOING TESTS	86
FIGURE 5.8: TYPICAL ATS-2 TEST SETUP	87
FIGURE 5.9: SXCD ATS-2 TEST SETUP	87
FIGURE 5.10: ANALOGUE FREQUENCY RESPONSE.....	88

FIGURE 5.11: ANALOGUE TOTAL HARMONIC DISTORTION	89
FIGURE 5.12: ANALOGUE SNR WITH CD-DA AND MACHINE LIMITS	90
FIGURE 5.13: ANALOGUE SNR WITH 70 PS JITTER LIMIT.....	92
FIGURE 5.14: DIGITAL TOTAL HARMONIC DISTORTION	94
FIGURE 5.15: DIGITAL SIGNAL TO NOISE RATIO	95
FIGURE 5.16: EYE DIAGRAM FOR THE DIGITAL OUTPUT SIGNAL.....	96
FIGURE 5.17: FFT OF DIGITAL SIGNAL JITTER, HIGH BANDWIDTH.....	97
FIGURE 5.18: MAXIMUM INAUDIBLE JITTER AMPLITUDES.....	98
FIGURE 5.19: FFT OF DIGITAL SIGNAL JITTER, LOW BANDWIDTH	99
FIGURE 5.20: PROBABILITY CURVE FOR THE DIGITAL SIGNAL JITTER	99
FIGURE 6.1: A COMPARISON OF COMPETITIVE PRODUCTS ANALOGUE THD	103
FIGURE 6.2: COMPLETED PROTOTYPE FROM REAR.....	105

1 INTRODUCTION

1.1 SCOPE

The project is developed in conjunction with Perreux Industries Limited [1], a high-fidelity audio company, who design and produce leading edge audio products. The newest edition to Perreux's product range is the Silhouette series, a range of products targeted at consumers looking in the lower price ranges, or for smaller units. The goal of this project is to design a high quality CD player (the SXCD) for the Silhouette series. Because the series does not currently include a CD player, this project will complete the series and offer a complete system solution to the customer.

1.2 COMPANY OVERVIEW

Perreux is a manufacturer of hi-fi audio amplifiers, preamplifiers, CD players and other hi-fi audio equipment. A fine example of a Perreux product is the R200i 200 W integrated amplifier shown in Figure 1.1.



FIGURE 1.1: PERREUX R200i INTEGRATED AMPLIFIER

The company originated in Nelson in 1974 where it quickly gained a reputation for building highly reliable and well constructed amplifiers. In 1976 Perreux built the first NZ made large PA system which was capable of outputting 900 W RMS. The company expanded, moving to MOSFET technology in 1979 and entering the export market in the early 1980's. The first preamplifier (the SM2) was introduced in 1983, and by 1985 Perreux was exporting by the container load (mainly to the USA) and at its peak was producing over 20 units a day. In 1991 the factory was relocated to Auckland where the company embarked in further research, releasing a number of new products including the Silhouette range of entry-level products. Perreux moved to Dunedin in 2005 and continues to invest in research and design, increasing the quality and range of their products. To this day the mission of Perreux is the perfect re-creation of a musical event.

1.3 THE SILHOUETTE SERIES

The Silhouette series of products [2] has been designed to be a range of high quality, yet affordable, modules to either complement an existing system, or to be a starting point for a new hi-fi enthusiast. Currently the series includes a headphone amplifier (SXH2), a line stage buffer (SXL2), a 25 W stereo integrated amplifier (SX25i) shown in Figure 1.2, and a digital to analogue converter (DAC) with both USB connectivity (SXD2).



FIGURE 1.2: THE SILHOUETTE SERIES SX25i

A 60 W mono-block amplifier is currently being designed, which originates from the design of the SX25i, and is intended to provide a higher power solution for customers wishing to run larger speakers. To provide volume control functionality with the mono-block amplifiers, a passive preamplifier is also being developed. Both of these products will be released shortly.

The goal of the SXCD CD Player is to complete the system, allowing users to match it with an SX25i for a simple system, upgrade it with the SXD2, or expand it with the mono-blocks or line stage buffer for improved performance in sound. When the SXCD is made available, many system configurations would be possible to suit a large array of situations ranging from a computer based office setup to a small lounge stereo system.

1.4 PROJECT REQUIREMENTS

The requirements for the SXCD project were set out by Perreux as a list of features that must be implemented on the design. The list is as follows:

- A high quality (0.001 % THD or less) SPDIF output for connectivity to the SXD2 or alternative DAC module.
- A moderate quality (0.003 % THD or less) analogue output for stand alone operation.
- A simple method to load a CD into the device.
- A simple user interface on the front and a remote input connection on the back.
- Fit into the current Silhouette series look, feel, and (preferably) size.

The target price for the SXCD is in the range of \$800-\$1200, and it is envisioned that it will compete with products in the market that are priced below \$2000. Perreux have taken a minimalist approach with the Silhouette series, creating products that are simple to use and are physically (and visually) free from clutter. To continue this theme (and due to size and cost constraints) it is decided that no display will be used on the front panel.

All the current Silhouette products are the same size, being 45 mm high, 210 mm wide and 150 mm deep with the same base front panel design. It is important that the SXCD continues this trend with the same front panel dimensions.

The aesthetic aspects of the final product do have an effect on this project since the dimensions are constrained and hence component selection and placement must conform to these limitations. However, the aesthetic aspects themselves (such as chassis and front panel design) are not part of the project requirements as they are handled by other designers within Perreaux.

1.5 THESIS STRUCTURE

Chapter One, **Introduction**, gives an overview of the project, while introducing Perreaux Industries Ltd and the Silhouette series. The project requirements are outlined as provided by Perreaux, and an outline of the thesis structure is given.

Chapter Two, **Background**, provides information on various competitive products that are currently in the target market of the SXCD. These products are considered in terms of both their technical performance and operation, and are summarised to obtain an idea on what is desirable in the current market. The inspiration behind using a CD drive instead of a proprietary optical pickup is discussed and also information on technologies that are directly related to the SXCD project are described such as the compact disc, clock jitter and the SPDIF digital audio format.

Chapter Three, **Hardware Description**, gives a detailed description of the hardware aspects involved in designing the project. The circuit design can be divided into separate functions or sub-circuits. The chapter is broken down into sections, where each section focuses on one of these sub-circuits.

Chapter Four, **Firmware Description**, details the operation of the firmware which is programmed into the microcontroller. The chapter describes how the controller functions, along with the limitations and problems encountered throughout the

project and how these were overcome. Each aspect of the firmware is described at an operational level and, where relevant, code fragments are provided.

Chapter Five, **Final Design Performance**, presents the results obtained from the final revision of the SXCD. The operation of the CD player in terms of its response speed is measured, and the measurement equipment available at Perreux is used to obtain the performance results for both the analogue and digital outputs.

Chapter Six, **Conclusion**, summarises the SXCD's specifications and compares them with the products which were outlined in Chapter Two. The final performance of the project is discussed reflecting on the initial requirements set at the beginning of the project. Future improvements are considered and the project as a whole is summarised and the thesis concludes with a comment from the managing director of Perreux Industries Ltd.

2 BACKGROUND

2.1 COMPETITIVE PRODUCTS

As part of any new product design process, one must investigate the range of competitive products currently on the market to gain an idea on how a new product must perform in order to be successful. Details from several competitive products that fall into the lower price range (of below \$2000) are given below where the products are compared in both their specifications and overall performance.

2.1.1 CYRUS AUDIO CD 6S

The Cyrus Audio CD 6 CD Player [3], as shown in Figure 2.1, retails for around NZ\$2200. Although it is a bit more expensive than the target price range, it provides some competition due to its compact size and simplistic nature. The design includes quality components and eight regulated power supplies. The chassis is built with metal components on all six sides to create an electrical faraday cage and protect internal circuitry from stray interference. Cyrus also acknowledges the need for an accurate clock and therefore includes a remote re-clocking circuit at the DAC stage.



FIGURE 2.1: CYRUS AUDIO CD6 CD PLAYER

The CD 6 boasts specifications of 0.002 % Total Harmonic Distortion (THD) at 1 kHz, a Signal to Noise ratio (SNR) of -110 dB, and clock jitter of less than 100 ps. The unit includes an SPDIF optical output, a custom LCD display, a

24-bit current output DAC and a Cyrus system remote control, but does not support CDRW discs.

2.1.2 MARANTZ CD5400

The Marantz CD5400 [4], shown in Figure 2.2, retails for around NZ\$599, and although it is an older product in the market it still gets reviews as late as 2005 [5]. For a CD player at such a low price it is popular due to its looks and sound quality. It offers numerous features such as CD-Text compatibility, both optical and coaxial digital outputs, a headphone output with volume control, and several different playback options. The specifications state a 0.0025 % THD (although the frequency is not given), a signal to noise ratio of -110 dB, and a frequency range of 20 Hz - 20 kHz (however the deviation is not stated).



FIGURE 2.2: MARANTZ CD5400 CD PLAYER

2.1.3 CAMBRIDGE AUDIO AZUR 640C

The Cambridge Audio Azur series is relatively new as it was introduced in 2003. The product to compare is the 640C CD player [6] (shown in Figure 2.3), retails for around NZ\$899 and has been considered as the world's best budget CD player [6]. The chassis is completely made from metal, with a solid aluminium front panel that is matched with an aluminium faced remote control. On the 640C product page at Cambridge Audio's website, it boasts a completely separate power supply and upgraded audio filter and regulator stages, with the whole system housed on an acoustically damped chassis.

The analogue stage incorporates a WM8740 24-bit 192 kHz DAC from Wolfson. The frequency response has a 1 dB ripple across the audio band, the THD is reasonable at 0.002 % at 1 kHz, the SNR is -100 dB and the jitter is stated as

being below 260 ps. This overall performance has however been classed as one of the best for the product range that the Azur 640C sits in.



FIGURE 2.3: CAMBRIDGE AUDIO AZUR 640C CD PLAYER

2.1.4 NAD C521BEE

The NAD C521BEE [7] (shown in Figure 2.4) is another low cost CD player retailing at around NZ\$499 where the player has been designed to sit at the “sweet spot” in terms of price and performance. It is housed in a full unit size chassis and includes an EL backlit display, a remote control, a coaxial digital output and an analogue output. The design includes a digital output transformer, separate analogue and digital power supplies and high grade components, although only a 20-bit DAC is used. As far as the specifications go, the THD is 0.0035 % at 1 kHz, the SNR ratio is -108 dB, the frequency response of 5 Hz – 20 kHz deviates less than 0.5 dB, and NAD also mentions the CIRC error correction that is built into audio CD’s.



FIGURE 2.4: NAD C521BEE CD PLAYER

2.1.5 ARCAM DIVA CD73

The CD73 is the entry-level Diva CD player from Arcam and retails for around NZ\$1499. It features an LED display, 20-track program memory and supports CD text and most CD-R and CD-RW discs. It also includes both optical and coaxial digital outputs as well as two pairs of line level analogue outputs, and comes with

a custom designed remote control. Internally it utilises the WM8740 24-bit Sigma Delta DAC chip from Wolfson Electronics and a carefully designed master clock to provide low jitter (although a jitter figure is not provided). Arcam states figures for THD of 0.005 % and a SNR of -112 dB.



FIGURE 2.5: ARCAM DIVA CD73 CD PLAYER

2.1.6 DENON DCD-685

The DCD-685 from Denon is shown in Figure 2.6 and retails for around NZ\$699. It features an LED display with dimmer function, pitch control, a remote control, headphone output and an optical digital output. It supports CD-R and CD-RW discs and includes one of the PCM1702 20-bit 2-DAC chips from Burr-Brown on each audio channel. The specifications state an SNR of -110 dB and a THD of 0.003 % at 1 kHz. It also states a frequency response of 20 Hz to 20 kHz, however no deviation figure is stated.



FIGURE 2.6: DENON DCD-685 CD PLAYER

2.1.7 ROTEL RCD-1072

The RCD-1072 from Rotel retails for around NZ\$1495. The front of the unit is silver aluminium (with black handles at each side) and has a centre mounted CD tray. It features an LED display and a selection of buttons for various features

such as random, repeat and track programming. It also includes a remote control, a 12 V trigger, and both analogue and digital coaxial outputs.



FIGURE 2.7: ROTEL RCD-1072 CD PLAYER

The analog output uses an 18-bit equivalent sigma delta DAC, which is stated as the PCM-1732 from Burr-Brown. However, details on this part cannot be found on the manufacturer website. The specifications include a frequency response of 20 Hz to 20 kHz within a deviation of 0.5 dB, a SNR of -100 dB, and THD of 0.0045 % at 1 kHz.

2.1.8 SUMMARY

Looking at these products it is apparent that in the Hi-Fi market, it is important to produce a product that not only sounds good, but is also aesthetically pleasing. Unfortunately, comparing each of the products on how well they sound is not easy, so the construction and specifications become the primary focus. Many of these products include a display to show track and time information. A remote control is commonly included, and an optical output is not necessary. Several different types of DAC chips are used in these products, ranging from a 20-bit 2-DAC high precision DAC, to a 24-bit Sigma Delta DAC configuration.

Some of the products include re-clocking circuitry to reduce jitter at the internal DAC stage; however it would be better to minimise the jitter at the clock source instead. A jitter figure of less than a few hundred picoseconds seems to be accepted as high quality. To gain an advantage in this area, a value of less than 100 ps would be desirable.

It is also apparent that many of the products do not provide all the specifications. Usually the reason for doing so is because the specifications are either not so good, or are not obtainable with the company's available resources. The products

state similar specifications with an average of 0.003 % THD at 1 kHz and a SNR of around -107 dB. Only four stated a deviation in frequency response, with the average being 0.6 dB. It is important to provide all the relevant specifications, especially when stating that a large amount of work has been invested in a particular design area. Making such statements without backing them up with the facts can have a negative effect if the buyer is somewhat technically inclined.

2.2 A NOTE ON HI-FI

It is important to note at this stage that as an electronic engineer, a product's performance is typically determined by its specifications. However, in the Hi-Fi realm, it is not this simple, and can be somewhat reversed with better "performance" gained from a product with lower "specifications". This is because the term performance is synonymous to a customer's tastes and a reviewer's preferences in audio sound. A reviewer typically uses words such as "open", "warm", "clean", "grainy" and "momentum" to describe the sound of a product, yet it is very hard to connect such words to a product's specifications. For a CD player proving perfect specifications such as zero distortion (THD) and jitter, infinite dynamic range and a perfectly flat frequency response etc., one would *expect* it to sound the most natural and unimpaired. Therefore, this is what is aimed for in the design of the SXCD.

2.3 DESIGN CONSTRAINTS

2.3.1 COSTS

Because this project is to be an entry level product in the Silhouette range, component costs need be kept to a minimum without compromising the performance of the design. This however, is not difficult and through careful component selection and circuit design both the component count and costs can be minimised.

2.3.2 COMPONENT SOURCING

When designing a product for a small company with a relatively low turnover one must be aware of higher production costs due to lower buying power and restrictions from a supplier's minimum order quantity. In this situation, costs can be reduced by both reusing components from existing products and including fewer component types in more areas of the design. This is one of the main issues that is dealt with in this project, and largely influences the component selection.

2.3.3 SYSTEM CLOCK

The system clock in a CD player is the most critical component in order to minimise timing errors such as jitter and frequency stability. A poor clock can compromise the performance of the whole system, and must be chosen carefully.

2.4 THE OPTICAL MECHANISM

There are three possible solutions for the optical pickup. The first is to use a mechanism solution which does not include the control circuitry for the CD mechanism, the second is the use of a pre-assembled module with simple control and data interface, while the third is to use an existing computer CD drive and interface it to provide the necessary functionality.

A raw CD mechanism contains only the basic mechanical and optical components involved in reading the data from a CD, and requires the controller circuitry and motor drivers to be included into the design. This approach is only a viable solution if the designer is willing to design the control circuit themselves, and is not an option for this project due to limited design time, budget, and the adopted "why re-invent the wheel" approach.

The majority of CD players available today use a CD mechanism module designed for audio applications. Such a module consists of the CD mechanism and the control circuitry in one unit where the control circuitry handles the low level control of the motors and laser and communicates through a simple interface such as the DSA interface used in the Philips L1210 module from Daisy Laser [8].

Although this solution is definitely viable in terms of the mechanical and electrical complexity of the design, the cost and production aspects are not so favourable. Daisy quoted a price of US\$43.85 for 240+ pieces for this L1210 module, and given the project's budget and the fact that these modules tend to change on occasion, this is not an ideal option.

It is therefore decided that the project will be designed around an ordinary computer CD drive such as a CDROM drive or CD Writer. These are well established optical devices used in computer applications for extracting both data and digital audio from CD media. Since the CD drive is a widely used device and is mass produced, they can be obtained for very little cost compared to the mechanisms available. They are also constructed according to the physical dimensions stated in the ATA specification and all use a well established ATAPI interface (see Section 3.2) which is an industry standard and will remain unchanged long into the future. This makes them an extremely attractive solution for a tight budgeted, low volume product. There are other advantages to using a CD drive over a mechanism module, such as a potentially higher quality of audio and versatility of CD formats as they are compatible with all forms of CD's including CDROM, CDR and CDRW. Therefore, any disc that a computer with a CD drive can read is compatible.

2.5 OTHER BACKGROUND INFORMATION

2.5.1 THE COMPACT DISC

All data on a compact disc (CD) are stored in a single spiral track of pits representing logic ones and zeros. These pits are approximately 0.5 μm wide and the pitch between adjacent spirals is 1.6 μm . The spiral runs from the centre, to the outside of the disc, with a length of around 5 km over its 20,000 revolutions. The data are extracted using a laser to generate an 'eye pattern' and the reflected beam is then read by the optical pick up (OPU) and decoded using a CD decoder to create a stream of binary information.

Data errors are decoded and corrected using a well known Cross-Interleaved Reed-Solomon Code (CIRC) algorithm contained in the information, thus

ensuring a reliable data stream. This CIRC information is added to the data when the disc is created. Other information is also included in the bit stream such as position and data type. All this information is stored in a sub-channel byte on each frame, where each bit is assigned to each of the eight sub-channels, designated P, Q, R, S, T, U, V, and W. Spanning over a total of 192 frames, each of these sub-channels make up a 24 byte field. The only sub-channel of use here is the Q sub-channel (called Sub-Q) as it contains information such as the Minute Seconds Frame (MSF) address (see below), the track index and the control field.

A CD containing digital audio (CD-DA), contains data at a sampled rate of 44.1 kHz using 16 bits per sample, corresponding to a data rate of 176.4 kB/s. All data are stored in frames of 2352 bytes in size, where on an audio disc this is entirely CDDA data, corresponding to a rate of 75 frames per second. Each frame is made up of 98 small frames, each containing 24 bytes of data, and for each of these small frames, 588 bits are stored on the CD (the extra bits contain the CIRC and sub-channel information). Therefore a 70 minute, 700 MB, audio CD will actually hold 2.1 GB of information.

When accessing a CD there are two different address formats to use. The Logical Block Address (LBA) format defines the addressing mode of the drive by the linear mapping of the sectors from 0 to n. An LBA address is 4-bytes long which can represent a total of over 4294 million sectors. The second is the MSF format which is expressed as a sector count relative to either the beginning of the CD (absolute) or the beginning of the current track (relative). The format is made up of 3 fields, an F field unit is one frame (also called a sector), an S field unit is one second and is 75 F field units, and an M field unit is one minute and is 60 S field units. An absolute MSF address corresponds to a location on the CD as M minutes, S seconds, and F frames from the start of the CD, and can be as high as 99 minutes, 59 seconds, and 74 frames.

2.5.2 STREAM IS ACCURATE

The structure of the data on a CD is arranged in blocks of 2352 bytes of information where for data CD's this block contains 2048 bytes of user data. The rest of the sector is used for a synchronisation field, sector address tag and an

auxiliary field. In the case of a CD-DA disc however, the entire 2352 bytes of the sector contains audio data and therefore there is no header information. For this reason, if streaming of the data is interrupted and must restart where it was stopped there will be an error in where the drive starts streaming again. This error could be as much as one second due to the uncertainty of the address. If the drive supports the CD-DA Stream-Is-Accurate capability then it contains the necessary circuitry to overcome this error and therefore there will only be a delay due to rotational latency. Fortunately, all modern drives include this functionality and therefore this should not be a problem in the SXCD design.

2.5.3 JITTER

In a digital system, data transmission is robust and information is generally sent and received without any loss of data. The problem of jitter only arises in the ADC or DAC stages where each sample must be converted at a specific time. Jitter is the time variation in the sample clock used in an ADC or DAC and causes samples to be converted at a time other than the ideal. In a typical CD player, jitter is present in both the master clock driving the on-board DAC stage, and the digital output signal. Figure 2.8 shows this jitter on five successive edges where ' p ' is the individual time variation of the actual clock signal from the ideal.

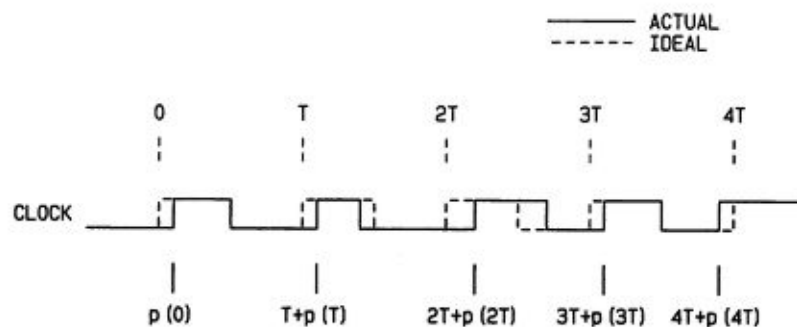


FIGURE 2.8: A JITTERED CLOCK COMPARED WITH AN IDEAL CLOCK.

Total jitter comprises two main types: deterministic jitter and random jitter as shown in Figure 2.9. Random jitter is independent of frequency and is characterised by a Gaussian distribution. It is caused by many noise sources such as thermal noise, power supply noise or noise caused by semiconductor process anomalies.

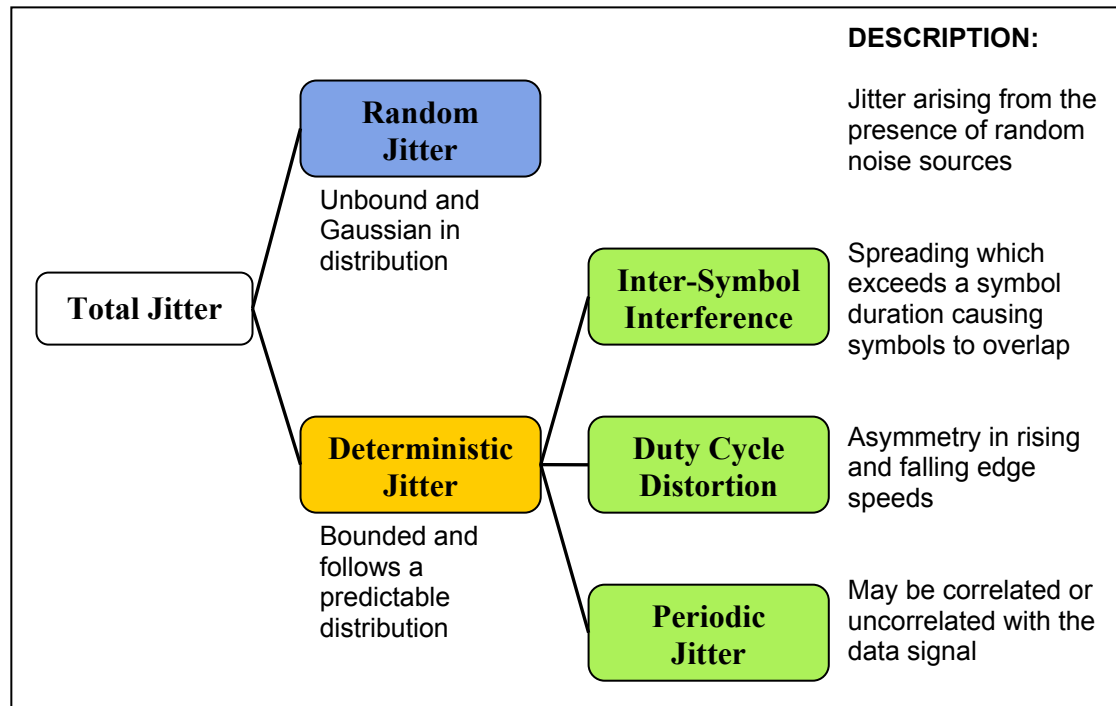


FIGURE 2.9: THE COMPONENTS OF JITTER

Deterministic jitter is bounded in amplitude and is not analysed statistically. It can be broken down into three types: periodic jitter, duty cycle distortion, and inter-symbol interference. Periodic jitter has a signature that repeats at a fixed frequency, and can be caused by sources such as EMI radiation from power supplies, AC power lines and RF signal sources. Duty cycle distortion is the result of asymmetries in clock cycles, and can be caused by differences in rise and fall times and threshold variations of a device. Inter-symbol interference is the variation of each bit width, and can be caused by improper termination or bandwidth limitations. Duty cycle distortion and inter-symbol interference are types of data dependant jitter, where the timing errors are dependant on the data pattern itself.

2.5.4 DIGITAL AUDIO TRANSMISSION

In the high-end audio market, the SPDIF format is used to transmit digital audio between hi-fi audio components. SPDIF is an acronym for Sony Philips Digital Interface and is the consumer version of the AES3 interface [9] used in professional equipment. SPDIF is commonly transmitted over a coaxial cable with either BNC or RCA interconnects, but sometimes an optical connection is used,

and although it has become the standard interface in the hi-fi industry, it is far from ideal. The SPDIF format transmits both the digital data and the clock over a single wire using bi-phase mark encoding, and is susceptible to timing errors such as jitter that corrupt the digital to analogue conversion stage. In a typical system, the DAC circuitry will regenerate a clock signal based on the data stream received and is therefore running as a slave to the CD player. A system that removes jitter much more effectively is called master mode [10], where the master clock is directly driving the DAC stage with the CD player running as a slave to the DAC. In this situation, the timing errors between the CD player and DAC are completely eliminated, although this requires a clock signal to be transmitted separately.

3 HARDWARE DESCRIPTION

3.1 OVERVIEW

The complete system can be broken down into sub-circuits, where each sub-circuit deals with a separate function, and is synchronised to a master clock. Each sub-circuit in the system, along with their corresponding connections, is shown in the block diagram in Figure 3.1. At the heart of the system is the controller, an ATmega8515 microcontroller, which runs the firmware responsible for operating the system. There are two output circuits, one being the SPDIF output based around a digital audio transmitter chip, and the other an analogue output based around a DAC chip. Both of these are driven by a serial data transmission generated by the clock and parallel to serial circuitry. The user interface, CD drive and remote input are directly interfaced to the controller, and an RJ45 Link is available to either receive a master clock, or transmit a SPDIF signal when the need for this functionality arises. Each sub-circuit, along with other hardware aspects of the design, are described in detail in the following sections.

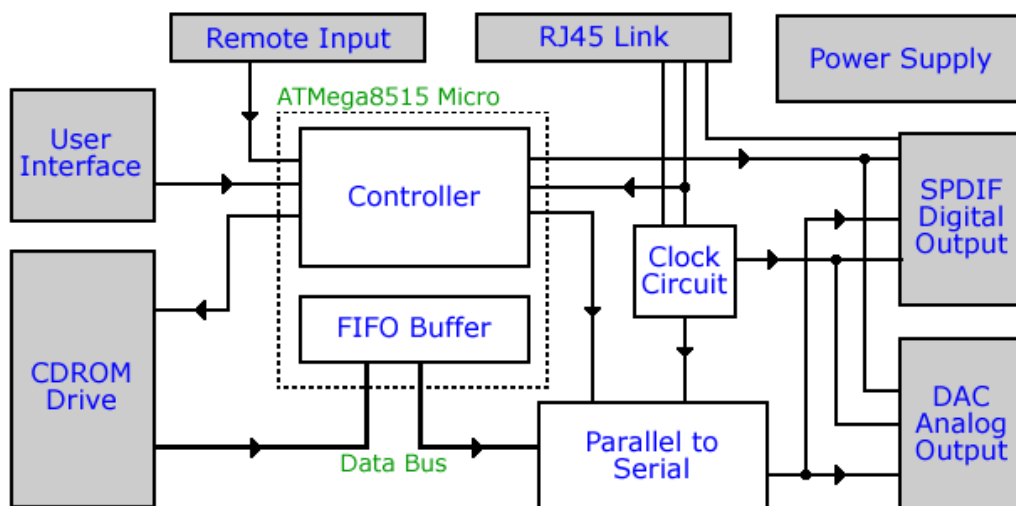


FIGURE 3.1: SYSTEM BLOCK DIAGRAM

3.2 THE ATAPI INTERFACE

The AT Attachment (ATA) interface has been developed for over 10 years and was originally only designed to work with hard disc drives for storage. As the use of multimedia expanded and CD drives became widely used, there was a need for an enhanced protocol, and so the ATA Packet Interface (ATAPI) was developed. This ATAPI protocol consists of a Packet Command feature set and is used to provide extra commands that are not included in the ATA specification to interface removable media drives such as CD/DVD drives, tape drives and ZIP drives. Physically the ATA interface consists of a 16-bit data bus, a 5-bit address bus, a read line and a write line. Other signals are available on the interface such as a Reset line, Interrupt, DMA Acknowledge, DMA Request, IO Ready and Cable Select but are not required, nor connected to the controller in this project. The operation of the ATAPI interface is discussed in detail in Section 4.2.

3.3 THE CONTROLLER

The ATmega8515L [11] microcontroller from Atmel was chosen simply because Perreux currently uses Atmel parts, and the ATmega8515L is already a stocked component. The ATmega8515L is an 8-bit microcontroller based on the AVR enhanced RISC architecture and is the low power version of the ATmega8515. It features 8 Kbytes of Flash, 512 Kbytes of EEPROM, 512 bytes of SRAM, 35 general purpose I/O lines, 32 general purpose working registers and many other features such as counters, USART, SPI interface and interrupts. Instructions are executed in a single clock cycle resulting in a throughput of 1 MIPS per MHz. Running at the maximum speed of 8 MHz gives an instruction execution time of 125 ns. Of the 35 general purpose I/O lines, 16 are used for the data bus, 7 to control the ATA interface, 5 for the SPI port, 3 for the serial data generation, 2 for the user interface, 1 for the remote input, and 1 for driving the LED. These connections are shown on the schematic diagram of the controller in Figure 3.2 and are summarised in Table 3.1.

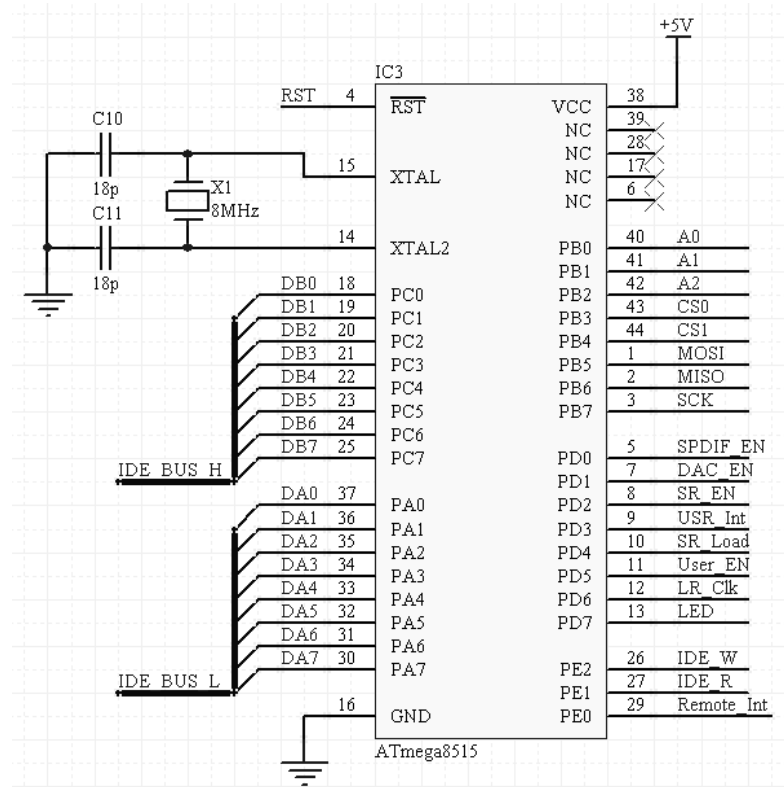


FIGURE 3.2: THE CONTROLLER

Function:	Number of Pins:	Pin Numbers:
Data Bus	16	18-25, 30-37
ATA Control Lines	7	26, 27, 40-44
SPI Port	5	1-3, 5, 7
Serial Data Generation	3	8, 10, 12
User Interface	2	9, 11
Remote Input	1	29
LED	1	13

TABLE 3.1: SUMMARY OF CONTROLLER PIN CONNECTIONS

The three external interrupts are located on pins PD2, PD3 and PE0 (pins 8, 9 and 29 respectively) and are connected in this particular order deliberately to provide the correct interrupt structure. This is because the ATmega8515 interrupt priorities are fixed, with the INT0 (PD2) the highest and INT2 (PE0) the lowest, resulting in a *SR_EN* interrupt having the highest priority, and a remote input interrupt having the lowest. More details on the interrupt structure are provided in section 0. The ATA address lines (pins 40-44) must be connected to port B because they need to be written to in the same manner as a complete data bus. This is possible because the upper three bits are used for the SPI port and are not affected by write operations to the pins themselves. The remaining pins are connected simply to reduce routing complexity on the final circuit board.

3.4 DIGITAL OUTPUT

A digital output is required to provide high quality audio transmission for connecting to a high quality external DAC module such as the Perreux SXD2. Because the SPDIF format has become an industry standard, it must be used in order for the digital output to be compatible with the various external DAC modules currently on the market. There are many digital audio transmitter chips available that transmit SPDIF and accept a range of serial digital audio interfaces, which helps to simplify the design.

3.4.1 SPDIF CHIP

A few digital audio transmitter chips were investigated including the AK4103 from AKM, the CS8406 from Cirrus Logic and the DIT4192 from Texas Instruments. Each chip provides a different package size and range of features such as the range of master clock and bit clock frequencies the chip will accept, however all of them provide the functionality required by the system. The DIT4192 [12] from Texas Instruments was therefore chosen primarily for its good pricing and availability.

The DIT4192 is designed for both professional and consumer operation, and contains a single differential line driver output. It also runs from a single supply voltage of 5 V and is available in a TSSOP-28 package. Although the chip can be used with sampling frequencies up to 192 kHz, it is fixed at 44.1 kHz in the SXCD system due to the CD-DA format. The chip is configured for software operation, where it is connected directly to the controller via an SPI interface. This provides flexibility for future upgrades and control over the transmitted user data.

3.4.2 OUTPUT CIRCUIT

In order to provide isolation and short circuit protection to the SPDIF output circuitry, an output transformer is included. The transformer must be a high quality pulse transformer for digital transmission to keep jitter and signal

distortion to a minimum. There are a few manufacturers who provide such pulse transformers, yet the SC982 from Scientific Conversion [13] is chosen for their good reputation (also note [14]) and optimised design for SPDIF applications between 32 and 192 kHz. The SC982 has a turns ratio of 1:2, which can be used in reverse to halve the signal voltage. It also has a very low leakage inductance, and is available in a surface mount package.

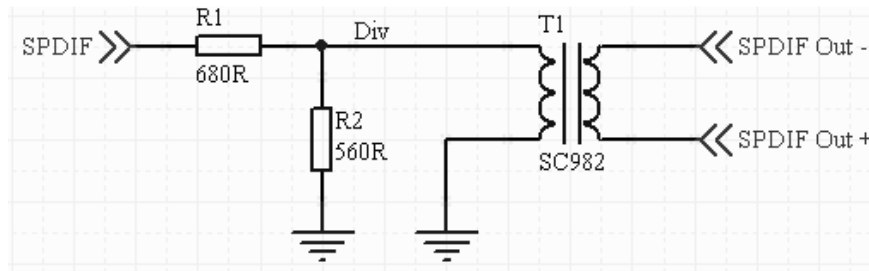


FIGURE 3.3: SPDIF OUTPUT CIRCUIT

There are two requirements of the output circuit: first it must meet the SPDIF specification [15] for an output voltage of 400 mV – 600 mV p-p, and second it must be matched to the required line impedance of a 75 Ω cable. To satisfy both these requirements from a supply voltage of 5 V and a 2:1 transformer ratio, a voltage divider is included to produce the final output circuit shown in Figure 3.3, which is a common unbalanced SPDIF configuration [16].

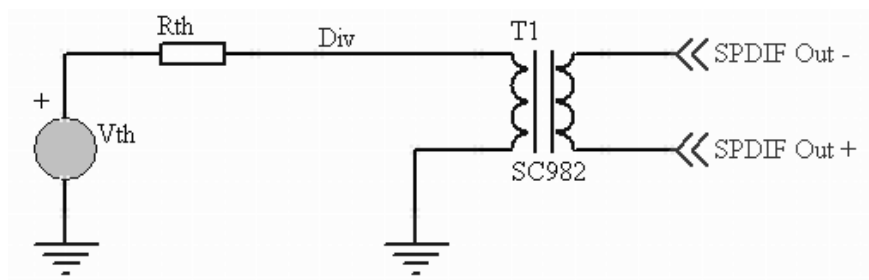


FIGURE 3.4: THEVENIN EQUIVALENT SPDIF OUTPUT

In this configuration a Thevenin equivalent circuit exists on the primary side of the transformer, as shown in Figure 3.4, where the Thevenin resistance (R_{th}) is given by the parallel combination of R_1 and R_2 , and the Thevenin voltage is given by the voltage divider created by R_1 and R_2 . It shows that if R_{th} is matched to the transformer primary impedance when the secondary is under load, the

voltage present at the transformer will be half the unloaded voltage, and that V_{th} is the same as the SPDIF signal voltage of 5 V.

$$R_{th} = Z_p = Z_s \left(\frac{N_p}{N_s} \right)^2 \quad \text{Equation 3-1}$$

$$V_{out} = \frac{N_s}{N_p} V_{th} \frac{R_2}{R_1 + R_2} \quad \text{Equation 3-2}$$

In order to match the 75 Ohm line impedance (Z_s) required for a setup using coaxial cable, the Thevenin equivalent resistance (R_{th}) must be 300 Ω according to Equation 3-1, given the transformer winding ratio ($N_p:N_s$) of 2:1. The loaded output voltage is chosen to be 550 mV to provide a high signal voltage within the specified limits, which will result in an unloaded signal of 1.1 V. Using this figure for V_{out} and a V_{th} of 5 V, the resistor values can be calculated from Equation 3-2 to be $R_1 = 680 \Omega$ and $R_2 = 535 \Omega$.

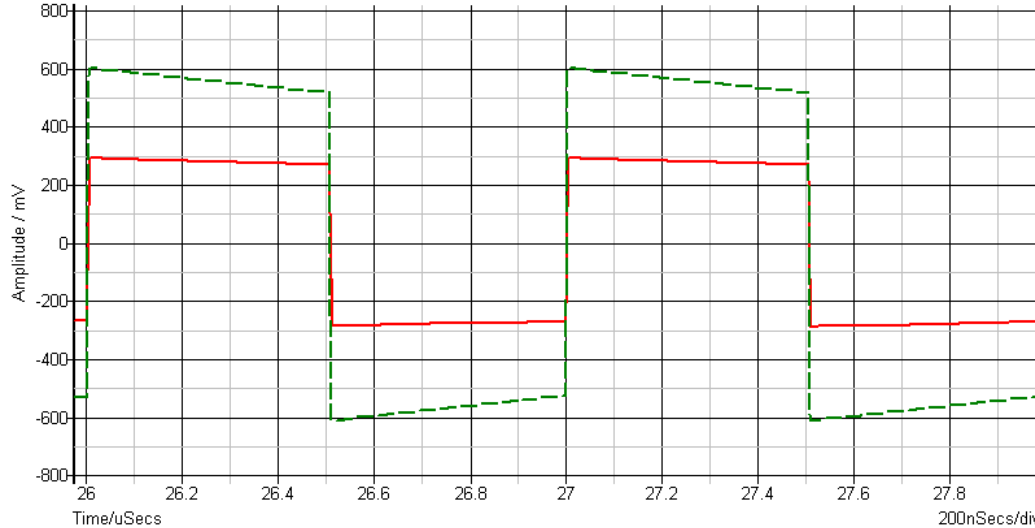


FIGURE 3.5: SIMULATED SPDIF SIGNAL (LOADED AND UNLOADED)

The final values chosen, shown in Figure 3.3, are $R_1 = 680 \Omega$ and $R_2 = 560 \Omega$ as these are the closest standard resistor values. The simulation of the output of the resulting circuit is shown in Figure 3.5. The solid line is the loaded output voltage, and it is 550 mV in amplitude, which is just below the maximum of 600 mV. The

dotted line is the unloaded signal with the expected amplitude of 1.1 V. This doubling in amplitude shows that the circuit is matched to the line impedance.

3.5 ANALOGUE OUTPUT

An analogue output is required in order for the product to operate as a stand-alone CD player, yet it does not need to be of the high quality standard that the SXD2 possesses as the SXD2 is intended to be an upgrade path providing a higher quality output. For this reason, it was an initial thought that the analogue output from the CD drive could be used directly, but this leads to the problem that both the digital and analogue outputs cannot operate at the same time. This is because there are different ATAPI commands for the analogue output such as “PLAY AUDIO” and for reading the digital data via the ATA interface such as “READ CD”, and of course these commands cannot be processed concurrently. For this reason, and also because the analogue output quality of the CD drive may be very poor, an onboard DAC is included in the design.

3.5.1 DIGITAL TO ANALOGUE CONVERTER

As the purpose of the DAC is to provide a good quality analogue output at a low cost, a single chip DAC solution with a low component count is desirable. Several different DAC chips from manufacturers such as Texas Instruments, AKM and Analog Devices were investigated. The PCM1780 from Texas Instruments was chosen due to its simple nature and high performance for a low cost. The PCM1780 is a 24-bit stereo sigma-delta digital to analogue converter running from a 5 V supply in a SSOP16 package. The chip includes the option of 8× over sampling and although sample rates up to 200 kHz are supported, the system is limited to 44.1 kHz due to the CD-DA format. The PCM1780 is interfaced in a similar manner to the SPDIF chip. It is configured for software operation where it is set up through an SPI interface and data are sent to it by one of several serial digital audio interfaces. The advantage of this is that the two output sections can be run in parallel without the need for separate interfaces, apart from their individual SPI enable lines from the controller.

3.5.2 OUTPUT FILTER

Although the DAC chip has built-in digital filtering, an output filter is required to provide extra attenuation of the switching frequency. By making this filter active it also acts to buffer the signal output from the DAC chip. To keep the circuit simple and reduce component count a second order active filter is chosen using a single op-amp. The two main configurations that exist are the Sallen-Key and Multiple Feedback (MFB), which are not only well documented but are easy to implement with the design software that is readily available. A multiple feedback topology, shown in Figure 3.6, is used over other configurations for its improved high frequency response and low sensitivity to component variations. By using a MFB circuit topology the signal is subject to a 180 degree phase shift through the filter, however this can be compensated in the DAC by setting its Output Phase Select bit (*DREV* bit on P-24 of the PCM1780 datasheet) to logic one to invert the output signal. The circuit is biased at V_{COM} , which is a 2.5 V bias voltage provided from the DAC, decoupled by a 47 μ F capacitor.

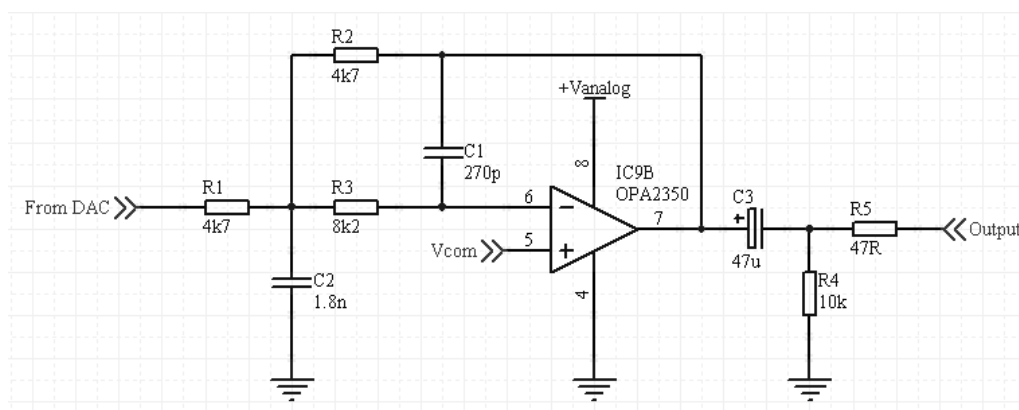


FIGURE 3.6: DAC MFB FILTER

In this situation, where the Nyquist frequency (22.05 kHz) is very close to the cut off frequency, it is important to choose a filter with a high roll off. The Chebyshev filter has the advantage of having a high roll off over other filter types. The disadvantages of the Chebyshev filter is a large pass band ripple and overshoot when presented with a step input. This overshoot is shown in Figure 3.7, where the overshoot from a Chebyshev filter is much greater than that of a Butterworth and Bessel.

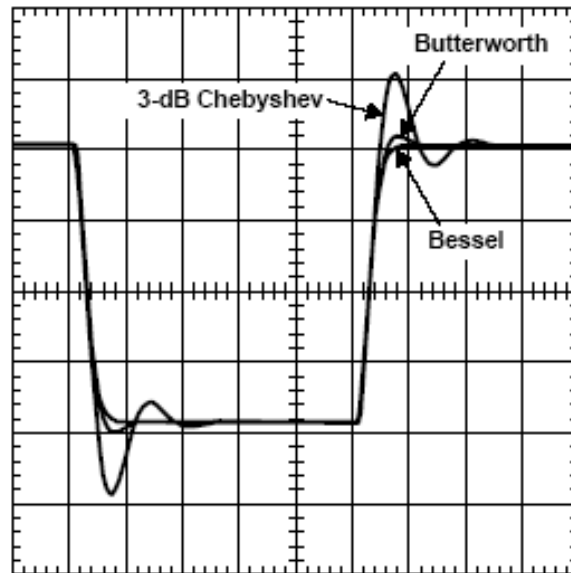


FIGURE 3.7: FILTER RESPONSE TO A STEP INPUT

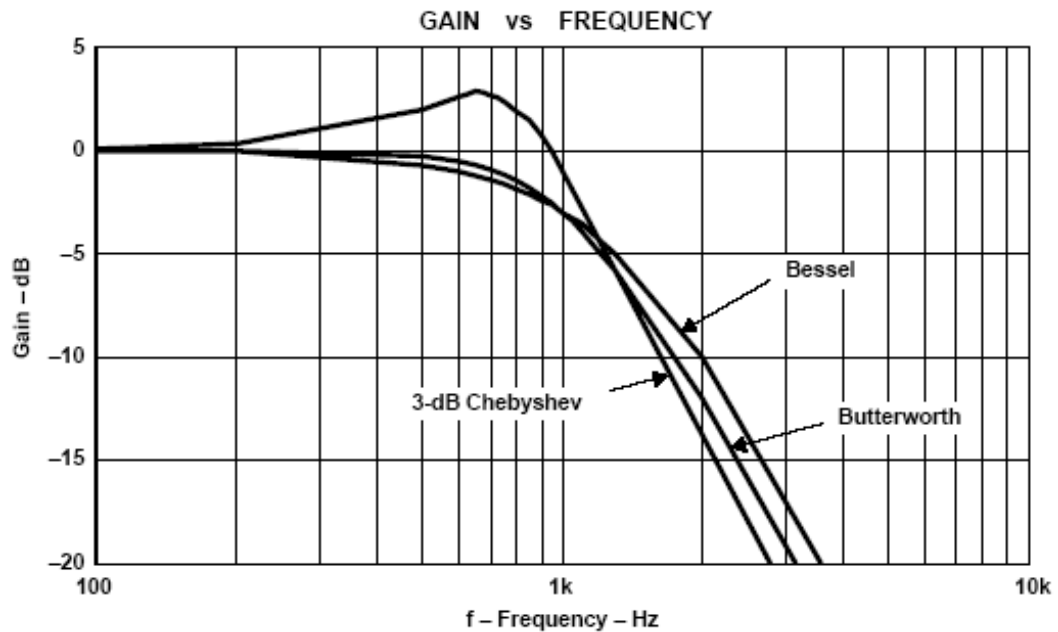


FIGURE 3.8: BUTTERWORTH, BESSEL AND CHEBYSHEV RESPONSE CURVES

A Butterworth filter however has the desirability of a maximally flat pass band, and only a small overshoot with a step input, however it has the disadvantage of a slow roll off. These curves are shown in Figure 3.8 where second order Chebyshev, Butterworth and Bessel filters are shown. The desired filter response can be obtained by designing a filter that sits in between that of the Chebyshev and the Butterworth configurations, giving a good compromise between a flat pass band and an initially high roll-off.

$$H(f) = \frac{-R_2}{R_1} \frac{1}{(j2\pi f)^2 (R_2 R_3 C_1 C_2) + j2\pi f \left(R_3 C_1 + R_2 C_1 + \left(\frac{R_2 R_3 C_1}{R_1} \right) \right) + 1} \quad \text{Equation 3-3}$$

The transfer function shown in Equation 3-3 can be used to generate each of these three filters by modifying the component values. However, the process for selecting component values and trying to find a curve that results in the best performance using this equation can be extremely tedious, so instead Texas Instruments Filter Pro design software [17] is used to find the component values for the chosen specifications. With this software, one of the filter types can be chosen with a particular cut-off frequency, and resistor and capacitor values can be tailored to a particular range.

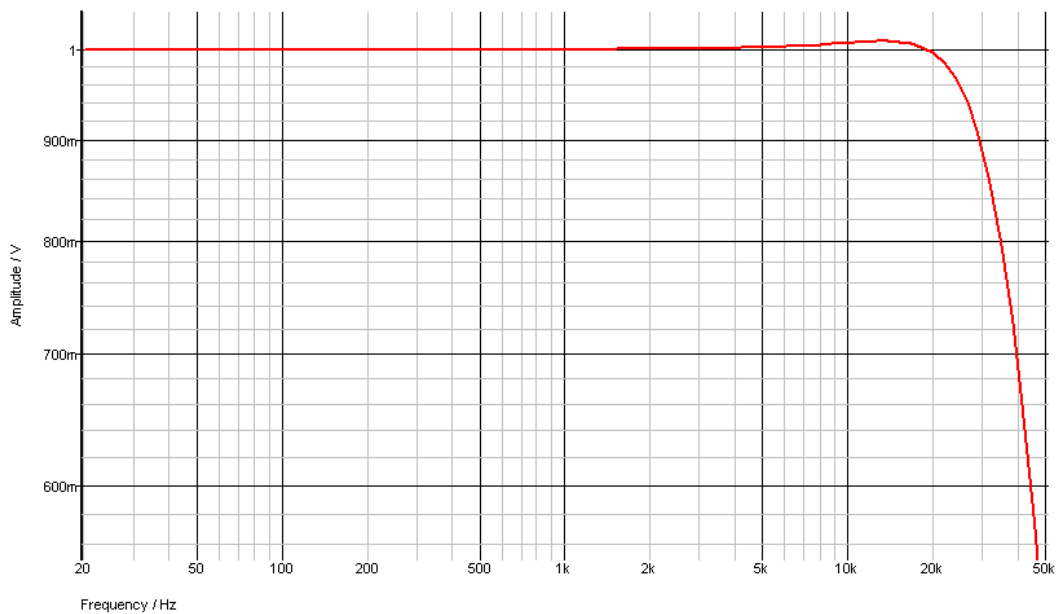


FIGURE 3.9: SIMULATED FREQUENCY RESPONSE

A Chebyshev filter is chosen in the Filter Pro software with a cut off frequency of 19.5 kHz and a pass band ripple of 0.1 dB to produce the theoretical frequency response shown in Figure 3.9. The curve has a peak of about 0.08 dB at 14 kHz, and is down 4.3 dB at 44.1 kHz which is enough to supplement the filtering that is already done in the DAC chip. The resulting values are those shown in the filter circuit in Figure 3.6. Also shown is a capacitor-resistor high pass filter on the op-amp output comprising of C3 and R4. The function of this filter is simply to

remove the DC offset as the signal is centred at 2.5 V, half way between the single supply voltage rails of 0 V and 5 V. Lastly, a resistor (R5) is added in series with the filter output which not only protects the op-amp from short circuits, but also improves the high frequency response. This is because at high frequencies, the two capacitors (C17 and C18) are effectively short circuits and the ability of the filter to attenuate high frequencies relies on the parasitic capacitance of the circuit and the op-amp's output impedance. A series resistance combined with the natural capacitance after the filter acts as an extra filter at high frequencies.

3.5.3 ANALOGUE POWER SUPPLY

The DAC is the only analogue section in the whole system, and it requires a clean power supply as free from digital noise as possible. It is not feasible to include a completely separate power supply for the analogue section due to board space and cost restrictions, so the analogue supply is taken from the existing 5 V power supply. However, for the analogue output section, the supply includes a separate L-C filter and extra smoothing capacitors, as shown in Figure 3.10, to minimise the power supply noise. The DAC chip has a separate L-C filter to further reduce noise from the digital section, and also to stop noise from the DAC itself polluting the power supply and eventually affecting the filter section. Additionally, the analogue output section is positioned close to the power supply where noise is minimal. The positioning of the analogue section is discussed in more detail in Section 3.13.

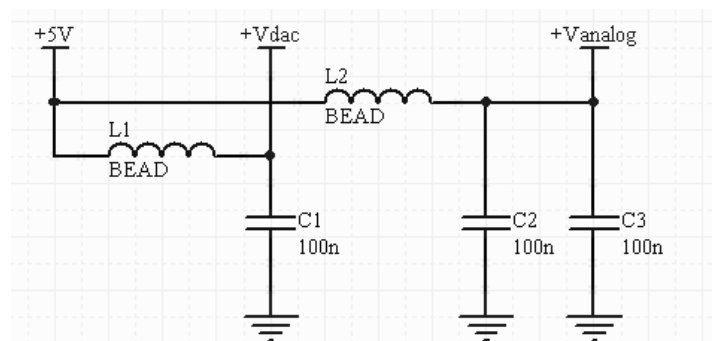


FIGURE 3.10: ANALOGUE POWER SUPPLY

3.6 INTER CIRCUIT DIGITAL AUDIO TRANSMISSION

3.6.1 SERIAL FORMATS

There are a number of methods for transmitting digital audio data around different sections of a circuit. The most common format used is the I²S [18] (Inter-IC sound) format invented by Philips for inter-circuit digital audio transmission, but both the DAC and SPDIF chips also accept Left and Right justified formats. The I²S format is shown in Figure 3.11 where SCK is the bit clock, WS is the left/right clock and SD is the serial data. The data are shifted out - most significant bit (MSB) first - with the WS edge being one clock period before the MSB is transmitted. The only difference between the I²S format and the Left and Right justified formats is the absence of this one bit delay with the MSB being transmitted on the immediate bit after a WS clock edge.

Regardless of what format is used, they all require the receiver to be set up for a SCK with a particular number of bits per WS period, where the frequency of the WS clock is the same as the sampling frequency f_s (44.1 kHz). Although the DAC chip accepts a SCK frequency of $32f_s$, $38f_s$ and $64f_s$, the SPDIF chip only accepts $64f_s$ and $128f_s$, therefore a SCK frequency of $64f_s$ must be used. Given that there are then 64 bits (clocked by the bit clock at f_s) per cycle of the WS clock, this provides 32 bits for each WS edge, and therefore 32 bits per transmitted sample.

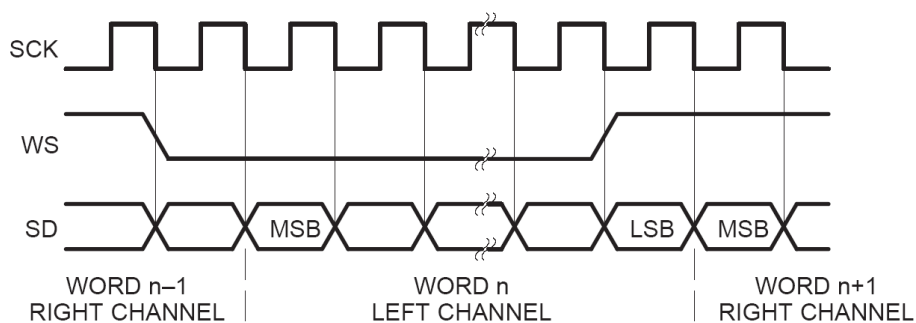


FIGURE 3.11: I2S FORMAT

3.6.2 PARALLEL TO SERIAL CONVERSION

The 16-bit parallel data extracted from the ATA interface must be converted to one of the above serial formats to be transmitted to the SPDIF and DAC circuits.

However, due to the limited processing power of the controller it is impossible to generate all the required signals in software while running at 8 MIPS. At this speed it is also impossible to generate the serial data stream and run the other tasks the controller must process, so a combination of hardware and software is used. The clock signals are generated in hardware and are discussed in Section 3.7, while the data signal is generated using two 74HC165 shift registers to create the required signals for the Left-justified format.

The DS signal (*SData*) is generated using the circuit shown in Figure 3.12 where the parallel data are loaded when the *SR_Load* line is taken low by the controller and is shifted out on the rising edges of the *BitClk_Inv* clock signal. This *BitClk_Inv* signal is required to be at the same frequency as the *BitClk* signal that latches the data into the DAC and SPDIF chips. It must also be sufficiently out of phase such that each bit is shifted through the shift registers before that bit is latched into the receiver. Ideally it would be the exact inverse of the *BitClk* signal.

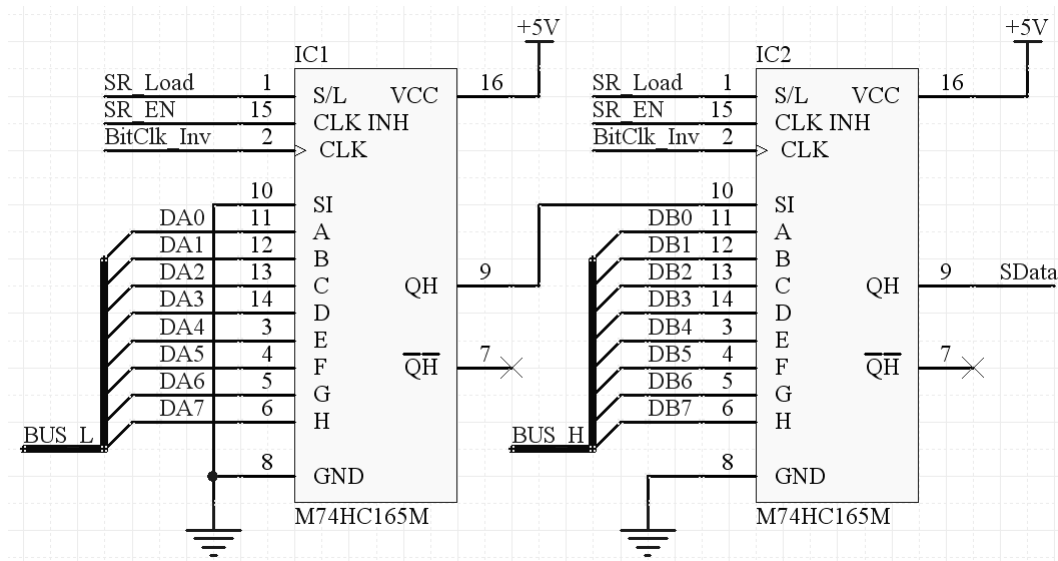


FIGURE 3.12: SERIAL DATA GENERATION CIRCUIT

A third control line, *SR_EN*, runs at twice the frequency of the *WS* clock signal and automatically disables the shift registers during the second half (inactive period) of the data transmission allowing the registers to be loaded with the next sample of data. This inactive period is shown in Figure 3.13, and it exists because for each *WS* clock period there are 64 bits transmitted (*SCK* is $64 f_s$) resulting in 32 bits for each sample, yet because each sample is only 16 bits long, there are

16 inactive bits transmitted in the second half of the period which must be all zeros. With this shift register configuration, when either all the data are shifted out of the registers (pin 10 of the first shift register is connected to GND) or when the shift registers are disabled the data out pin is tied to ground. The result of this is that the inactive bits are transmitted zeros as required. After getting the interrupt from the rising edge of the *SR_EN* line (midway through the data transmission) with this system configuration, the only requirement of the controller is to load the samples into the shift registers during the inactive period when the shift registers are disabled.

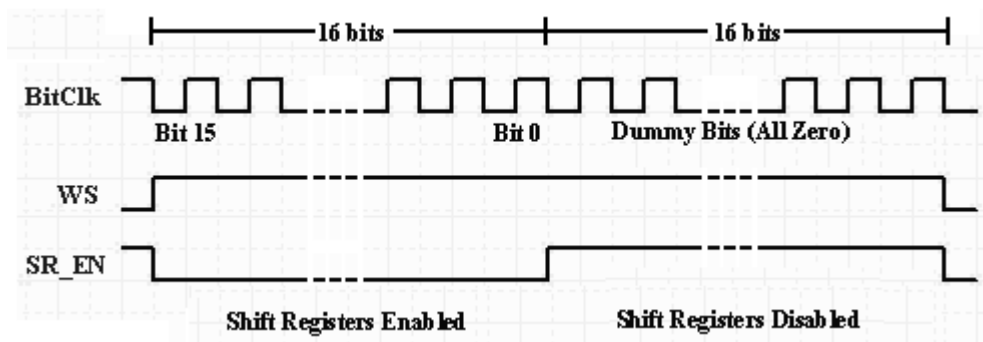


FIGURE 3.13: TIMING OF THE SHIFT REGISTER SIGNALS

3.7 CLOCK GENERATION

In total there are five different clock signals that need to be generated in the system. These are summarised in Table 3.2, with their frequency and multiple of the sampling frequency f_s . It is clear that each frequency is a power of 2, making it easy to generate each signal with simple dividers.

Signal Name:	Signal Description:	f_s Multiple:	Frequency:
MstrClk	Master Clock	$256 f_s$	11.2896 MHz
BitClk	Bit Clock	$64 f_s$	2.8224 MHz
BitClk_Inv	Bit Clock Inverse	$64 f_s$	2.8224 MHz
SR_EN	Shift Register Enable Interrupt	$2 f_s$	88.2 kHz
WS	Left/Right Clock	f_s	44.1 kHz

TABLE 3.2: SUMMARY OF REQUIRED CLOCK SIGNALS

Because each counter can divide by a maximum of 8, and a total division of 256 is needed, this can be achieved with just two counters. However, this only results in a single $64f_s$ output, where the requirement is for two out of phase signals at this frequency. One solution is to take the $32f_s$ frequency and divide it by 2 with a flip-flop giving complementary outputs. A simpler solution, as shown in Figure 3.14, is to use a third 74AC161 chip and configure the first as a divide by 4 counter. This chip counts through the sequence: 13, 14, 15, 0 and repeats, and the TC output is taken as the *BitClk_Inv* signal, with the Q3 output as the *BitClk* signal.

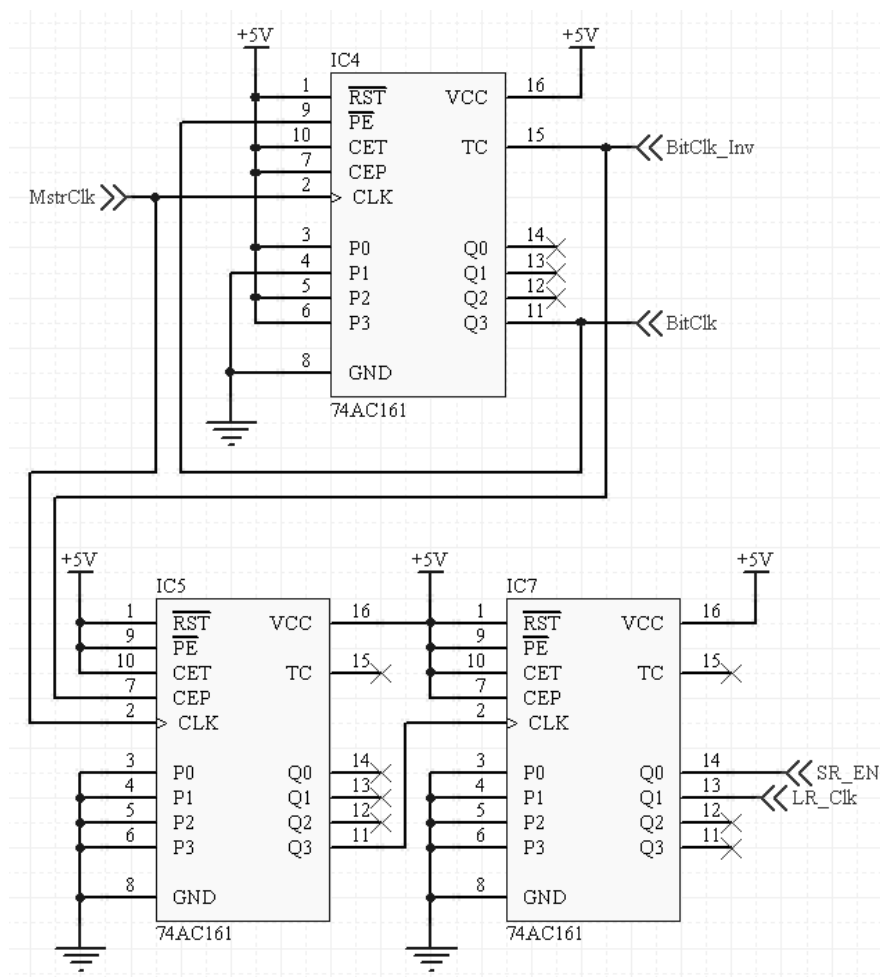


FIGURE 3.14: CLOCK GENERATION CIRCUIT

The advantage of this configuration is the offset of the falling edge of the *BitClk* signal with the rising edge of the *BitClk_Inv* signal as shown in Figure 3.15. Normally these edges would occur almost simultaneously, along with the falling edge of the *WS* and *SR_EN* signals, causing a possible yet unpredictable shift in the shift registers. This would occur if the *BitClk_Inv* rising edge occurs one setup

period after the falling edge of the SR_EN clock, where a “setup period” can possibly be as low as 6 ns due to slow rising edges and propagation delays (page 5 of the 74HC165 datasheet [19]). This phase relationship between the $BitClk$ and $BitClk_Inv$ signals created a very robust timing platform for the parallel to serial conversion circuitry to operate from.

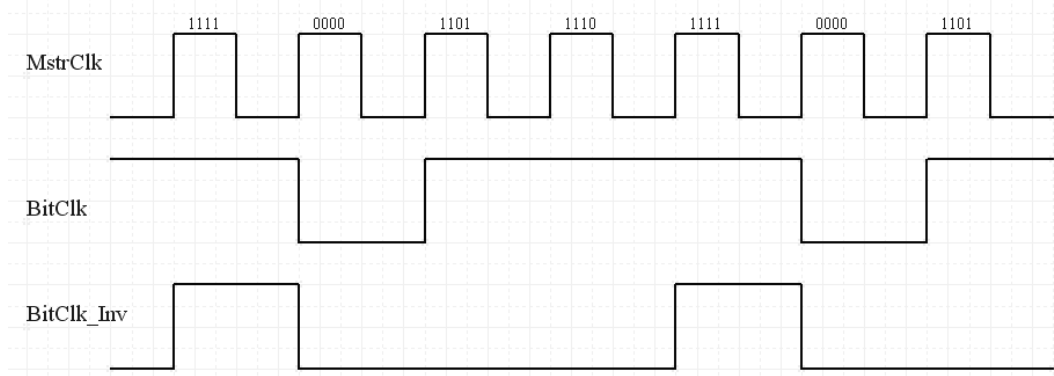


FIGURE 3.15: $BITCLK$ AND $BITCLK_INV$ PHASE RELATIONSHIP

The second 74AC161 counter chip is incremented synchronously with the first when the $BitClk_Inv$ signal is high, and further divides the signal by eight. Finally the third chip divides the signal by 2 and 4 to obtain the SR_EN and WS (LR_Clk in Figure 3.14) clock signals. Also, the SR_EN and WS clock edges occur at a similar time to the falling edge of the $BitClk$ and $BitClk_Inv$ signals as they normally would.

3.8 MASTER CLOCK

The system runs from a common clock source that drives the clock circuit described above and synchronises the SPDIF and DAC circuits to the rest of the clock signals. This clock source is called the Master Clock and must be running at 11.2896 MHz as it has already been chosen to run at $256 f_s$. Much effort was invested in the selection of a master clock to make sure jitter performance was attractive, and it became clear that better jitter performance was achieved with a single crystal oscillator than with the phase lock loop (PLL) circuit. Modern crystal oscillators can have jitter performance in the sub-pico second range making jitter almost nonexistent in the clock source. The chosen oscillator is the CFPS-72 Simple Packaged Crystal Oscillator (SPXO) from CMAC [20], which

has a CMOS tri-state output with rise and fall times of 6 ns. The jitter for this particular oscillator was not immediately available, so CMAC were contacted and the phase noise of a similar product (10 MHz CFPS-73) was provided as shown in Figure 3.16.

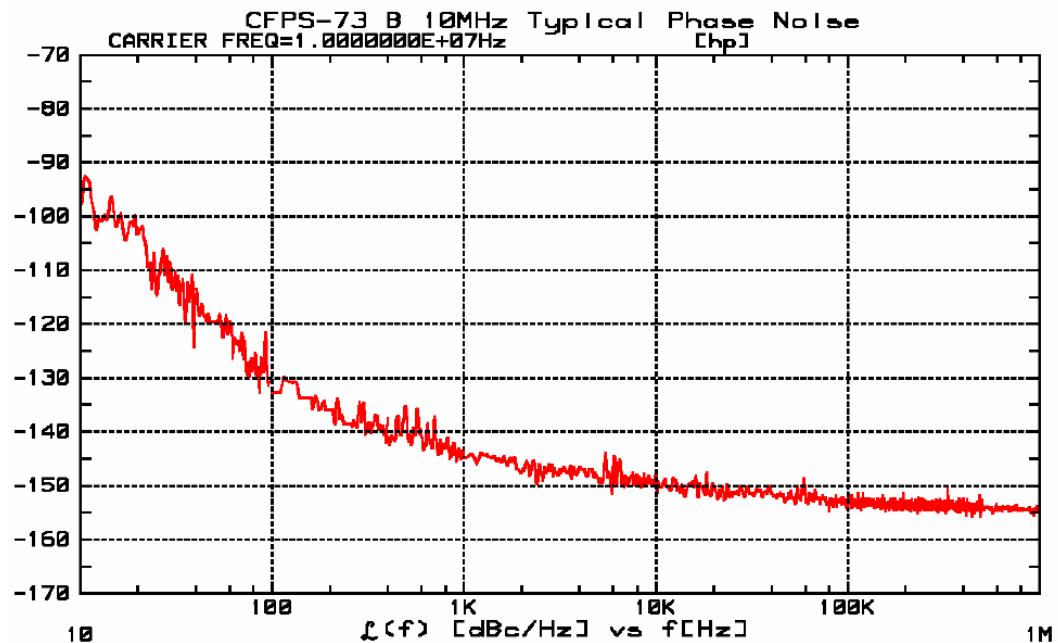


FIGURE 3.16: PHASE NOISE FOR A 10 MHz CFPS-73

The phase jitter can be calculated from the phase noise curve using a series of complicated calculations [21], however it is more practical to use an online calculator such as the one provided by Ralton [22]. Several points were taken from the phase noise curve which results in a phase jitter of around 0.7 ps (0.00287 degrees) for an 11.2896 MHz oscillator. This value gives an idea on the performance of the chosen oscillator (CFPS-72) and it is apparent that a typical jitter figure is less than 1 ps. This is insignificant compared to the jitter that would be inherent in the rest of the circuit, thus making a single crystal oscillator much more desirable than a PLL.

3.9 USER INTERFACE

The simple user interface, chosen by Perreaux, consists of a circular array of buttons with 4 around the circumference, and one in the centre. A sample was obtained, which is shown in Figure 3.17, and the final product will include printed

labels to indicate each button function such as Play/Pause (Centre), FF/Scan Forward (Right), RW/Scan Back (Left), Stop(Top) and Eject (Bottom).

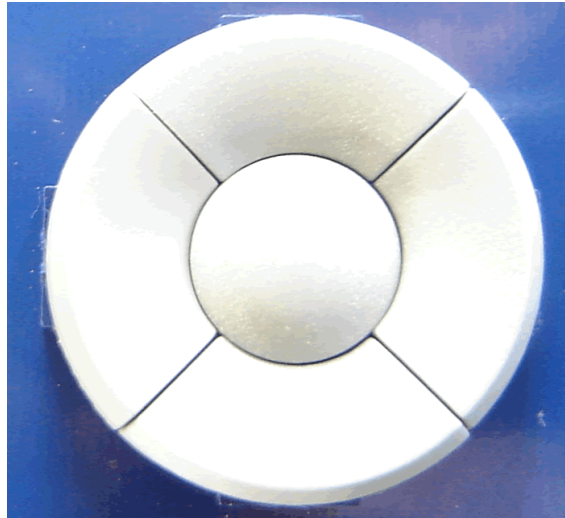


FIGURE 3.17: IMAGE OF USER INTERFACE BUTTONS

The button array is mounted on a separate circuit board, and connects to the main board through a 10-way ribbon cable, where one contact from each button connects to ground, and the other connects to one of five data lines that feed back to a 74HC373 latch chip as shown in Figure 3.18. Each button signal connects through an AND arrangement made from individual diodes and pull-up resistors that drives the “user interrupt” pin on the controller. With this arrangement the controller detects the interrupt when any button is pressed and can then read the low data bus to obtain the current state of the buttons.

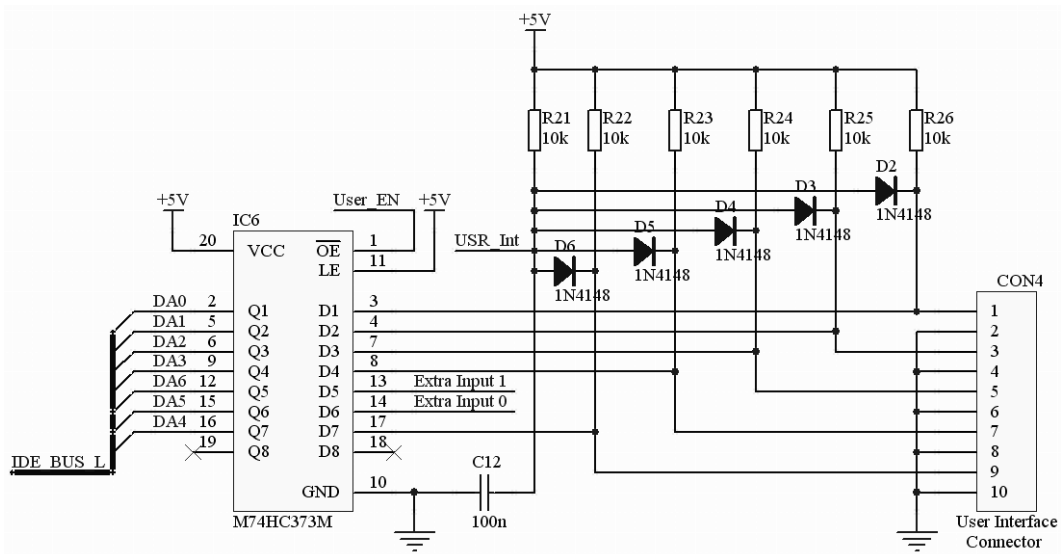


FIGURE 3.18: USER INTERFACE CIRCUIT

3.10 POWER SUPPLY

In a compact design such as the SXCD, a switch mode power supply would be a good solution. However, this is a high quality audio product that requires the cleanest possible power supply at the lowest feasible cost. A switch mode power supply typically generates a lot of high frequency noise compared to a transformer power supply, and due to the costs and large amount of time involved in switch mode power supply design it is not a feasible option. A simple transformer DC power supply is therefore implemented as shown in Figure 3.19 to provide the required 5 V and 12 V rails.

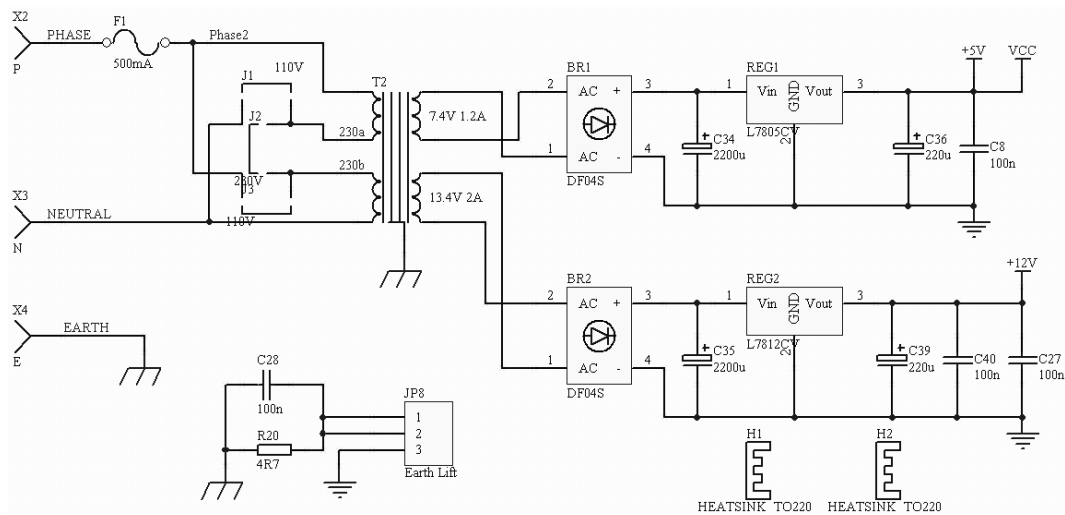


FIGURE 3.19: POWER SUPPLY CIRCUIT

3.10.1 CIRCUIT OVERVIEW

Firstly, basic over-current protection is provided with a 500 mA slow blow fuse in series with mains input, which then passes through three jumpers to provide the option of running from either 230 V or 110 V. These jumpers configure the dual primary windings of the transformer in either series or parallel to suit the mains voltage provided. The transformer is custom designed and built by Toroid International [23] according to the specifications required and includes two secondary windings to provide separate 5 V and 12 V supplies. Each of these supplies is derived by rectifying the AC signal, filtering with some initial capacitance, and regulating to the desired voltage with a regulator chip [24]. Extra localised smoothing capacitors are provided immediately after the regulator to

provide extra smoothing. The resulting 5 V supply is used to power both the CD drive and the systems circuitry, while the 12V supply is only required to power the CD drive. An earth lift switch is also provided to connect the circuit ground to the chassis earth via a 100 nF capacitor in parallel with a 4.7 Ω resistor. This helps to remove noise generated from ground loops and is standard on all Perreux products.

3.10.2 TRANSFORMER RATING

When calculating the transformer secondary winding voltages, all of the voltage drops are taken into account including the regulator drop out voltage (V_{REG}), two rectifier diode drops (V_D) and any voltage ripple (V_R) present on the supply before the regulator. Also the power supply is specified for the worst case scenario of a 90 V input voltage on a 110 V supply, equating to a drop to 82 % of the nominal voltage. This means that each of the transformer secondary winding voltages must be 1.22 times the calculated value. All of these factors have been included in Equation 3-4, which can be used to calculate winding voltages of 7.4 V for the 5 V rail, and 13.4 V for the 12 V rail, assuming a V_R of 0.5 V, a V_D of 0.6 V and a V_{REG} of 1.8 V.

$$V_{AC} = \frac{1.22(V_{DC} + V_{REG} + 2 \times V_D + V_R)}{\sqrt{2}} \quad \text{Equation 3-4}$$

$$VA = V_{S1} \times I_{S1} + V_{S2} \times I_{S2} \quad \text{Equation 3-5}$$

CD drive Model:	5 V Supply Current		12 V Supply Current	
	Average	Peak	Average	Peak
Samsung SH-152	0.4 A	0.6 A	0.4 A	1.5 A
A open CD-950E	0.4 A	0.5 A	0.2 A	0.7 A
Diamond Data 640A	0.4 A	0.5 A	0.3 A	1.2 A
Asus CRW-5232A	0.3 A	0.5 A	0.3 A	1.4 A

TABLE 3.3: CD DRIVE CURRENT REQUIREMENTS

The required currents for the two supplies have been estimated from several measurements of CD drive current consumption, and the addition of the relatively

small amount of current required for the circuitry. The current requirements of several CD drive models are shown in Table 3.3 where a maximum current of 850 mA for the 5 V rail and 1.4 A for the 12 V rail are estimated. These are both DC currents, which result in AC RMS values of 1.2 A and 2 A respectively. Equation 3-5 can then be used to calculate the total transformer AV rating, where the final transformer secondary windings of 7.4 V @ 1.2A for the 5 V rail and 13.4 V @ 2A for the 12 V rail, give a total rating of 35.7 VA.

3.10.3 VOLTAGE REGULATORS

The power dissipation of each regulator depends on the voltage difference across the regulator itself and the average current draw on the power supply. The power dissipation for each regulator is calculated using Equation 3-6, to be about 2.2 W for the 5 V regulator, and 2.0 W for the 12 V regulator, where $V_{IN} - V_{OUT}$ is the voltage across the regulator and I is the average current through the regulator.

$$P_D = (V_{IN} - V_{OUT}) \times I \quad \text{Equation 3-6}$$

$$T_j = T_A + P_D \times (R_{th,j-case} + R_{th,case-sink} + R_{th,sink-air}) \quad \text{Equation 3-7}$$

The maximum power dissipation for the regulators alone is limited by the junction to ambient thermal resistance (50 °C/W) resulting in a junction temperature of over 140 °C with a power dissipation of 2.2 W. For this reason, both regulators are mounted on a 25.4 mm high heatsink [25] with a thermal resistance of 13.4 °C/W. The junction temperature of the regulator is then calculated using Equation 3-7 where T_A is the ambient temperature around the heatsink and each R_{th} is the thermal resistance between the junction, case, heatsink and air. With a $R_{th,j-case}$ of 5 °C/W, $R_{th,case-sink}$ of 1.2 °C/W and a $R_{th,sink-air}$ of 13.4 °C/W, the typical junction temperature is around 73 °C, assuming an ambient temperature of 30 °C. This is satisfactory as it results in a heatsink temperature of 60 °C in natural convection. The two regulators and heatsinks mounted on the circuit board are shown in the photo in Figure 3.20.

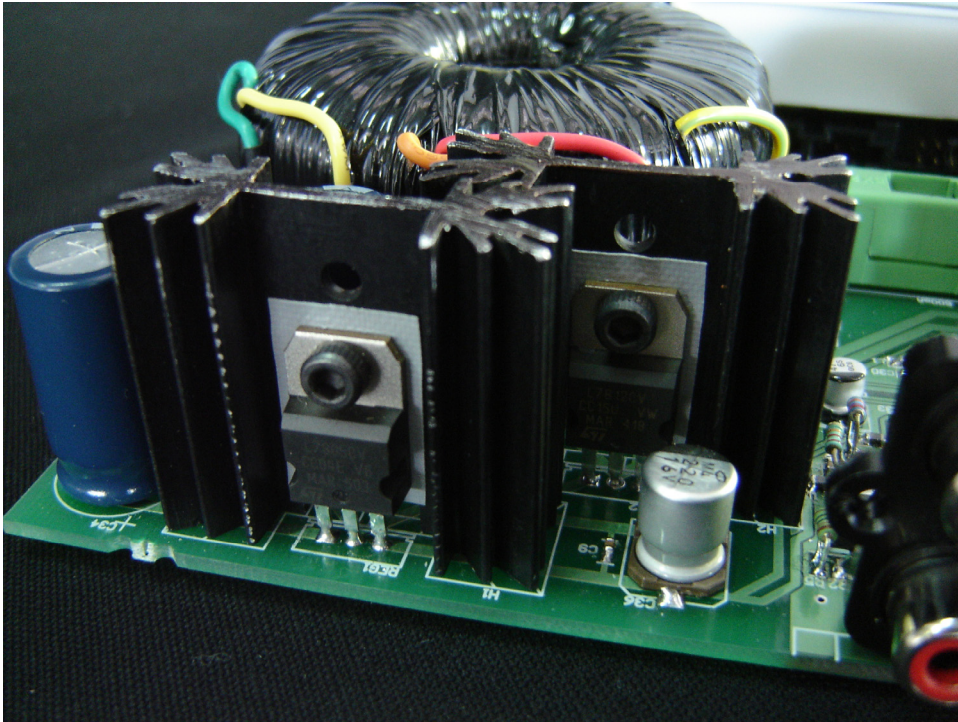


FIGURE 3.20: POWER SUPPLY REGULATOR HEATSINKS

3.10.4 FILTERING

To further improve power supply regulation and noise reduction at each sub-circuit, four localised L-C filters are included, as shown in Figure 3.21, and are located at the crystal oscillator, the SPDIF and DAC chips, and at the analogue output stage as discussed in Section 3.5.3. They not only help remove noise from the main power supply, but they also help prevent the noise that is generated in each sub-circuit from returning to the main power supply and effecting the operation of other sub-circuits.

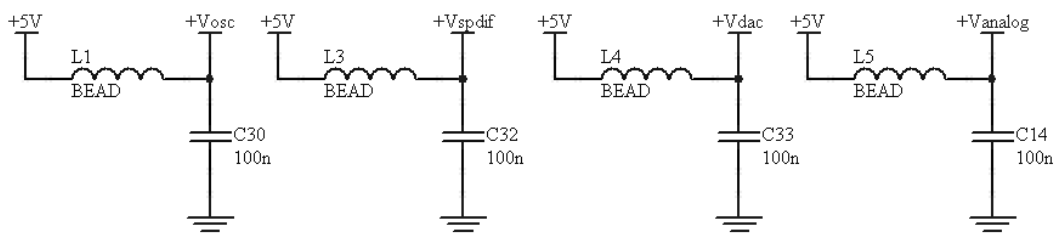


FIGURE 3.21: LOCAL L-C FILTERS

3.11 REMOTE INPUT

For compatibility and integration into existing systems, and also for future upgrades it was decided to include a remote input on the SXCD. This consists of a simple 3.5mm mono headphone socket connected to one of the controller pins through a simple filter circuit, and the necessary controller firmware to detect remote input signals and process them as user interface commands. There are many types of remote protocols, but they all use a modulated signal similar to the one shown in Figure 3.22 to increase noise immunity. The most common protocol is RC5 [26], originally invented by Philips, which uses a carrier frequency of 36 kHz, and Manchester encoding [27] for the data bits.

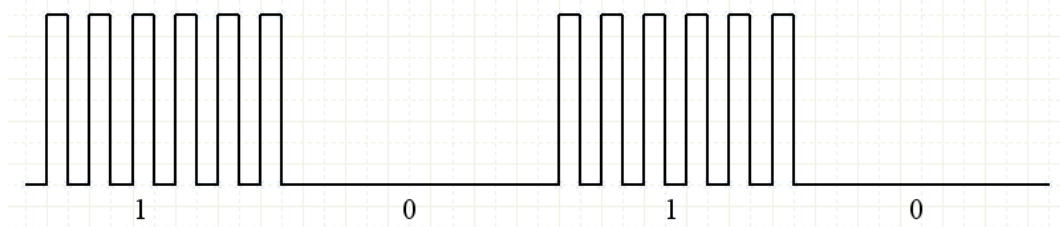


FIGURE 3.22: A SIMPLE MODULATED SIGNAL

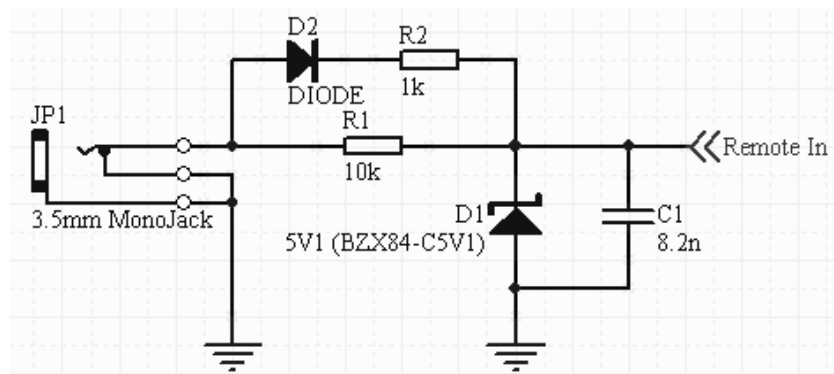


FIGURE 3.23: REMOTE INPUT CIRCUIT

As the data are modulated with a carrier frequency, decoding the raw signal (including the carrier frequency itself) in firmware would be very time consuming. To make decoding easier, the circuit shown in Figure 3.23 is used to filter out the carrier frequency and provide just the data to the controller. The circuit operation is as follows: The diode D1 is used to clamp any input voltages above 5 V, as many products put out a 12 V signal. Because at 12 V the current through D1 is only 6.4 mA, the diode will actually clamp at around 4.8 V, but this

is not a problem. The capacitor C1 charges through the parallel combination of R1, R2 and D2, but only discharges through R1. This acts as a filter and provides a higher average voltage at *Remote In* when a 5 V signal voltage is used. The resulting signal at *Remote In* is simulated in Figure 3.24, where the signals from both 5 V and 12 V inputs are shown. Both the signals are large enough as they pass above the controller high level threshold (V_{thH}) of 3 V (page 195 of the ATmega8515 datasheet).

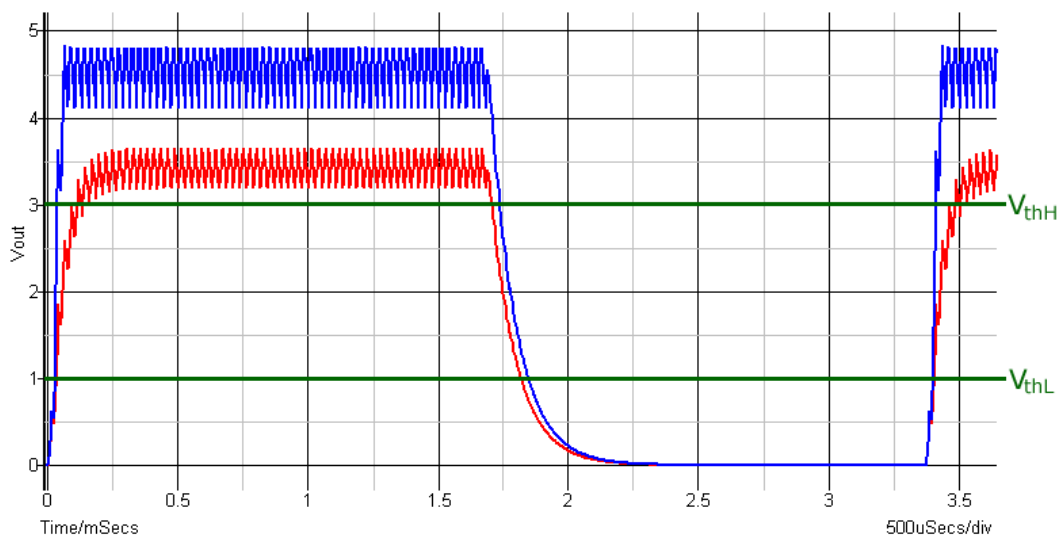


FIGURE 3.24: SIMULATED REMOTE IN FOR 5 V AND 12 V SIGNALS

3.12 RJ45 CONNECTOR

The RJ45 connector is a socket that has been added to the design to provide options for future upgrades and connectivity to other new products. Two features are implemented in the connector. The first is to provide a system upgrade path when mated with the SXD2 module. By running the SXCD as a slave to the SXD2 DAC, which can be implemented by receiving the master clock signal through this connector, timing errors inherent in the SPDIF transmission are eliminated. The internal crystal oscillator is enabled by connecting pin-7 to ground, and the master clock can be driven from pin 4 as shown in Figure 3.25.

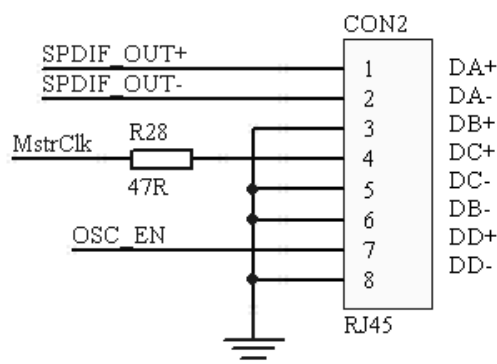


FIGURE 3.25: RJ45 CONNECTOR CIRCUIT

The second feature that is added to the RJ45 connector is to transmit the SPDIF signal out through a typical Ethernet cable that either connects to a remote DAC module, or several devices through an expander module. This provision is made by connecting the output from the SPDIF chip to a third twisted pair on the RJ45 connector. The fourth and final twisted pair is simply connected to ground as it is unused.

3.13 PCB LAYOUT

The sub-circuits described in sections 3.1 to 3.12 make up the complete CD player circuit that is laid out and routed onto a single circuit board (PCB). This PCB is a double sided board with 1 oz copper, measuring just 204 mm by 98 mm, and contains all electronic components apart from the CD drive itself. The circuit components are laid out in their corresponding sub-circuits, as shown in Figure 3.26, with the aim to minimise high frequency track lengths, power supply noise and board size.

Although the mains power is connected to the board at the right of Figure 3.26, the position of the power supply is limited to the left hand side due to the size of the transformer, which measures 65 mm in diameter. The two power rails connect to the CD drive power connector to supply power to the CD drive, and the 5 V rail supplies power to the rest of the circuit.

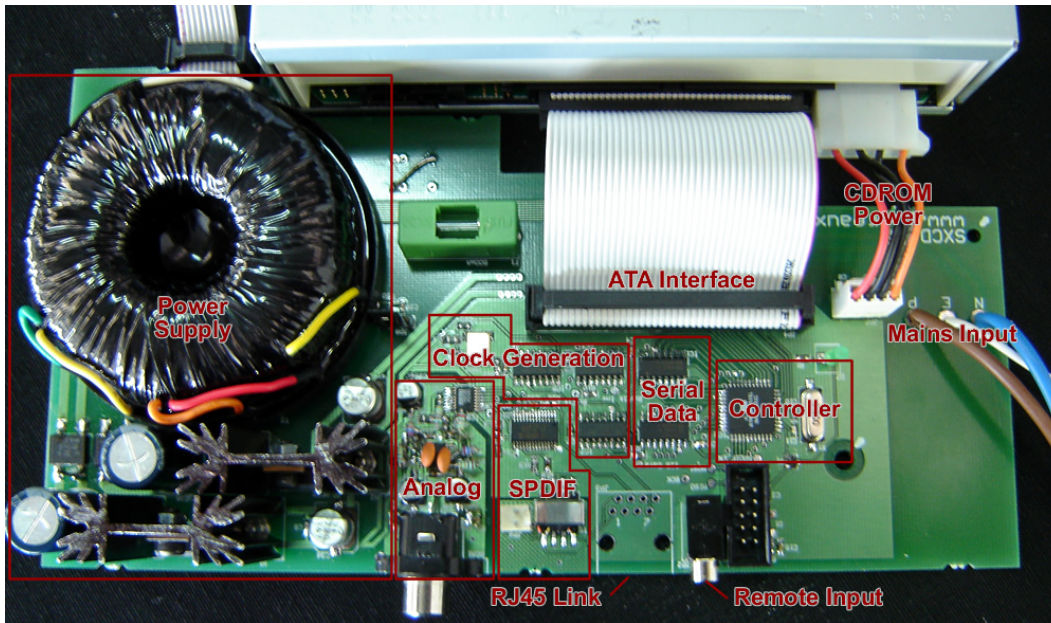


FIGURE 3.26: PCB CIRCUIT OVERLAY

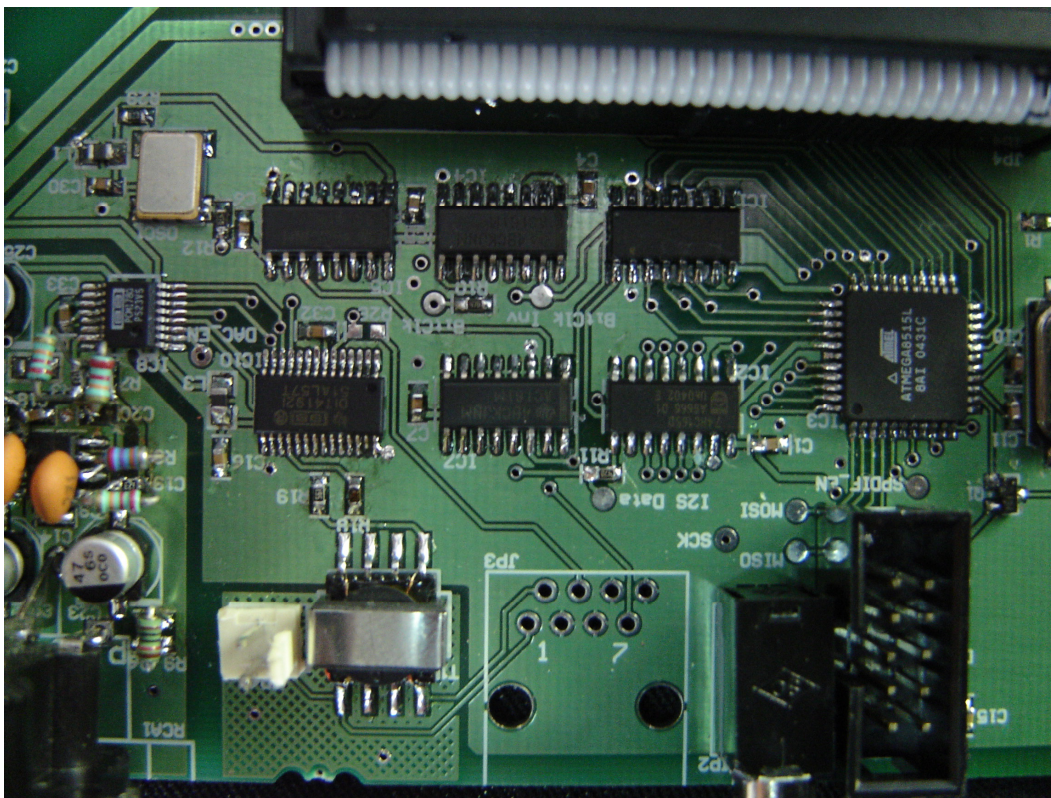


FIGURE 3.27: MAIN SUB-CIRCUITS ON THE PCB

The analogue output sub-circuit is positioned close to the power supply to ensure it gains the cleanest possible supply. The controller is positioned close to the ATA interface, and also close to the serial data sub-circuit, to minimise the track lengths for the data bus. The crystal oscillator is positioned such that the master

clock track lengths to the analogue and SPDIF outputs, and the clock generation sub-circuits are all minimised.

Basic guidelines are followed with the routing of the PCB traces such as avoiding long parallel tracks between different signals, physically separating noisy and sensitive signals and using star points for the power supply tracks. Because the difference between a 90 degree, 45 degree and a rounded corner in a track is not significant unless dealing with signals in the upper Gigahertz frequency range, or using edges faster than 15 ps [28], all tracks are routed using 45 degree corners simply to maintain a tidy look. The tracks carrying the high current power supply to and from the CD drive run parallel to each other. This is done so the returning track, and thus the returning current, has a cancelling effect on the EMI radiated from the power supply. Also a small resistance of 47 Ω is added to the master clock line which helps prevent noise from coupling into the analogue output section.

3.14 INTERCONNECTS

The interconnects within the design include a 40-way ribbon cable for the ATA interface, a 4-way power connector connecting from the CD drive to the PCB, a 10-way ribbon cable connecting the PCB to the user interface on the front panel and a 2-way loom connecting the front panel LED to the PCB. In production these four cables will be manufactured by Munro Cable Fabrication [29].

3.15 DIMENSIONS

It was desirable that the SXCD product would use the same dimensions of the existing Silhouette series products of 210 mm wide, 45 mm high and 150 mm deep, yet it was even more desirable to use a computer CD drive as the optical mechanism. This resulted in the use of a larger chassis, but because the same front panel profile must be maintained, a chassis of exactly twice the depth (300 mm) is used. Because the CD drive is a maximum of 198 mm long, this allows a maximum depth of 100 mm for the circuit board.

3.16 FRONT PANEL

It has already been noted that the front panel for the SXCD must be consistent with the existing Silhouette range of products. This not only includes keeping the same width and height, but also maintaining the location of the Perreux name and blue LED for power indication. A door that allows the CD drive tray to open and close and the mounting for the user interface are also part of the front panel design. However, the front panel design is not an immediate part of this project and is handled externally to Perreux.

4 FIRMWARE DESCRIPTION

4.1 OVERVIEW

The aim of the controller is to monitor the user interface and remote input, interface to the CD drive, and extract the digital audio data from the CD drive and pass it to the SPDIF and DAC circuits in a serial format (during audio playback). As described in Section 3.6, a hardware circuit exists to convert the parallel 16-bit data to the serial Left-justified format, and the controller is to load the data into the shift registers on the high cycle of the *SR_EN* clock line. A buffer for the digital audio data is required between the CD drive interface and the shift registers to allow for any delays caused by the CD drive. This is handled in the controller using the internal memory.

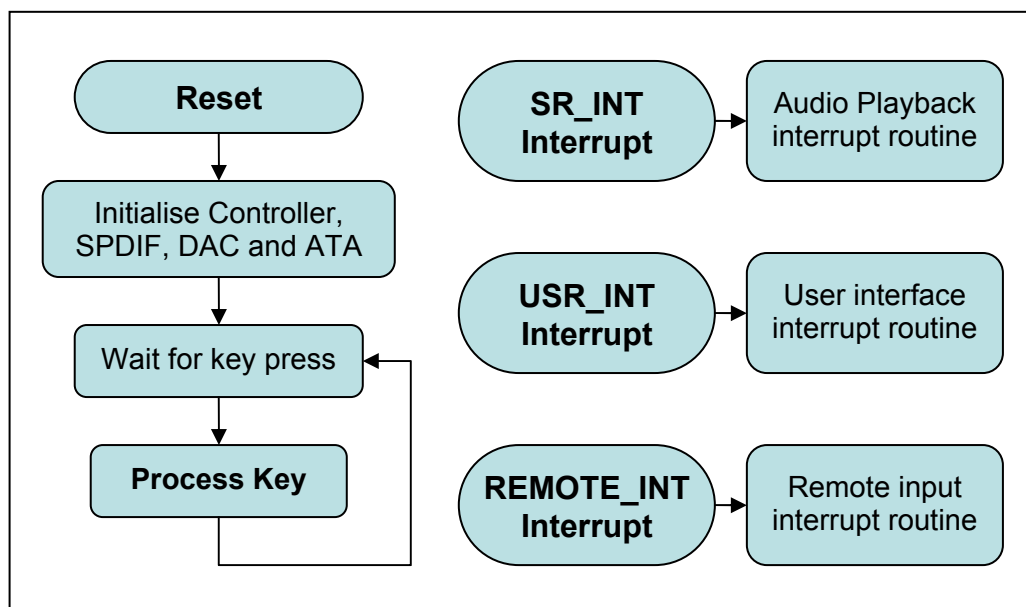


FIGURE 4.1: FIRMWARE OVERVIEW STATE DIAGRAM

Most of the code is written in the ANSI C programming language [30], but some is optimised using the assembly language [31]. The code follows an event-driven programming structure [32], where the code is executed when an external event occurs. In this configuration, there is very little code in the main loop, with the majority of the code being executed from separate interrupt routines as shown in Figure 4.1. The main loop starts at a power-on reset, and immediately initialises

the controller by setting all the pin directions and peripheral registers and resetting all global variables to their default values. The SPDIF, DAC and CD drive are then initialised, and the controller waits for a key command to be received. Also shown in Figure 4.1 are the interrupt routines for the three external events which drive the firmware: *SR_INT* is executed on the rising edge of the *SR_EN* clock line, *USR_INT* is executed when one of the user interface buttons is pressed, and *REMOTE_INT* is executed when a remote signal is received.

4.1.1 RESTRICTIONS

Two main restrictions exist regarding the firmware for the project: the first is the code size, which is limited to 8 kB of Flash memory in the ATmega8515. The final code size is 3733 words, or 7466 bytes, which equates to 91.1 % of the code space available. The total code consists of over 3200 lines of code (including debugging code) and is optimised for size in the compiler in order to reduce the code size below the 8 kB limit.

The second is the speed of the micro processor, where although the ATmega8515 is limited to 16 MIPS, the currently stocked part (the ATmega8515L) is limited to 8 MIPS. If all the data processing were to be completed in firmware this would be of concern, but because the data conversion is entirely done in hardware it is possible to perform all the functions from the 8 MHz clock. This allows the ATmega8515L to be used in the project reducing the number of new components needed. The timing restrictions are discussed in more detail in Section 4.6.3.

4.1.2 INTERRUPT STRUCTURE

As discussed in the sections above, there are three external interrupts that correspond to the *SR_EN* interrupt for audio playback (INT0), the user interface interrupt (INT1), and the remote input interrupt (INT2). All the interrupts are shown in Table 4.1 and each has a fixed priority structure, where the highest priority interrupt is at the top of the table, and the lowest at the bottom. The three external interrupts are carefully assigned to each of the three available functions based on importance of real-time operation. The result is that the *SR_EN* interrupt, being the highest priority, can interrupt any other interrupt, and therefore audio

playback is not effected by other interrupt routines operating in the background. Also, key presses and remote input commands can be processed simultaneously with audio playback.

Vector No.	Program Address	Source	Interrupt Definition
1	000h	RESET	Pin, Power-on, Brown-out & Watchdog
2	001h	INT0	External Interrupt Request 0
3	002h	INT1	External Interrupt Request 1
4	003h	TIMER1 CAPT	Timer/Counter1 Capture Event
5	004h	TIMER1	Timer/Counter1 Compare Match A
6	005h	TIMER1	Timer/Counter1 Compare Match B
7	006h	TIMER1 OVF	Timer/Counter1 Overflow
8	007h	TIMER0 OVF	Timer/Counter0 Overflow
9	008h	SPI, STC	Serial Transfer Complete
10	009h	USART, RCX	USART, Rx Complete
11	00Ah	USART, UDRE	USART Data Register Empty
12	00Bh	USART, TXC	USART, Tx Complete
13	00Ch	ANA_COMP	Analog Comparator
14	00Dh	INT2	External Interrupt Request 2
15	00Eh	TIMER0 COMP	Timer/Counter0 Compare Match
16	00Fh	EE_ROY	EEPROM Ready
17	010h	SPM_RDY	Store Program Memory Ready

TABLE 4.1: LIST OF INTERRUPTS

To enable any of these interrupts, both their individual interrupt enable flags and the global interrupt enable (GIE) bit must be set. The GIE bit is a flag the controller's status register that can enable or disable all interrupts. The GIE bit is automatically cleared by hardware when an interrupt occurs and is set again when an interrupt routine is exited.

4.1.3 NESTED INTERRUPTS

When an interrupt occurs, global interrupts are disabled automatically and the interrupt routine is executed. Once in the interrupt routine, if a higher priority interrupt occurs after a lower priority one, it can only interrupt the current routine if global interrupts have been re-enabled. Until global interrupts are enabled, this higher priority interrupt will not be executed. This results in a period of time

(interrupt dead time), between when the first interrupt occurs and global interrupts are re-enabled, where the higher priority interrupt will be delayed. This applies to the *SR_EN* interrupt, and so this time period must be minimised so audio playback is not affected. This is discussed in more detail in Section 4.6.3.

4.1.4 CODE DEFINITIONS

The firmware makes use of macro definitions, where a value, a register, or even a complex expression is replaced by an easy to remember word or symbol. The definitions include all pin locations, many commonly used values (such as timing and status values), register addresses and status bits, and even small enable and disable macros containing one or two instructions. Some definitions are defined twice, for use in both the C code and assembly language.

Two structures are also defined for storing MSF addresses using the type definition (*typedef*) keyword. The first structure, called “*TrackInfoStruct*”, stores the track address information and contains three signed character fields: one for each of the Minute, Second and Frame (MSF) fields in the address. The second structure, called “*TimeStruct*”, stores any current position and is used for keeping track of the current audio position being played. This structure contains three unsigned character fields for the M, S, and F fields in the address, and another unsigned character field for the current track number. These structures are defined as a type as they are passed as function parameters throughout the firmware.

4.2 ATAPI INTERFACE

As described in Section 3.2, the ATA interface consists of a 16-bit data bus, 5 address lines, a read line, and a write line. The ATA protocol is based around a set of nine command block registers where commands are sent by loading these registers with the desired configuration values. All data transfers are 16-bits wide although register access only uses the lower 8-bits of the data bus. The ATAPI protocol uses the same ATA hardware interface, but instead of using simple register delivered commands the ATAPI device receives its commands through the use of a Packet mode in addition to the normal ATA protocol.

4.2.1 REGISTER ACCESS

Interfacing the standard ATA interface involves reading and writing a set of command block registers. There are nine register locations in total, as shown in Table 4.2, which are addressed by the five-bit address bus (CS1:0 combined with A2:0) and accessed using the lower 8 bits of the data bus. Some of the registers have different functions for reading and writing: for example; reading address location 0x17 would read the status register, while writing to this location would write to the command register. ATA commands, called “register delivered commands”, are issued by loading the required registers with the required parameters, then writing the command code to the Command Register.

CS1	CS0	A2	A1	A0	Value	ATA Label	ATAPI Label
A	N	A	A	N	0x0E	Device/Status	Device/Status
N	A	N	N	N	0x10	Data	Data
N	A	N	N	A	0x11	Features/Error	Features/Error
N	A	N	A	N	0x12	Sector Count	Sector Count
N	A	N	A	A	0x13	Sector Number	LBA Low
N	A	A	N	N	0x14	Cylinder Low	LBA Mid
N	A	A	N	A	0x15	Cylinder High	LBA High
N	A	A	A	N	0x16	Device/Head	Device
N	A	A	A	A	0x17	Command/Status	Command/Status

TABLE 4.2: COMMAND BLOCK REGISTER LIST

The reading and writing of the registers is achieved using the two routines shown in Figure 4.2, where the command address is put on the address bus using the *SetAddr* function, and then the data are either written by toggling the *ATA_WR* line, or read by toggling the *ATA_RD* line. These two functions form the basis of all transfers over the ATAPI interface.

The drive status can be read from the status register at any time, consisting of eight flags, three of which are used by the controller. These are a busy flag (BSY) set whenever the drive has access to the command block, a drive ready flag (DRDY) set when the drive is capable of accepting a command, and a check flag (CHECK) set when an error occurred. The firmware waits for a particular flag in

the status register to be set or cleared when one of the functions in Figure 4.3 are called.

```

    /* Writes a byte to the specified ATA register */
    void RegWrite (unsigned char reg, unsigned char data)
    {
        SetAddr (reg);
        DATA_LOW_DIR = 0xFF;    // Set data port as outputs
        DATA_LOW = data;
        ATA_WR = 0;              // Enable
        ATA_WR = 1;              // Disable
        DATA_LOW_DIR = 0x00;    // Set data ports as inputs
    }

    /* Reads the specified ATA register */
    unsigned char RegRead (unsigned char reg)
    {
        unsigned char data;
        SetAddr (reg);
        // Data bus ports are already inputs.
        ATA_RD = 0;              // Enable
        #asm ("nop");            // Essential on 8 MHz !
        data = DATA_LOW_IN;
        ATA_RD = 1;              // Disable
        return data;
    }

```

FIGURE 4.2: REGISTER READ AND REGISTER WRITE FUNCTIONS

```

    /* Wait for the bit in the status register to be cleared */
    void WaitForFalseStatusBit (char Bit) {
        int timeout = 0;
        while (RegRead(STATUS) & (Bit)) {
            timeout ++;
            delay_ms (1);
            if (timeout >= ATA_TIMEOUT) {
                #asm ("JMP __RESET");
            }
        }
    }

    /* Wait for the bit in the status register to be set */
    void WaitForTrueStatusBit (char Bit) {
        int timeout = 0;
        while (!(RegRead(STATUS)&(Bit))) {
            timeout ++;
            delay_ms (1);
            if (timeout >= ATA_TIMEOUT) {
                #asm ("JMP __RESET");
            }
        }
    }

```

FIGURE 4.3: ROUTINES TO WAIT FOR A GIVEN STATUS BIT

```
/* Reads one word from the data bus. Make sure the
   bus is set up prior to calling this function. */
void ReadWord (char *LO_byte, char *HI_byte)
{
    ATA_RD = 0;           // Enable
    #asm ("nop");        // Essential on 8 MHz !
    *LO_byte = DATA_LOW_IN;
    *HI_byte = DATA_HIGH_IN;
    ATA_RD = 1;         // Disable
}
```

FIGURE 4.4: READ WORD FUNCTION

Data transfer takes place by reading or writing the data register at any time, although the value in the data register is only valid when the DRQ flag in the status register is set. The data register is the only register that is 16-bits wide, where the lower byte precedes the upper byte in the data stream on the disc. Because data are not written to the data register at any time from the controller, only the *ReadWord* function in Figure 4.4 is needed, where one word is read into two separate bytes. Once a word is read from the data register, the next word in the data stream is automatically loaded into the register ready to be read.

The LBA Mid and LBA High registers are permanently set to 0xFE and 0xFF respectively as these two are the two byte count registers. The values in these registers hold the maximum number of bytes to be transferred at any time, and are set to 0xFFFFE as this is the maximum allowed. Both the LBA Low and Features registers are not used by the controller and are set permanently to 0x00. The Sector Count register is also unused, but is not set as it does not affect the interface.

4.2.2 PACKET COMMANDS

For ATAPI devices, such as a CD drive, commands are delivered using packet commands, where the command packet contains the command and parameters that the device is to execute. The packet is transferred using Programmed Input Output (PIO) transfer, where the data are latched at a rate determined by the controller. To send a packet command, first a PACKET register command is sent by setting the command registers as described above, and then the command packet is transferred to the device (page 274 of [33]). This process is handled by

the function shown in Figure 4.5, which follows the ATAPI specification for packet command transfer. The *WriteCMDData* function sends the packet command, and once the drive is ready, data transfer may proceed if required.

```

/* SendPacketCommand sends the ATAPI packet command
   according to the ATAPI specification. */
unsigned char SendPacketCommand (void)
{
    // Wait for BSY = DRQ = 0
    WaitForFalseStatusBit (SR_BSY | SR_DRQ);

    // Setup Command Registers:
    RegWrite (BYTE_COUNT_LOW, 0xFE); // Limit does not seem to work
    RegWrite (BYTE_COUNT_HIGH, 0xFF); // so set it to maximum.
    RegWrite (COMMAND, 0xA0); // Packet Command

    // Wait for DRQ = 1
    WaitForTrueStatusBit (SR_DRQ);

    if (RegRead (STATUS) & SR_ERR) // Check for error
        return FAILURE;

    WriteCMDData (); // Write command data

    // Wait for BSY = 0
    WaitForFalseStatusBit (SR_BSY);

    // Return status register
    return RegRead (STATUS);
}

```

FIGURE 4.5: SENDPACKETCOMMAND FUNCTION

Command List	Operation Code	Usage
Read TOC	43h	Read the TOC data
Read CD MSF	B9h	Read CD-DA data
Start/Stop Unit	1Bh	Eject/Load tray
Request Sense	03h	Obtain error information
Test Unit Ready	00h	Test if the drive is ready and clear errors
Mode Sense	5Ah	Obtain media status
CD Speed	BBh	Set maximum CD spin speed

TABLE 4.3: PACKET COMMANDS USED BY THE CONTROLLER

At least 25 commands exist in the ATAPI packet command set, yet only seven of these commands are required for the operation of the system. The commands that are used are summarised in Table 4.3, along with their operation code and usage.

As with several of the available commands, the implementation of the CD Speed command is optional for drive manufacturers. However, because it is vital that the drive speed is limited to limit the noise in this product, a CD drive that supports this command must be used. In practice, simple CDROM drives do not support this command as it is not required, but any drive that is capable of writing to media does support the command. This is because the drive must be able to control the write speed to match the speed of the media being written to. Therefore as long as a CD writer is used instead of a CDROM drive, the CD Speed command will work and the drive speed (and thus noise) can be limited to a satisfactory level.

4.3 SYSTEM CONFIGURATION

When power is first applied to the system, the controller must initialise itself, the DAC chip, SPDIF chip, and the CD drive before the system is ready to operate. Initializing the ATmega8515 controller requires setting variables and flags to their default values, configuring the peripheral registers, and setting the pin register directions. Here Timer 0 is configured for a frequency of 31.25 kHz running from the system clock. Timer 1 is set to count on the T1 pin (Pin 3) falling edge, and the Compare interrupt is enabled with a value of 1175 (as explained in Section 4.6.5). Both timers are initially disabled until required. The external interrupts are configured with INT1 triggering on a falling edge and the other two (INT0 and INT2) on a rising edge, and the SPI module is configured for a 62.5 kHz clock frequency, most significant bit first, and a low clock polarity.

The DAC and SPDIF chips are both connected to the controller through a SPI serial communications bus to allow the controller to configure the internal registers of these chips. Most of the default values for the two chips result in the correct operation for the system. The only configuration required for the DAC chip is in register number 22, where the DREV bit is set. This inverts the phase of the output signal, which is required to maintain the correct phase due to the inverting nature of the output analogue stage (as discussed in Section 3.5.2). However, the registers in the SPDIF chip require much more modification. Firstly the chip is reset, and the chip is set to accept a bit clock of $64 f_s$, a master clock

of $256 f_s$ and a 16-bit Left-justified data format. The chip is powered up and the channel status data registers are configured for professional mode specifying a sampling frequency of 44.1 kHz, two channel mode, 16-bit word length, and an origin of “SXCD” and destination of “USER” in accordance with the AES3 specification [9].

Setting up the CD drive involves a sequence of steps, which are followed in a specific order according to the ATAPI specification. Following a power-on reset, the controller waits for 1 second before “Device 0” is selected in the device register (as the CD drive is configured as device 0 in hardware). Next the CD drive is issued with a software reset command, and the controller waits for 50 ms before disabling the ATA interrupt and clearing the features register (as this register is never used). Once the busy bit is clear, the interface is ready to process commands.

The drive is then polled with a Mode Sense command until the drive is ready and stops returning the error “NOT READY TO READY TRANSITION”. Once the drive is ready, the drive is set to the lowest accepted drive speed using the CD Speed command. Here the command sends a series of speeds starting with $1\times$, and moving up to $16\times$ until the returned speed matches the speed sent. Each value is a multiple of the audio playback bit rate, where $1\times$ is equal to 176.4 kbps. This procedure is to maximise the chance that the drive accepts the command (as some drives do not accept speeds as low as $1\times$ or $2\times$).

4.4 KEY PROCESSING

When either a key is pressed or a valid command is received on the remote input, that key value is pushed onto a key buffer (refer Section 4.7) and it is processed so the required action can be taken. The main firmware routine waits until the key buffer contains one or more key values by polling the *num_keys* variable, which contains the number of keys in the key buffer. Once a key is available, it retrieves the key value using the *pop_key* function (refer to Figure 4.16), where *num_keys* is decremented and the leading key is returned. Once the key value is received, the main firmware routine processes the key.

4.4.1 PLAY KEY

The process for a play key depends on many factors such as the current audio status, the state of the table of contents, and the status of the CD drive. The flow of the firmware when processing a play key is shown in Figure 4.6. When a play key value is received, the *audio_status* flag is checked, which holds whether audio is currently stopped or playing. If audio is currently playing, the audio status is simply changed to indicate audio is stopped. No other action is required because when the key is received, *go_flag* is cleared which halts audio playback, and as the current audio position is up-to-date, the audio is already in a paused state. If audio is currently stopped, the key press is indicating that the audio is to start playing. However before audio playback can take place, a valid table of contents (TOC) needs to be obtained, and any other key presses need to be processed first.

Firstly, the *TOC_invalid* flag is checked, which is set to logic one if the TOC currently stored in the *TrackInfo* structure array is invalid. The *TOC_invalid* flag is set at startup and whenever an eject command is processed (indicating that the disc has been changed). If the TOC is invalid, the CD drive is polled until the drive is ready, and then the current disc status is checked using the Mode Sense command and the media type field. If the disc is bad, indicating it is of an unknown format or simply unreadable, the tray is ejected for the user to change the disc. If no disc is present, the command is ignored, and if the disc is ready, the TOC is read from the disc.

Once a valid TOC is obtained, the key buffer is checked once again, and if another key is pending, the play operation is aborted and the pending keys are processed. Otherwise, the emphasis status is updated by calling the *SXCD_update_emphasis* function, and then the *SXCD_play* function is called to begin audio playback.

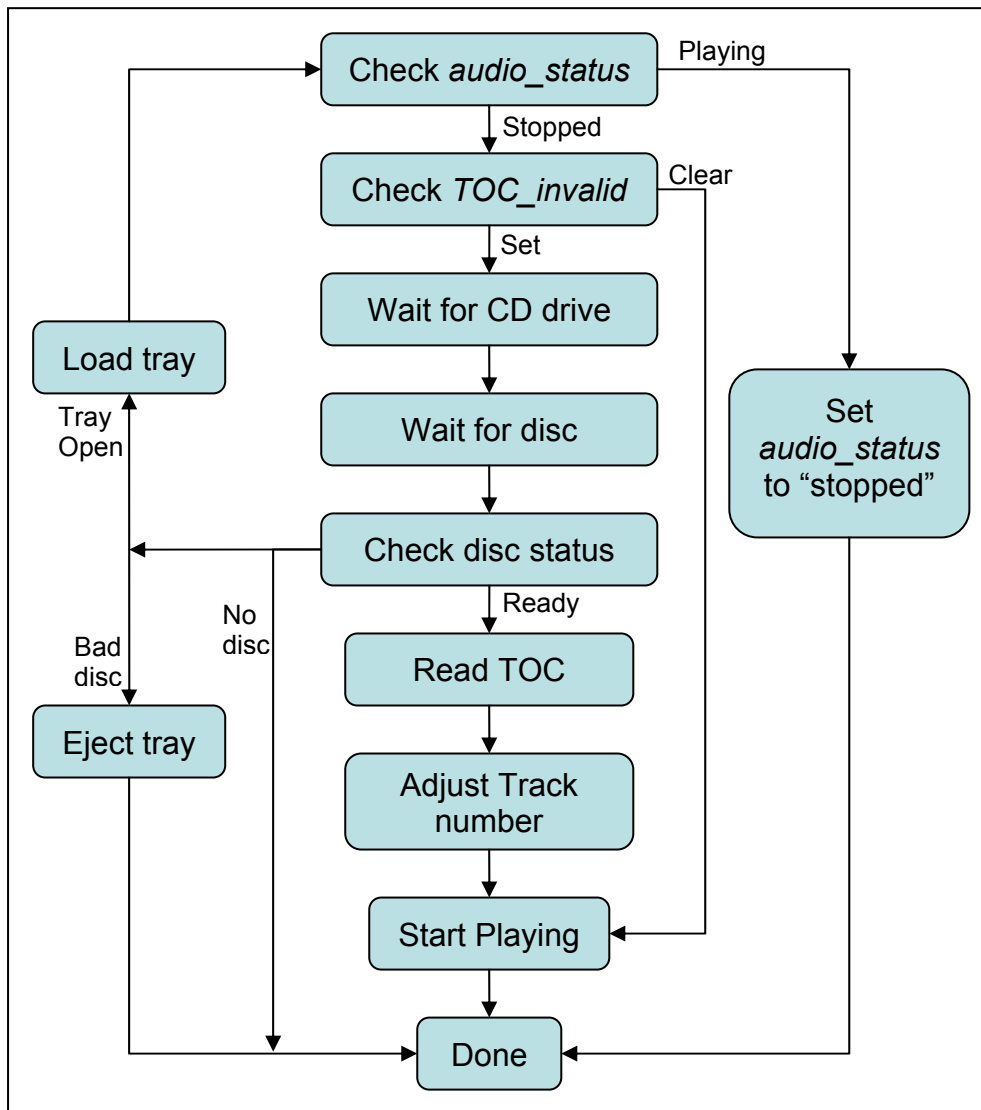


FIGURE 4.6: PLAY KEY PROCESSING FLOW DIAGRAM

4.4.2 STOP KEY

Processing the stop key is similar to processing a play key when audio is currently playing. It involves updating the audio status to indicate that audio playback has stopped. However, when the stop key is pressed, the next play operation should start from the first track on the disc, and so the current position is also reset to the start of the first track.

4.4.3 SKIP KEYS

When either a skip forward key (FF) or skip backwards key (RW) is received, the current track is to be either incremented or decremented respectively. This occurs

regardless of whether audio playback is in progress or not, or if a valid TOC is currently held, but depending on the current status, the functionality differs slightly.

When the TOC is invalid, the current track number is incremented on a FF key or decremented on a RW key (as long as it is greater than zero). If the TOC is valid and audio is stopped, the track number is only incremented on a FF key if the current track is less than the last track. When audio playback is in progress, a play command is pushed onto the key buffer to start audio playback at the next or previous track. However, during audio playback, the track number is only decremented on a RW key if audio playback is within about 2 seconds (termed as *RW_SKIP_PERIOD* in the code) from the start of the track, as shown in Figure 4.7. This feature is common amongst CD players, where a RW key will cause audio playback to begin at the start of the current track, and if the key is pressed again within two seconds, the previous track is played.

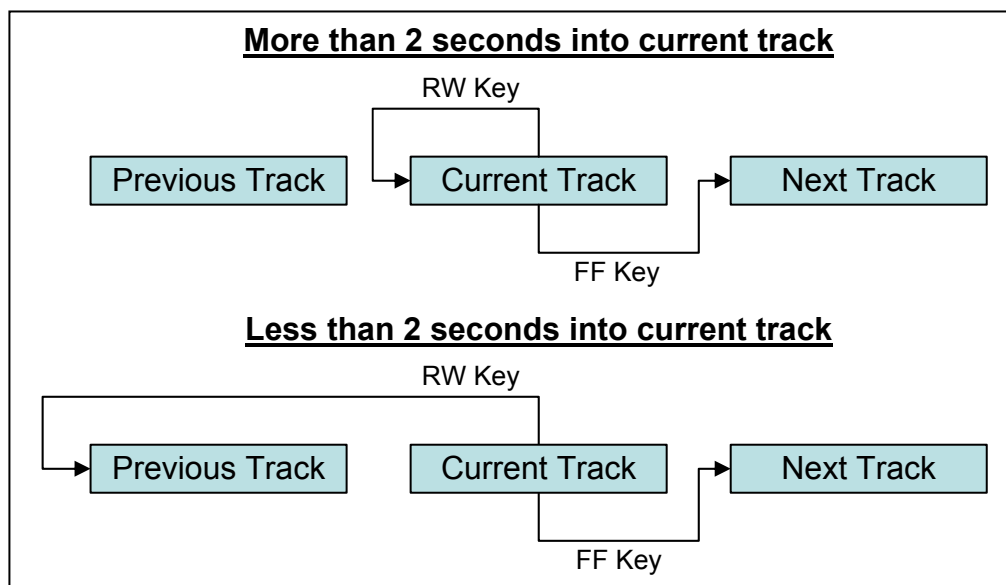


FIGURE 4.7: FF AND RW KEY FUNCTIONALITY DURING AUDIO PLAYBACK

If the CD contains one or more data tracks between the audio tracks, the generated TOC will contain a track entry which is used solely to hold the end position of the previous track (see Section 4.5 for more details). If one of these tracks is reached, it is skipped, and the following track is played for a FF key or previous track for a RW key (as this is the next audio track to be played). The first or last track in the

TOC track list is never a data track, so the track number does not need to be checked in this situation.

For each key, when the TOC is valid, the controller waits for a small delay (termed *SKIP_PRESS_DELAY*) before processing any other keys, and polls the next key in the buffer looking for another FF or RW key command. If another FF or RW key is in the buffer to be processed next, it is processed immediately without resuming audio playback. This eliminates any delays when a FF or RW key is pressed multiple times in quick succession and allows the user to jump many tracks quickly.

4.4.4 SCAN KEYS

When the forwards key or backwards key is pressed and held down it is stored as a scan key rather than a skip forward or skip backwards key as discussed above. When the scan forwards key (SCANF), or scan backwards key (SCANR) is processed, *audio_status* is checked and if audio is currently playing, it calls the *SXCD_scan* function as described below. When the function is complete, if the return value indicates either a problem or the beginning or end of the CD was reached, a stop key is loaded for processing, otherwise a play key is loaded to continue audio playback.

The *SXCD_scan* function handles both forwards and backwards scanning determined by the *direction* parameter passed to the function. The function flow diagram is shown in Figure 4.8 where it plays a small audio segment then either moves forward or backward depending on the direction, and then repeats. The size of the audio segment and the distance jumped determines the sound of the scanning audio. The ATAPI specification recommends a segment size of 6 frames (80 milliseconds), jumping 190 frames (2.53 seconds) when scanning forwards and 150 frames (2 seconds) when scanning backwards. However, it was found that a segment size of 6 frames and a jump size of 170 frames in either direction resulted in the desired sound and so this is used instead.

A *TrackInfoStruct* structure is defined as “*EndPos*” to hold the end position of each audio segment, which is the start position of the following track. At the start

of the loop *EndPos* is updated to be 6 frames after the current position (*CurrentPos*). A Read CD MSF command is then sent and audio playback of this 6 frame segment begins, but while the audio is playing the next position is calculated. This reduces the time delay between audio segments being played for better results. If the scan is in the forward direction, *CurrentPos* is calculated before the position is checked, while if the scan is in the reverse direction, it is calculated after. The position is checked to see if the start or end of the disc has been reached, in which case the function exits and audio playback stops. Once the audio segment has finished playing, several flags are checked so that if the scan button has been released, or remote input command is complete, the scan function exits, otherwise it repeats.

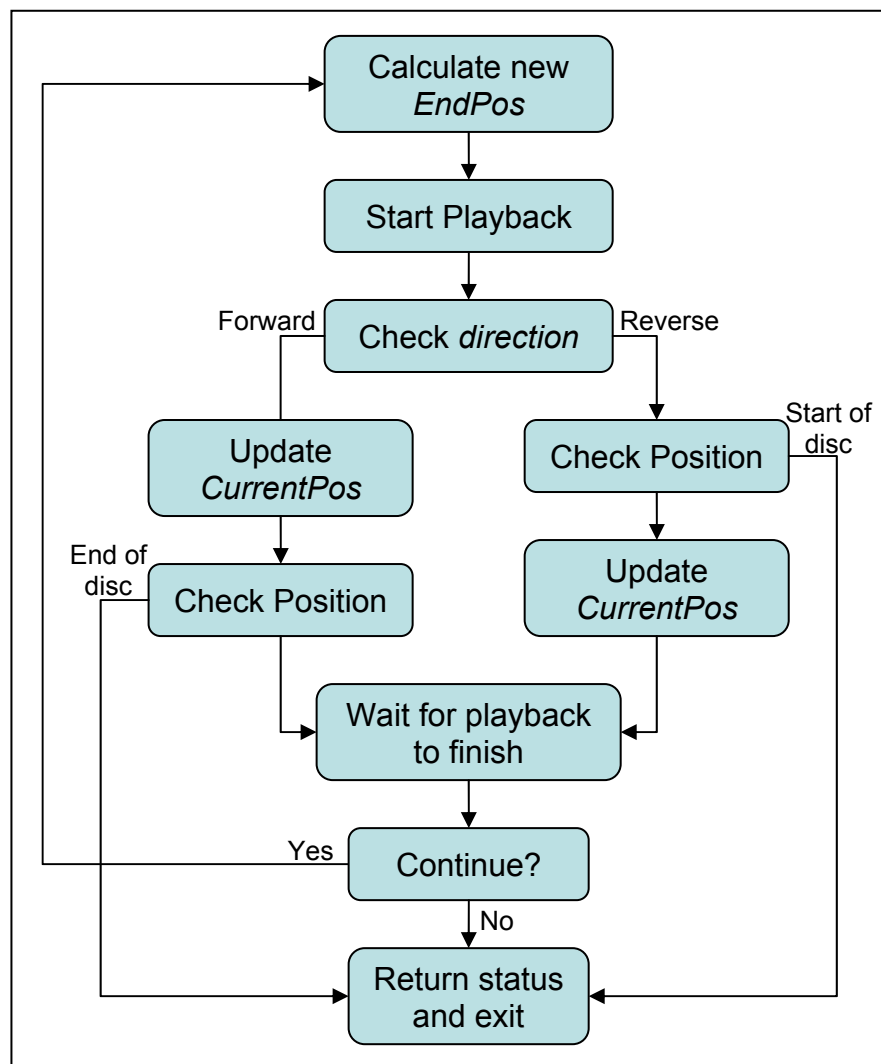


FIGURE 4.8: SCAN FUNCTION FLOW DIAGRAM

4.4.5 EJECT KEY

When the eject key is pressed, the tray status is unknown, so a mode sense command is sent. Within the data returned from this command is a “medium type” field. This field holds a value indicating both the type of media currently in the drive, and the status of the tray. The tray status is read from the “medium type” field, and if the CD drive tray is open, the *LoadTray* function is called. This function sends a “Start/Stop Unit” command where the value of the LoEj field (bit 1 of byte 4 of the command packet) determines if the drive should eject or load the tray. The LoEj bit is set to load the tray into the drive. If the tray is closed, the *EjectTray* function is called, which sends the “Start/Stop Unit” command with the LoEj bit cleared to eject the tray. The *EjectTray* function is continually called until the drive returns a “medium type” field that indicates the tray is open, because it is found that sometimes the drive will not eject the tray immediately if a previous command was interrupted. There is no action following either of these commands as the eject key is simply requesting that the tray be opened or closed.

4.5 READING THE TABLE OF CONTENTS

When a reset occurs or the CD drive tray is opened, a flag (called *TOC_invalid*) is set to one to indicate that the TOC is invalid. As described in Section 4.4.1, the table of contents (TOC) is read from the CD if a play command is processed and the *TOC_invalid* flag has been set. When the TOC is read the controller sends a Read TOC command to the CD drive with a “Format” field (byte 2 of the command packet) equal to 0x02. This indicates that all the Sub-Q Channel data (as described in Section 2.5.1) in the lead-in area be returned (where the TOC is stored). This differs from reading the basic TOC provided as it returns all session details and allows a TOC to be generated for audio tracks spanning over multiple sessions, even on a mixed data/audio disc.

For each track on the disc, 11 bytes of data are returned, starting with the session number followed by a Control field (ADR), a Point field with a MSF address, a Zero field, and an address pointer. The Point field determines the type of information in the two address fields that follow. The eight fields that are used by

the controller are shown in Table 4.4 with the various field possibilities for each Point field. The possible values for the Point field are summarised in Table 4.5.

ADR	Point	Min	Sec	Frame	Zero	PMin	PSec	PFrame
1	01-99	00 (Absolute time is allowed)			00	Start position of track		
1	A0	00 (Absolute time is allowed)			00	First Track num	Disc Type	00
1	A1	00 (Absolute time is allowed)			00	Last Track num	00	00
1	A2	00 (Absolute time is allowed)			00	Start Position of the lead-out area		
5	B0	Start time of the next possible program in the Recordable Area of the Hybrid Disc			# of pointers in Mode5	Maximum start time of the outermost Lead Out area in the Recordable area of the Hybrid Disc		
5	B1	00	00	00	00			
5	B2-B4	Skip#	Skip#	Skip#	Skip#	Skip#	Skip#	Skip#
5	01-40	End time for the interval that should be skipped			Reserved	Start time for the interval that should be skipped on playback		
5	C0	Optimum recording power	Application Code	Reserved	Reserved	Start time of the first Lead In Area of the Hybrid Disc		

TABLE 4.4: LEAD IN AREA (TOC), SUB CHANNEL Q FORMATS

Point value	Description
01-99	Track number references
A0	First track in the program area
A1	Last track in the program area
A2	Start location of the lead out area
B0	Contains the start time of the next possible program area
B1	Number of skip interval pointers and skip track assignments
01-40	Skip interval pointers
B2-B4	Skip track assignment pointers
C0	Start time of the first lead in area of Hybrid disc

TABLE 4.5: LEAD IN AREA (TOC) POINT VALUE DESCRIPTIONS

The ATAPI specification states that the Point field values A0-A2 must be the first entries from each session, which makes it easy to store the TOC as these contain the first and last track numbers. The track pointers which are the starting address of each track are then read out and stored in a *TrackInfoStruct* array called “*TrackInfo*”. If a non-audio (data) track is encountered, it is flagged as a lead-out track in an array called “*LeadoutTrack*” and any subsequent data tracks after it are

ignored until another audio track can be stored. If multiple sessions exist on the disc, the lead-out tracks for each session are also flagged in the “*LeadoutTrack*” array and the audio tracks for each session are appended onto the first.

The result is a TOC consisting of the information required to play back all the audio tracks on the disc. This includes the position of the start of the tracks where there is a discontinuity in the sequence of audio tracks. When one of these tracks is reached in audio playback, it is simply skipped, but its pointer value is used when the preceding track is played as both the starting and ending position (start of next track) of each track are required.

When each audio track is stored, the control field is read for the emphasis flag by masking it with a value of 0x01. This emphasis flag is stored in an array called *TrackEmph* and is later used to detect what tracks are recorded with emphasis. This array is then used to configure the emphasis option in the analogue and digital outputs. Both the *TrackEmph* array and the *LeadoutTrack* array are stored as a string of bit values spanning over multiple bytes. For example, the 9th flag in each array is the first bit on the second byte in the array.

4.6 AUDIO PLAYBACK

During audio playback the controller must transfer data from the CD drive, manage a small first in first out (FIFO) buffer, and latch the samples into the shift registers. These operations are to be performed for each audio sample, which occur at twice the sampling frequency as there are two channels.

When audio playback is to begin, the *SXCD_play* function is called from the main firmware loop (where the audio buffer is set up), the required interrupts are enabled, and then *go_flag* is set. This flag globally holds whether audio is to be played back or not (where audio playback can be stopped at any time simply by clearing *go_flag*). The flag is cleared when a valid key press is detected or a valid remote input command is decoded. The function then simply waits for the *go_flag* to be cleared (indicating audio playback has stopped) before disabling the interrupts, updating the current track number from the current audio position and

returning to the main firmware loop. With the interrupts enabled, audio playback is handled entirely within the INT0 interrupt routine, which is executed once for every *SR_EN* rising edge (every sample) and is shown in Figure 4.9.

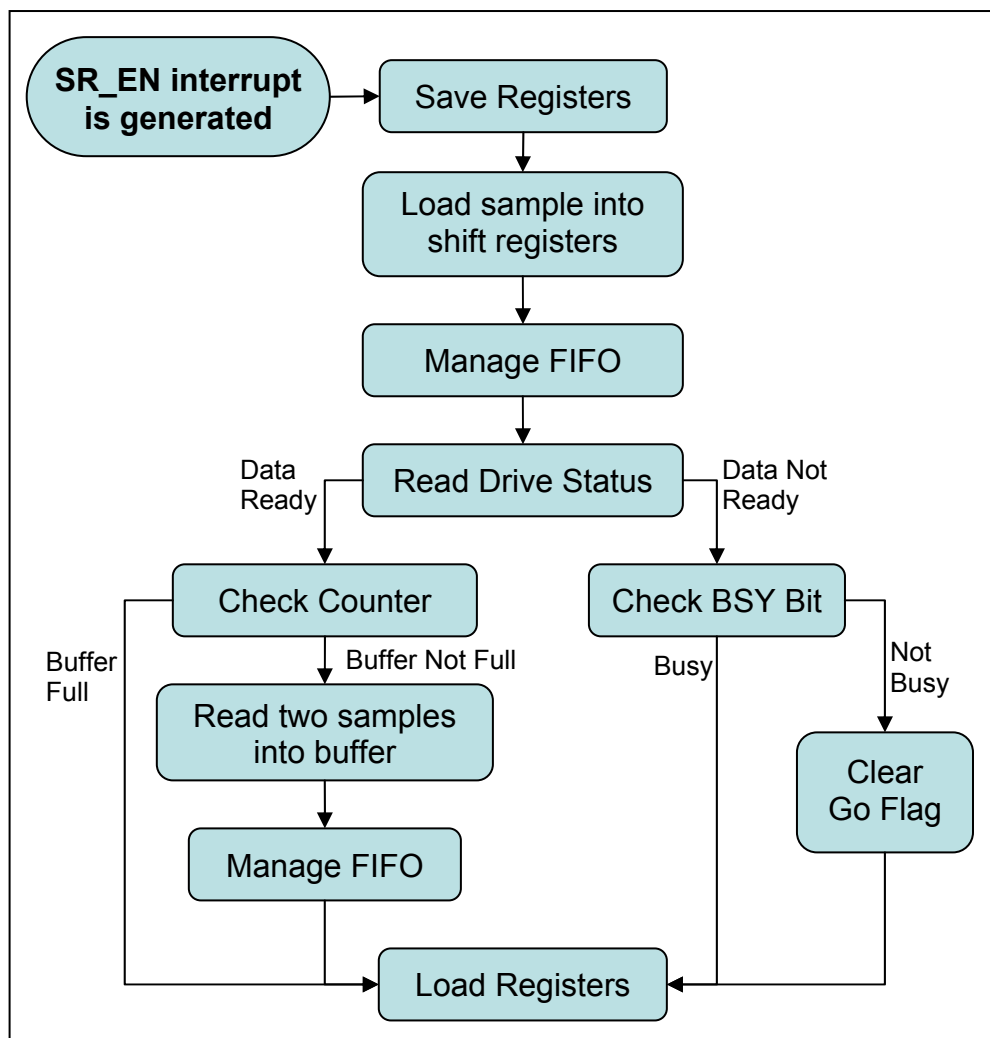


FIGURE 4.9: PLAY OPERATION STATE DIAGRAM

4.6.1 AUDIO BUFFER

The CD-DA audio data are stored in the controller's internal Random Access Memory (RAM) in a firmware implemented FIFO buffer. The buffer memory is reserved in a block entirely in the lower 255 bytes of address space so single byte read address and write address pointers can be used. These address pointers hold the next address to read from and write to and is incremented with each read and write performed. When either pointer reaches the end of the buffer memory, it is loaded with the address of the first address slot (byte 0).

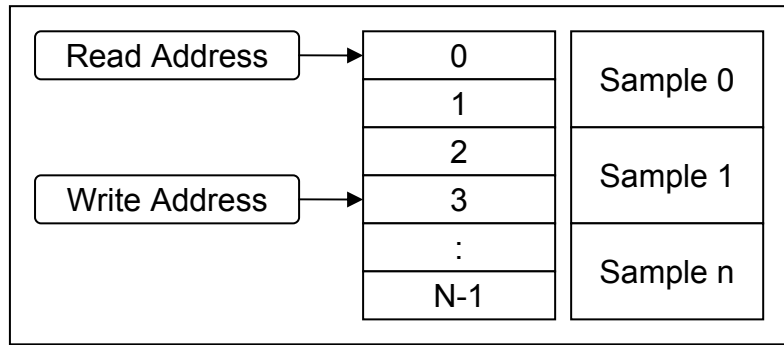


FIGURE 4.10: BUFFER OVERVIEW

Each sample is stored in pairs of bytes in the buffer, as shown in Figure 4.10, with the lower 8-bits being stored in the lower of the two bytes and the upper 8-bits in the upper of the two bytes. The buffer counter (*_counter* in assembly code) holds the current number of samples in the buffer and is incremented or decremented each time a sample is written to or read from the buffer. As shown in Figure 4.9, the counter is checked each time a sample is processed to determine if there is room in the buffer for two more samples.

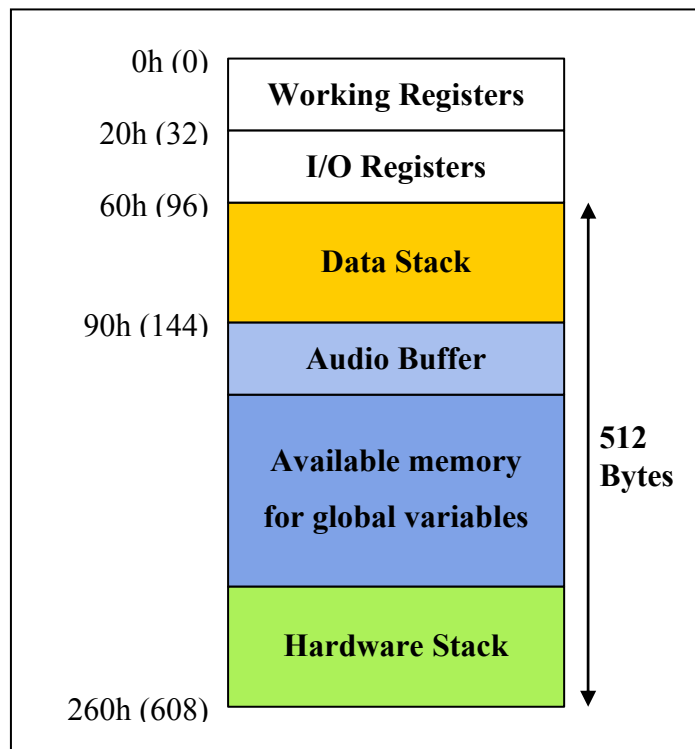


FIGURE 4.11: CONTROLLER MEMORY ORGANISATION

Two restrictions exist that limit the maximum size of the buffer. The first exists because the buffer must not extend past the first 256 bytes of RAM. The memory organisation is shown in Figure 4.11 where the first 96 bytes are used by the

working registers and I/O registers (not part of the dedicated 512 bytes of RAM). Above this is a data stack, which is used to store local variables and function parameters. Because 48 bytes are allocated to the data stack in the firmware, only 112 bytes remain available for the buffer.

The second restriction is due to the size of the hardware stack and the space required for global variables. The hardware stack is a section of RAM that is used for storing the return addresses of each function called (including interrupts). This area is allocated to be at the top of the RAM space and grows downwards as more memory is needed (more simultaneous functions are in progress). However, the memory required by the hardware is not known when the firmware is compiled. If not enough memory is reserved for the hardware stack at the top of RAM, it will extend into memory locations being used by other variables. The values of these variables will then be corrupted causing unwanted operation.

To overcome this problem, the memory locations at the top of RAM are monitored during program execution. The maximum number of bytes at the top of RAM that are modified by the controller is the size of the hardware stack, found to be approximately 24 bytes. To ensure enough RAM is provided, a block of 32 bytes is allocated. However the final firmware requires 368 bytes of RAM for global variables. This leaves only 64 bytes ($512 - 368 - 48 - 32$) available for the audio buffer (the maximum size given the firmware requirements). If less RAM was required for the global variables or hardware stack, the buffer could be extended, but not past the maximum size of 112 bytes.

4.6.2 DATA SETUP

Before audio playback can occur, the data in the buffer must be set up and the data for the left and right channel must be synchronised with the left/right (*WS*) clock. The buffer is configured by loading the first two samples into the buffer, setting the read pointer to the first address of the buffer, and the write pointer to the fourth address in the buffer. The reason two samples are initially read is twofold; firstly at least one sample must be read as a sample is sent during the interrupt routine prior to the buffer being replenished, and secondly, samples are always replenished in pairs. The data are synchronised to the left/right clock by waiting

for the *WS* clock line to be low (indicating a right channel sample) before enabling the *INT0* interrupt. The first sample is therefore transmitted as a left channel sample. This meets the format for CD-DA audio, where the byte stream starts with a left channel sample, and subsequent words alternate between left and right channel samples (page 149 of [34]).

4.6.3 TIMING RESTRICTIONS

The time available for the controller to achieve all the playback operations is limited, where the number of instruction cycles (N) available between each sample (*SR_EN* interrupt) is calculated with Equation 4-1 to be 90.7, given an oscillator frequency (f_{osc}) of 8 MHz and a sampling frequency (f_s) of 44.1 kHz. This means the controller has an average of 90 instruction cycles available to execute the code within the *INT0* interrupt routine in each sample cycle, where the term “sample cycle” is used to describe the time between successive samples.

$$N = \frac{f_{osc}}{2f_s} \quad \text{Equation 4-1}$$

To combat these time restrictions, the firmware has been written to minimise the time each operation takes. This has been achieved using four main methods, the first of which is that the firmware within the interrupt routine is completely written in assembly language, although the chosen language for the code is C. This is because assembly gives much more control over each instruction executed and the exact number of cycles can then be calculated. Part of the interrupt routine, written in assembly language, is shown in Figure 4.12, where the bracketed numbers are the number of cycles taken to execute each instruction.

The second method is disabling the automatic register saving within the compiler options, and saving only the minimum required registers manually. Some registers must be saved when entering an interrupt routine and restored when exiting if that register is used within the routine. This prevents the value in the register from being corrupted and causing unwanted behaviour. By saving the registers manually, more control over what instructions are executed when entering and

exiting an interrupt routine is obtained, and the number of registers that are saved and restored is reduced.

```
        ; ***** Load the sample *****
        ; Data bus = Outputs:
LDI    R30,0xFF          ; (1)
OUT    DATA_LO_DIR,R30 ; (1)
OUT    DATA_HI_DIR,R30 ; (1)
        ; Load the read pointer
LDI    R27,0x00         ; (1)
MOV    R26,_rd_ptr      ; (1)
        ; Put out data low:
LD     R30,X+           ; (2)
OUT    DATA_LO,R30     ; (1)
        ; Put out data high:
LD     R30,X+           ; (2)
OUT    DATA_HI,R30     ; (1)
        ; Latch data into registers (PORTD.4)
CBI    SR_EN,4         ; (2)
SBI    SR_EN,4         ; (2)
        ; Save back the read pointer
MOV    _rd_ptr,R26     ; (1)
        ; Data bus = Inputs:
LDI    R30,0x00         ; (1)
OUT    DATA_LO_DIR,R30 ; (1)
OUT    DATA_HI_DIR,R30 ; (1)
```

FIGURE 4.12: LOAD SAMPLE CODE IN ASSEMBLY LANGUAGE

The third method is to allocate the 16 general purpose registers manually and assign them to time critical variables, even if that variable is only one bit in size, such as a flag. When a variable in RAM is accessed, that address location must be copied to a register before any operation can take place on that value. By storing a variable in a register instead of in RAM this first step is eliminated, and the time taken to access that variable is halved to a single instruction.

The last method is to store actual memory locations instead of using pointers to access the data buffer. Usually when a buffer is accessed, an index pointer is added to the starting location of the buffer in order to obtain the address of the desired value. The pointer is also treated as two bytes in size, as the memory space extends past address location 255. Buffer access time can be significantly reduced by storing the actual address location in a single byte register and specifying the buffer memory location to be in the lower 256 bytes of the memory space. Using this method it is calculated that buffer access time is 187 % faster

than using normal C code, reducing the number of instruction cycles down to only 8 as shown in Figure 4.12.

4.6.4 ACTUAL TIMING

Although the timing restrictions exist, there is only one requirement for continuous audio playback to occur, which is having the data latched into the shift registers before each transition in the WS clock signal. The *SR_EN* interrupt routine firmware requires only 29 cycles to latch the data into the shift registers after the interrupt occurs. With the controller running at a speed of 8 MIPS, 45 cycles are available before the data are shifted out as shown in Figure 4.13. Therefore, an interrupt routine can be delayed up to a maximum of 16 cycles (45 - 29) after the interrupt occurs. The final *SR_EN* interrupt routine takes a total of 92 cycles to execute on the longest path. This will extend into the following cycle by 2 cycles, therefore a maximum of 14 cycles remain for the “interrupt dead time”.

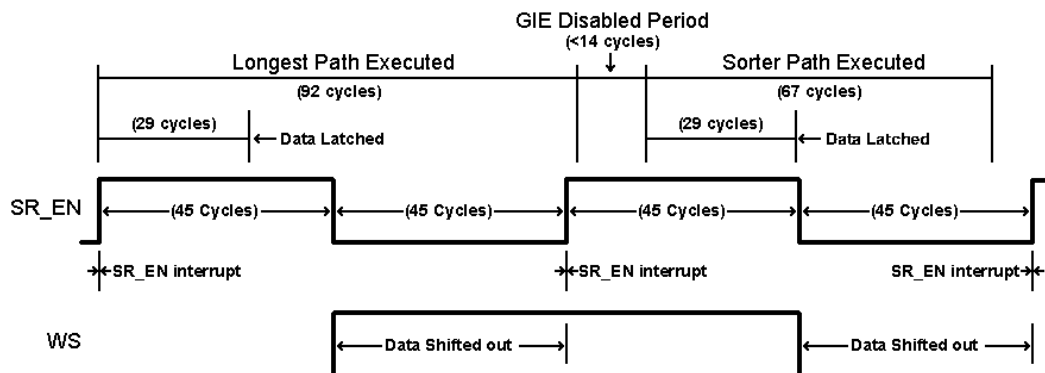


FIGURE 4.13: *SR_EN* INTERRUPT WORST CASE TIMING

When this worst case scenario occurs, the sample cycle period immediately after the longest path was executed has only 74 cycles (90 - 16) remaining to execute the entire interrupt routine. However, with the way the routine is designed, the longest path cannot be executed on two successive occasions, nor is it required to, and the second longest path of execution takes only 67 cycles to complete. This gives enough time before the next sample cycle and results in continuous audio playback as long as no “interrupt dead time” is longer than 14 cycles. This applies to any of the lower priority interrupts that will be occurring during audio

playback, and includes the user interface and remote input interrupts, and the compare interrupt for Timer 1.

4.6.5 CURRENT AUDIO POSITION

One problem encountered is the retrieval of the current position at the end of audio playback. The only way to do so in the ATAPI specification is to read the data with the “READ SUB-CHANNEL” command, yet the command states that the position given here may be one of two as *“the ATAPI CD-ROM Drive may either report position data for the last sector processed for that operation or may report position data from the sector at the current read head position”* (page 163 of [34]). However the current read head position can be several seconds different from the position of the data last read from the ATA interface. Therefore this command is rendered unreliable for maintaining the current audio position for pause and scanning functions.

One reliable way to keep track of the current position is to use a counter internal to the controller, but to eliminate the possibility of time shifting, the counter must be synchronised from a clock signal. As a result of the final controller pin configuration, the trigger pins for the two counters are the bottom two address lines for the ATA interface. Because these lines toggle during each sample read (as the address toggles between 0x10 and 0x17), the counter can be incremented at a rate of 88200 times a second. Also, by using the compare feature of counter 1 with a value of 1175, a compare interrupt can be generated every frame (75 times a second). This can be used directly to increment the current MSF position of the audio in real-time.

The only disadvantage of this approach is the extra firmware that must be implemented in the controller, which uses some of the time available for audio playback. Within the Timer 1 compare (TIM1_CMPA) interrupt routine, which is executed each time the timer value equals 1175, the “frames” value for the current position is incremented. When it reaches 75, it is set to zero and the “seconds” value is incremented. When this reaches 60, it is set to zero and the “minutes” value is incremented, thus maintaining the current audio position with audio playback.

4.6.6 EMPHASIS CONTROL

Although not very common, the ATAPI standard is set up for CD-DA audio discs to be recorded with emphasis. To provide this functionality additional firmware is included. As described in Section 4.5, the emphasis flags for each audio track are stored whenever the table of contents is read. Before any audio playback occurs (including scan commands) the *update_emphasis* routine (shown in Figure 4.14) is executed, where the emphasis flag for the current track is read and de-emphasis is turned on or off for both digital and analogue playback.

For digital playback, this is simply a matter of setting the emphasis flag in the SPDIF channel status format field [9]. To achieve this in the DIT4192 SPDIF chip, buffer transfer must first be disabled, then the emphasis flags in both channels are modified, and buffer transfer is re-enabled. For the onboard analogue playback, the de-emphasis enable bit in the PCM1780 DAC chip is modified which enables or disables the de-emphasis control feature.

```

/* Updates the De-Emphasis on the DAC and Emphasis flag
   in the SPDIF transmission for the current track.
   TIMING: Takes approx 2.0ms without debug. */
void SXCD_update_emphasis (void)
{
    // Check emphasis bit for current track:
    if (GetArrayFlag (TrackEmph, CurrentPos.Track)) {
        // Turn on De-Emph in DAC:
        DAC_wr (19, 0x10);    // De-Emphasis = On
        // Set De-Emph flag in SPDIF transmission:
        SPDIF_wr (0x07, 0x01); // Disable buffer transfer
        SPDIF_wr (0x08, 0xB2); // 44.1 kHz, Professional, Emph On
        SPDIF_wr (0x09, 0xB2); // 44.1 kHz, Professional, Emph On
        SPDIF_wr (0x07, 0x00); // Enable buffer transfer
    } else { // Emphasis is off:
        // Turn off De-Emph in DAC:
        DAC_wr (19, 0x00);    // De-Emphasis = Off
        // Clear De-Emph flag in SPDIF transmission:
        SPDIF_wr (0x07, 0x01); // Disable buffer transfer
        SPDIF_wr (0x08, 0xA2); // 44.1 kHz, Professional, Emph Off
        SPDIF_wr (0x09, 0xA2); // 44.1 kHz, Professional, Emph Off
        SPDIF_wr (0x07, 0x00); // Enable buffer transfer
    }
}

```

FIGURE 4.14: UPDATE EMPHASIS FUNCTION

4.7 USER INTERFACE

As described in Section 3.9, the user interface consists of 5 buttons, which all connect to a common interrupt line on the controller. When a button is pressed, an INT1 interrupt is generated, and the code within the *user_int* interrupt routine is executed. Within this routine the button value is read from the low data bus, decoded, and pushed onto a key buffer stack. The routine is shown in the diagram in Figure 4.15.

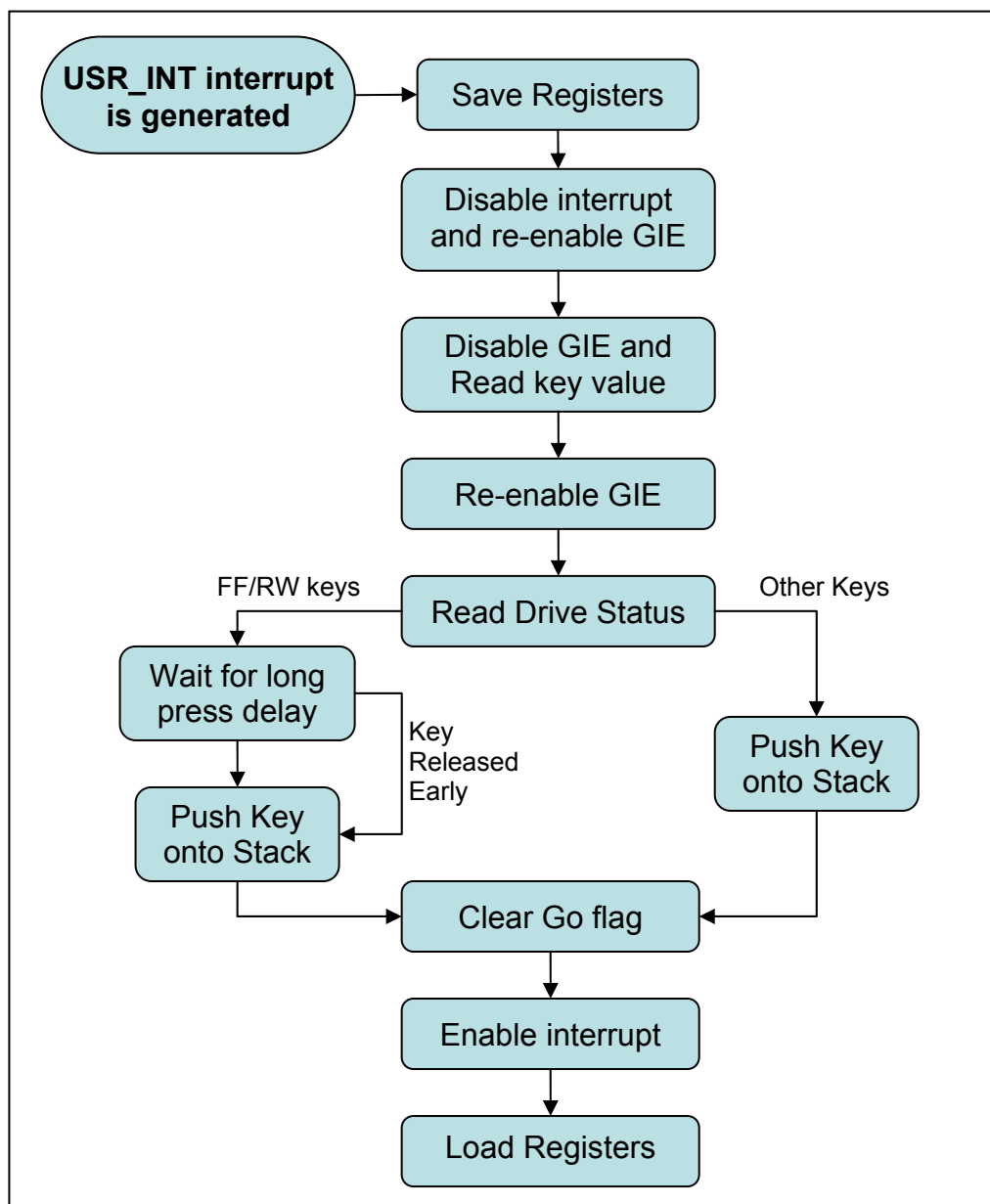


FIGURE 4.15: USER INTERFACE INTERRUPT ROUTINE STATE DIAGRAM

When an interrupt is generated, the controller saves the required registers manually and immediately disables the INT1 interrupt and re-enables the global interrupts (GIE), allowing any pending interrupts to be executed. No other action can be executed before this time, due to the 14 cycle limitation on the interrupt dead time as described in Section 4.6.4. However, the key value must be read from the low data bus while interrupts are disabled to ensure a bus collision does not occur between the key value and any CD-DA data being transferred on the bus. For this reason, GIE is disabled a second time so the key value can be read, and then re-enabled again. This essentially splits the interrupt dead time into two time periods, both of which are smaller than the 14 cycle limitation, while allowing audio playback to operate properly.

The key value is then decoded. If the value is the fast forward (FF) key or the rewind (RW) key, it is decoded as a press and hold key, otherwise the key value is used directly. A press and hold key is decoded by waiting for a press and hold delay. If the key is released within this time delay, the keys are processed as a quick press. If the key is held down for the duration of the delay, the key is processed as a press and hold key. For the FF key the press and hold function is scan forward (SCANF) and for the RW key it is scan backward (SCANR).

If a valid key value is obtained, that value is passed to the *push_key* function (shown in Figure 4.16), where the key value is pushed onto the end of the key buffer stack and *num_keys* is incremented. Finally, *go_flag* is cleared to cancel any current play command as all valid key presses result in the discontinuity of audio playback.

```
/* The push_key and pop_key routines are used to manage the
   key FIFO buffer. When a key is pressed, push_key is called
   to push it onto the stack, and then pop_key is called to
   retrieve each key press. Just check num_keys for the number
   of keys on the stack. */
char push_key (char key)
{
    if (num_keys < KEY_BUFFER_SIZE) {
        key_buffer [num_keys] = key;
        num_keys ++;
    } else {
        return FAILURE;
    }
    return SUCCESS;
}

char pop_key ()
{
    char key, i;
    if (!num_keys) {
        return KEY_NULL;
    } else {
        key = key_buffer [0];
        for (i = 0; i < num_keys-1; i ++ ) {
            key_buffer [i] = key_buffer [i+1];
        }
        num_keys --;
        return key;
    }
}
```

FIGURE 4.16: PUSH KEY AND POP KEY FUNCTIONS

4.8 REMOTE INPUT

As described in Section 3.11, the remote input circuit filters the signal, removing the carrier frequency, and provides a demodulated signal to the controller. The signal triggers the *remote_int* (INT2) interrupt on a rising edge, and the INT2 interrupt routine is then executed, decoding the remote command. The flow diagram for the interrupt routine is shown in Figure 4.17.

When the interrupt routine is executed, the required registers are saved manually, global interrupts are re-enabled as above, and the INT2 interrupt is left disabled to prevent it from nesting within itself. The *remote_flag* flag is then checked and if it is set, *remote_cnt* (a remote input timeout counter) is reset to zero and the whole operation is aborted, indicating a remote scan command is in progress.

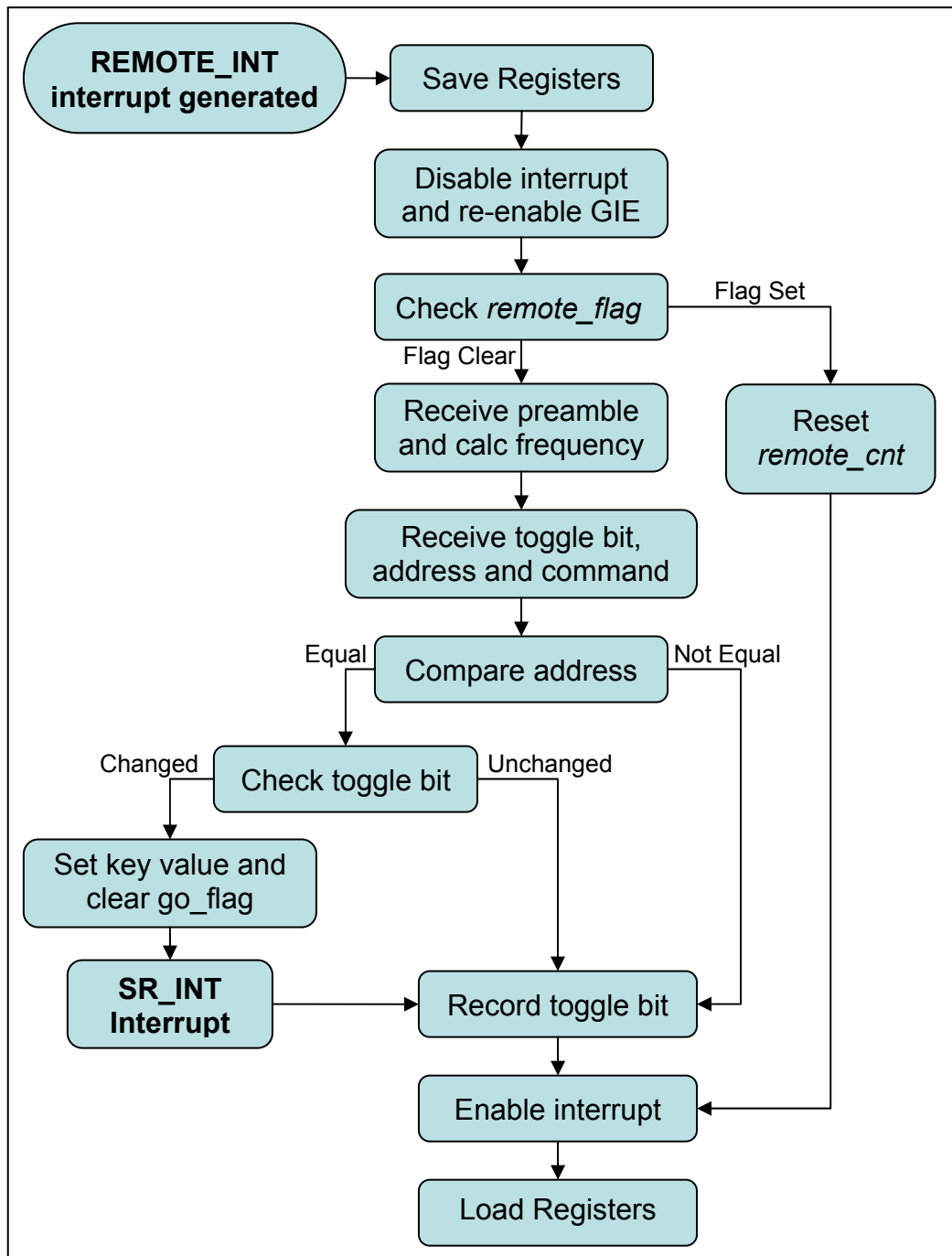


FIGURE 4.17: REMOTE INPUT INTERRUPT ROUTINE STATE DIAGRAM.

$$F_{value} = \frac{3}{4}N = \frac{3}{4} \frac{31250}{f_{BR}} \quad \text{Equation 4-2}$$

If *remote_flag* is clear, the preamble is received, from which a frequency value (F_{value}) is calculated according to Equation 4-2, where N is the number of Timer 0 counts during 1 period of the preamble, and f_{BR} is the signal bit rate frequency (562.5 Hz for RC5 [26]). The frequency value is used to ensure each

sample is taken in the middle of the second half of each period of the remote input signal as shown in Figure 4.18. This helps to maximise the systems tolerance to error and results in a robust system that can adjust to a wide range of bit frequencies. Bit by bit, the toggle, address and data fields are received (according to the RC5 specification) and if the address matches the dedicated SXCD address and the toggle bit has changed since the last received command, the command is processed.

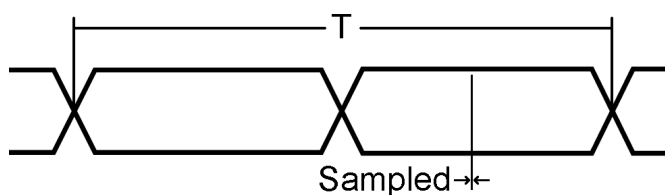


FIGURE 4.18: SAMPLE LOCATION FOR REMOTE INPUT

4.9 THE LED

As with all other Silhouette products, a blue Light Emitting Diode (LED) is mounted on the front panel to indicate that the unit has power. For the CD player, this LED is also used to indicate the current player status, where the LED will flash at three different rates depending on whether the player is busy, paused during audio playback, or an error occurred. The different LED states are summarised in Table 4.6.

LED	Rate	Status	Note
On	NA	Waiting/Playing	Waiting for user, or playing audio.
Flashing	2-4 Hz	Busy	Processing command or tray eject/load.
Flashing	1-2 Hz	Paused	Audio is paused.
Flashing	6-10 Hz	Error	Bad or no disc or drive error.

TABLE 4.6: LED STATUS SUMMARY

The LED is connected to a controller pin through a simple MOSFET inverter to provide some current gain, and the LED flash rate is handled by the Timer 0 overflow interrupt. A global variable, called *LED_delay*, is used to set the flash rate of the LED, which is then used in the interrupt routine. A counter is

incremented each time the interrupt routine is executed and when the counter reaches the value of *LED_delay*, the LED state is toggled and the counter is reset. Therefore a lower value of *LED_delay* results in a faster flashing rate for the LED. The LED flashing is enabled and disabled by enabling or disabling the Timer 0 overflow interrupt. When disabled, the LED state is set to be on.

5 FINAL DESIGN PERFORMANCE

The project underwent three board revisions before a final production-ready prototype was completely functioning. At the time of project completion, Perreux had not obtained the chassis and front panel samples and so the prototype could not be viewed as a final product (preliminary front panel design files are available on the attached project CD). However, with a temporary interface set up on a breadboard, this final revision is able to be tested as a fully working prototype as shown in Figure 5.1.

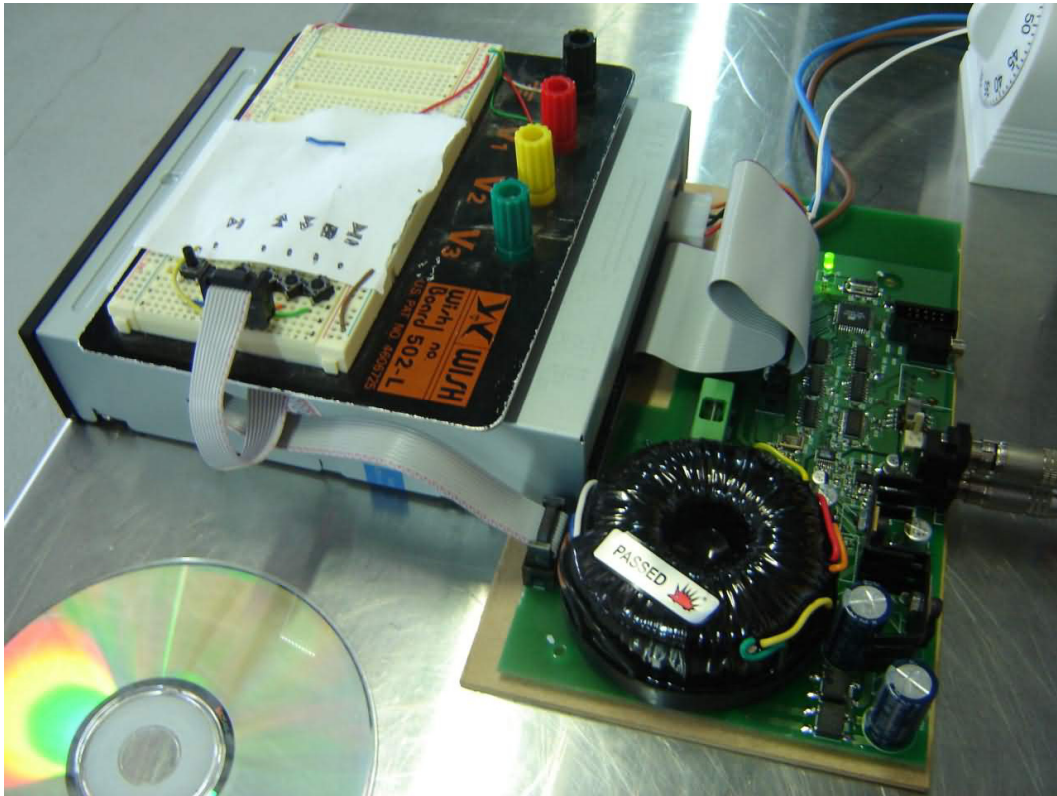


FIGURE 5.1: PROTOTYPE RUNNING WITH TEMPORARY INTERFACE

5.1 PHYSICAL CONSTRUCTION

The outline of the final revision of the PCB is shown in Figure 5.2, where the back of the board (rear of the chassis) is positioned at the top of the image. The outside dimensions measure 203.6 mm wide and 110.0 mm deep. The width fits

inside the width of the standard Silhouette series chassis which is approximately 205 mm.

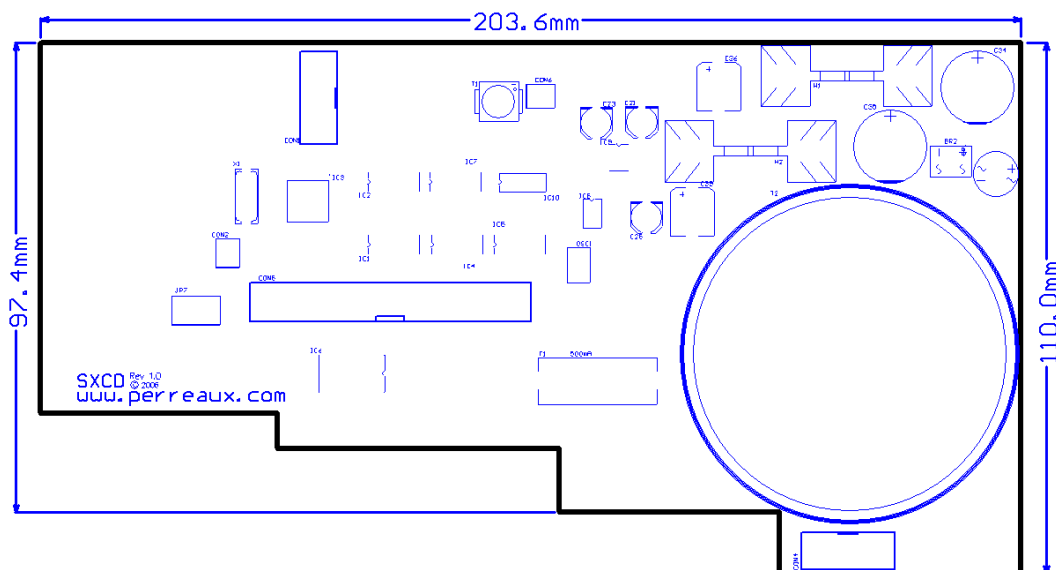


FIGURE 5.2: PCB OUTLINE AND CRITICAL DIMENSIONS

The critical depth is also shown, which is the distance between the back of the PCB and the edge that the back of the CD drive sits next to. This edge is the second level up the 4-level edge on the front of the board. The critical depth is 97.4 mm, which fits inside the maximum 100 mm allowed for the PCB depth as described in Section 3.15. The other two levels are to allow room for the ATA interface cable (middle level) and the CD drive power connector (left hand level). These levels can be seen (along with the connectors) in the photo in Figure 5.3. The right hand level on the front edge of the board is positioned next to the right hand side of the back of the CDROM.

Also shown in Figure 5.2 are the positions of the main components on the PCB. The transformer (the critical component) is shown by the large round circle on the right hand side of the board. Here the transformer is positioned as far right and as close to the front edge as possible to allow room for the power supply components behind it. The power supply components and transformer can be seen in the image in Figure 5.4.

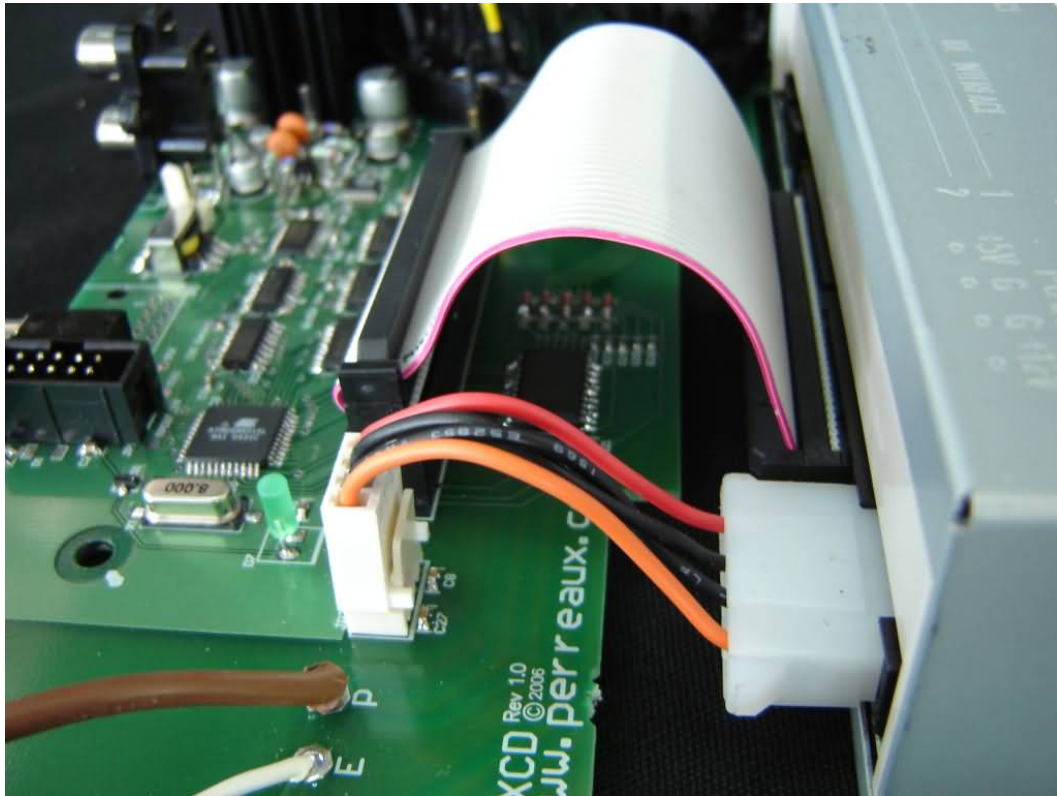


FIGURE 5.3: CD DRIVE INTERCONNECTS

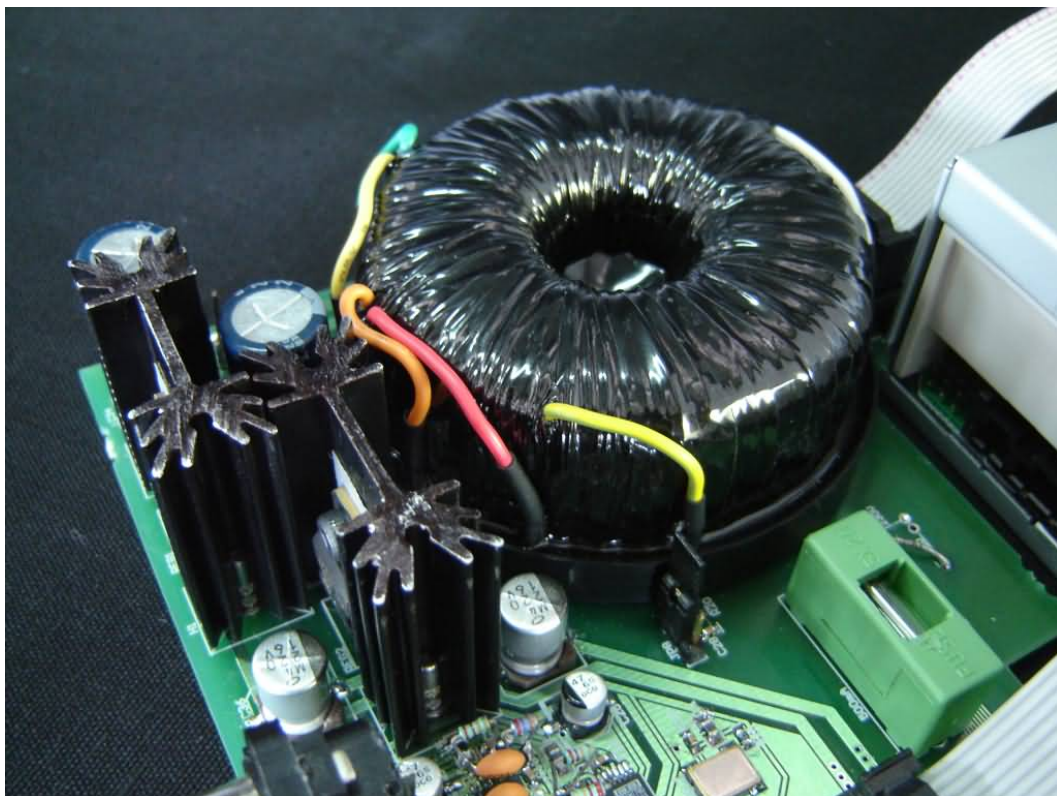


FIGURE 5.4: COMPLETED POWER SUPPLY

5.2 SXCD OPERATION

There are three main operating states for the SXCD: these are *Playing*, *Tray Open* and *Waiting* as shown in Figure 5.5. During the *Waiting* state, the drive has a valid TOC and has the disc spinning ready to process a command. The transitions between these states occur when a Play, Stop, or Eject key is pressed. The time it takes to make each transition depends on the state of the drive and is shown by the bracketed numbers (in seconds).

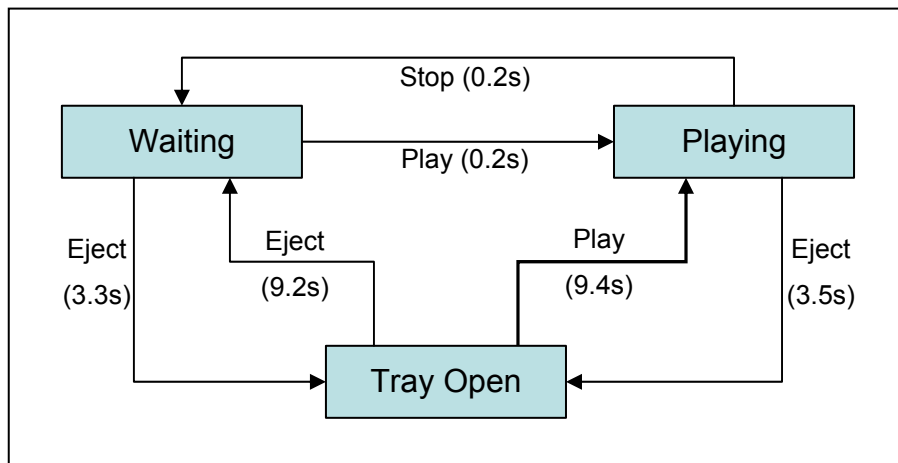


FIGURE 5.5: SXCD MAIN OPERATION STATE DIAGRAM

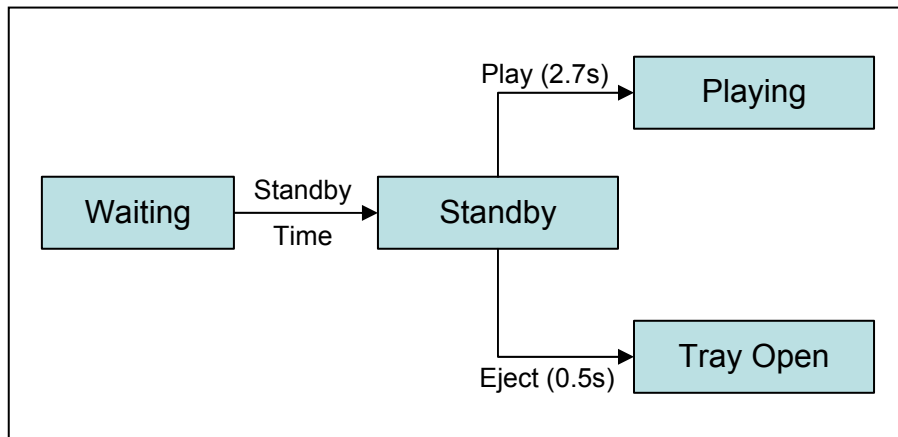


FIGURE 5.6: SXCD STANDBY STATE DIAGRAM

When the drive is inactive for an extended period of time (approximately 10 minutes) the drive can enter a *Standby* state as shown in Figure 5.6. During this state, the system has a valid TOC, yet the drive has spun down the disc. From this state the drive will either move to the *Playing* state or the *Tray Open* state on the

press of a Play key or Eject key respectively. The times for these transitions are shown on the respective transitions in the diagram.

An important transition is between the *Tray Open* and *Playing* states. This is the time it takes for the tray to load, the disc to spin up and audio playback to begin from the time the Play key is pressed. The transition takes approximately 9.4 seconds which is as a typical time from other CD players that were available at Perreux was approximately 10 seconds. A time of greater than 15 seconds would certainly be unacceptable. The other transitions are also performed at satisfactory speeds. When the drive is in a Standby state, it consumes less power as the disc is not spinning. If the Play key is then pressed the disc must be spun up before playing audio, however the transition only takes 2.7 seconds.

The times for these transitions are largely due to the CD drive make and model being used. The times recorded are for an Asus CRW-5232A CD writer drive as this was the chosen drive for the final prototype design. All the times recorded are satisfactory, and any variation in the times would not be a cause of concern provided the *Tray Open* to *Playing* transition remains below 15 seconds.

5.2.1 CD FORMATS

A mixed mode test CD was made containing three audio tracks and one data track, and was played with no detectable errors. The TOC data read from the CD are shown in Table 5.1. As described in Section 4.5, the Point field determines the type of information contained on each row, being either a track listing or other information. The first entry (Point value of A0h) holds the first track number in the “*PMin*” column and the second entry (A1h) holds the last track number. From these two values the number of tracks is calculated to be four. The A2h entry holds the start address of the lead out area, which is stored and used as the end position of the last track on the CD (if it is an audio track). The remaining entries (the lower four rows) contain the starting address of each of the four tracks in the “*PMin*”, “*PSec*”, and “*PFrame*” fields (the “*PMin*” and “*PFrame*” entries are only 00h because the tracks are exactly 10, 12, 12 and 10 seconds in length respectively). The “*ADR*” field is used to differentiate between data and audio

tracks, where bit 2 is clear for audio and set for a data track. As shown, the first track is data, and the last three (Point values of 02h-04h) are the audio.

The resulting TOC containing the three audio tracks is stored in “*TrackInfo*” which has four entries. Three of the entries are the audio tracks as shown in Table 5.2, while the last valid entry in the array (entry number 3) is labelled as a lead out track as indicated in the “*LeadoutTrack*” array in Table 5.3. This means that this entry is only used to determine the end of the previous audio track, contained in the previous entry. Only the first four entries in each array are considered valid as the *NumTracks* variable holding the number of tracks in the table is set to four (three audio tracks and one “*Leadout*” track).

Sess	ADR	Point	Min	Sec	Frame	Zero	PMin	PSec	PFrame
01h	14h	A0h	00h	00h	00h	00h	01h	00h	00h
01h	14h	A1h	00h	00h	00h	00h	04h	00h	00h
01h	14h	A2h	00h	00h	00h	00h	00h	2Eh	00h
01h	14h	01h	00h	00h	00h	00h	00h	02h	00h
01h	12h	02h	00h	00h	00h	00h	00h	0Ch	00h
01h	12h	03h	00h	00h	00h	00h	00h	18h	00h
01h	12h	04h	00h	00h	00h	00h	00h	24h	00h

TABLE 5.1: TOC FOR EXAMPLE MIXED MODE CD

	0	1	2	3	4	5	...
M	0	0	0	0	-	-	...
S	12	24	36	46	-	-	...
F	0	0	0	0	-	-	...

TABLE 5.2: *TRACKINFO* ARRAY CONTENTS

0	1	2	3	4	5	...
0	0	0	1	-	-	...

TABLE 5.3: *LEADOUTTRACK* ARRAY CONTENTS

A similar test was performed using a multi session disc (Coldplay X&Y CD, 2005) where the second session contained a video file (data track). An extract of the TOC read from the disc is shown in Table 5.4. The table shows the last three audio tracks of the first session (“*Sess*” value of 01h) indicated by the ADR values of 50h. These are followed by an entry showing the start address of the next

program area (“*Point*” value of B0h) and an entry stating the start time of the first lead in area (“*Point*” value of C0h). The next session (“*Sess*” value of 02h) begins in the same manner as the first session with two entries holding the first and last track numbers (Point values of A0h and A1h), and an entry holding the starting address for the start of the lead out area. Finally, the data track entry for the video file is listed, where the “*Point*” value shows a track number of fourteen (0Eh) which continues from the previous session that contained 13 tracks in total. This disc, as well as all the SXCD functionality, performed as expected during testing of the firmware for reading the TOC.

Sess	ADR	Point	Min	Sec	Frame	Zero	PMin	PSec	PFrame
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
01h	10h	0Bh	00h	00h	00h	00h	31h	19h	40h
01h	10h	0Ch	00h	00h	00h	00h	35h	18h	3Dh
01h	10h	0Dh	00h	00h	00h	00h	3Ah	1Ah	33h
01h	50h	B0h	41h	07h	1Ah	02h	4Ah	00h	00h
01h	50h	C0h	00h	00h	00h	00h	5Fh	00h	00h
02h	14h	A0h	00h	00h	00h	00h	0Eh	00h	00h
02h	14h	A1h	00h	00h	00h	00h	0Eh	00h	00h
02h	14h	A2h	00h	00h	00h	00h	4Ah	00h	00h
02h	14h	0Eh	00h	00h	00h	00h	41h	09h	1Ah

TABLE 5.4: EXTRACT FROM COLDPLAY X&Y CD TOC

5.3 METHOD OF MEASUREMENT

Perreux’s testing and measurement equipment is based around the ATS-2 Audio Analyser from Audio Precision [35]. This analyser is a high quality and world renowned piece of measuring equipment, and has been providing Perreux engineers with performance results well within its limits. The ATS-2 is connected to a computer and by using its ATS control software (version 1.2) it provides capabilities such as sweeps, single point measurements, spectrum analysis, multitone analysis and discrete harmonic analysis. The SXCD final prototype is shown connected to the test equipment in Figure 5.7.

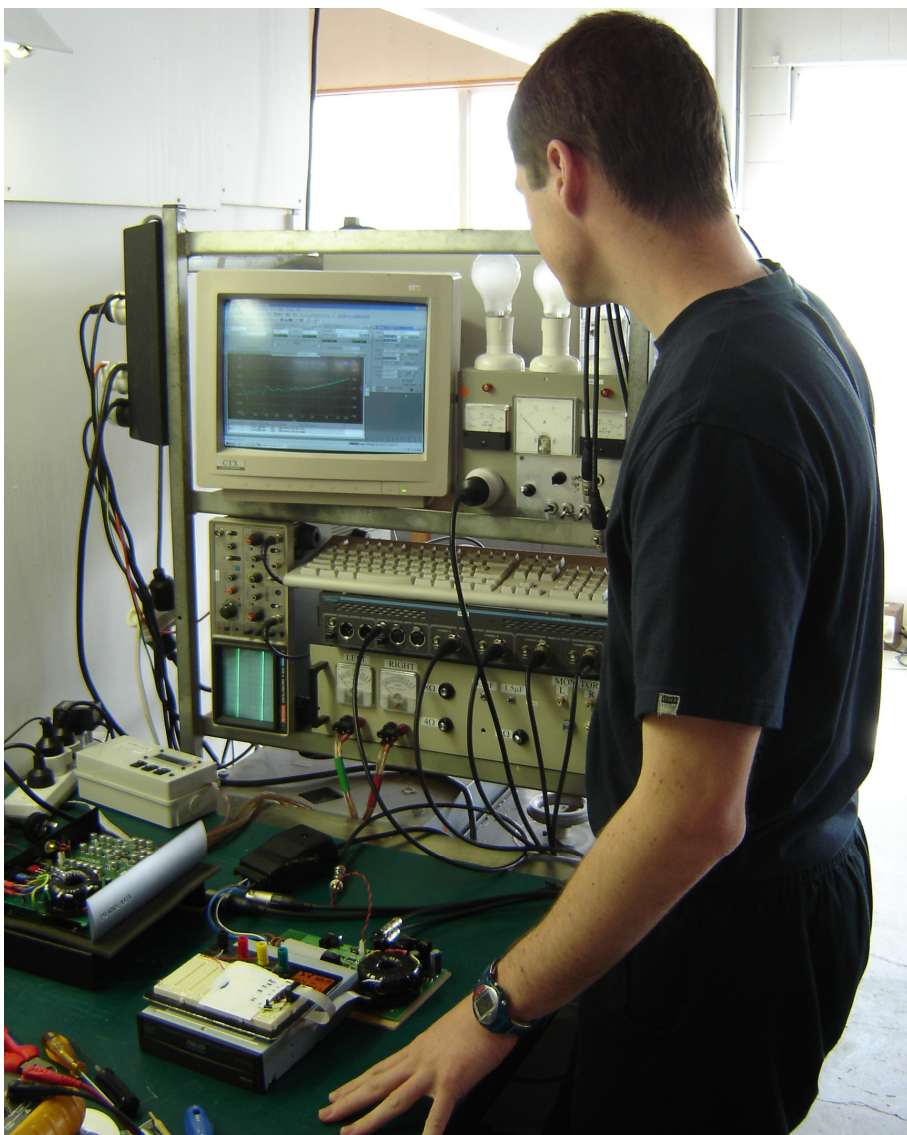


FIGURE 5.7: THE SXCD UNDERGOING TESTS

The traditional method of measuring the quality of a piece of audio equipment such as an amplifier or pre-amplifier is to connect the output of the audio analyser to the input of the device under test. The output of the device is fed back into the audio analyser as shown in Figure 5.8. Here a test signal is generated in the audio analyser, and the output of the device is measured relative to the generated signal. For many of the tests, the signal will sweep over a range of frequencies so graphs can be produced for parameters such as total harmonic distortion (THD), signal to noise ratio (SNR) or frequency response.

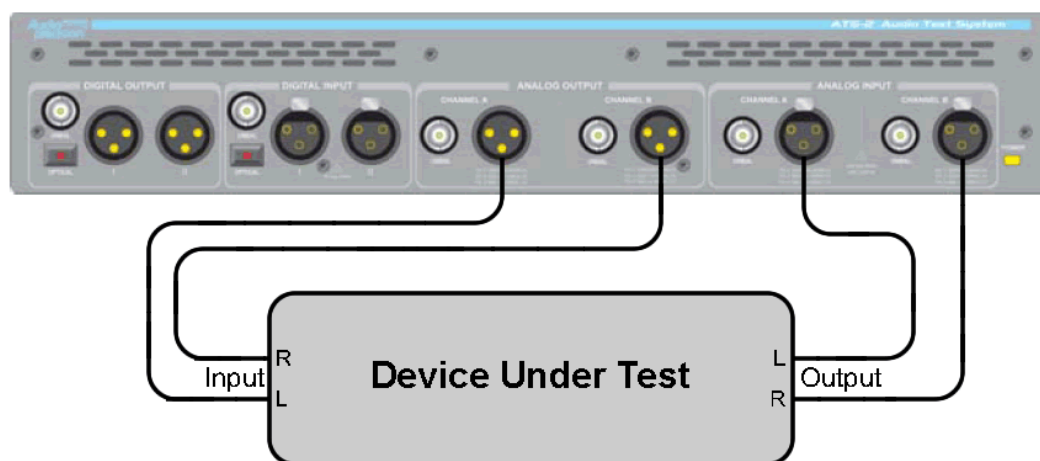


FIGURE 5.8: TYPICAL ATS-2 TEST SETUP

However, with CD players this is not possible as the only input source available is from a CD. To resolve this problem a test CD is made with computer generated signals to be played in the CD player and the output of the player can then be compared with the original signal file. With the SXCD, both the analogue and digital outputs can be measured when configured as shown in Figure 5.9. This method of testing CD players using the Audio Precision ATS-2 and software is well documented in an application note from Audio Precision [36]. The software is used to generate a multitone test file containing 50 different frequency components within the audio range of 20 Hz to 20 kHz, which will result in 50 different point measurements. When this multitone signal is played back, the ATS-2 synchronises the received signal with the original signal and compares the two using the multitone signal analyser feature for either the digital or analogue input.

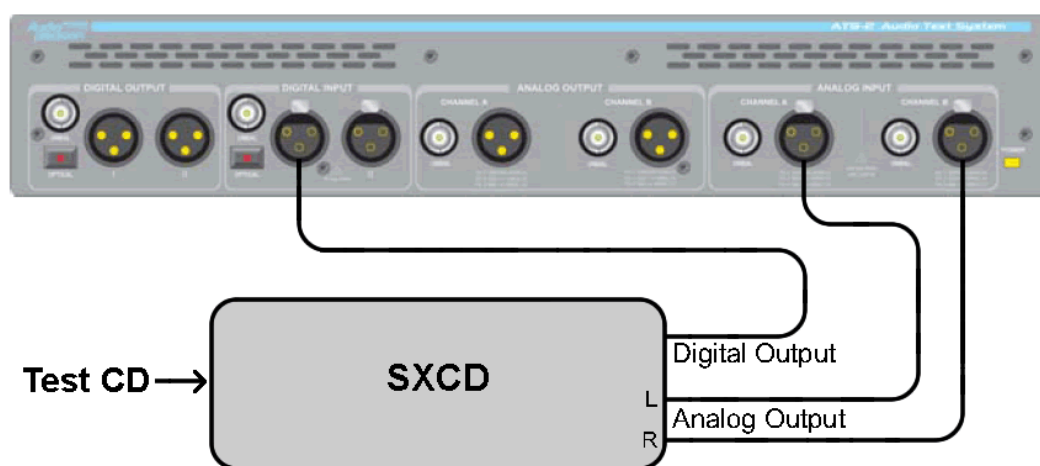


FIGURE 5.9: SXCD ATS-2 TEST SETUP

5.4 ANALOGUE OUTPUT

The analogue output performance was measured for the final board revision. The ATS-2 test equipment was used to obtain several graphs such as the frequency response, THD and SNR. The graphs produced by the ATS-2 are expressed in dBV, the decibel value relative to a signal 1 volt RMS in amplitude.

5.4.1 FREQUENCY RESPONSE

A very common measurement in an audio system is the frequency response over the audio range (20 Hz – 20 kHz). It shows the gain of a signal over a range of frequencies, where an ideal system has a flat response throughout the entire audio range. The final frequency response for the SXCD analogue output is shown in Figure 5.10. The left channel is shown to be about 0.05 dB higher in gain compared to the right channel, which is insignificant (inaudible to the human ear) and could be due to component tolerances within the output filter circuit.

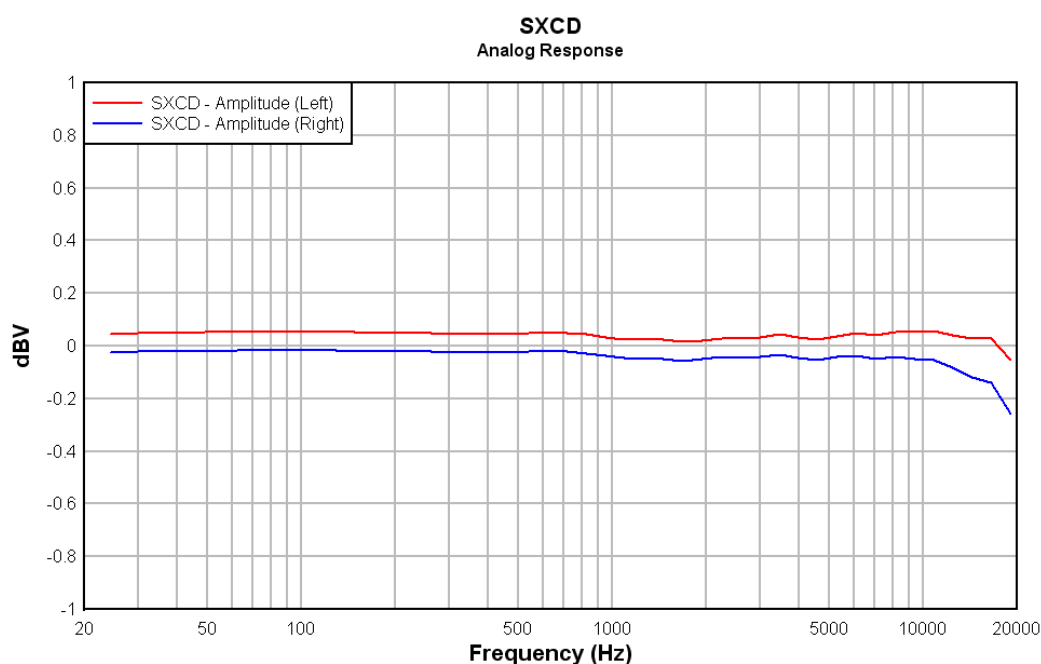


FIGURE 5.10: ANALOGUE FREQUENCY RESPONSE

The main feature of the result is how flat the response is over the audio range, where both channels are flat to within 0.2 dB. A response that is flat within 1 dB is adequate for many products, and no product that was compared in Section 2.1

stated a deviation of less than 0.5 dB as shown in the table in Section 6.2. Comparatively, the SXCD has an extremely flat frequency response (superior to competing products), which minimises any effect the system has on the original audio signal.

5.4.2 TOTAL HARMONIC DISTORTION

The THD is a measure of the harmonic distortion present on the output signal. It is defined as a ratio of the sum of the powers of all the harmonic components to the power of the fundamental signal component. The THD curve calculated by the ATS-2 test equipment is shown in Figure 5.11.

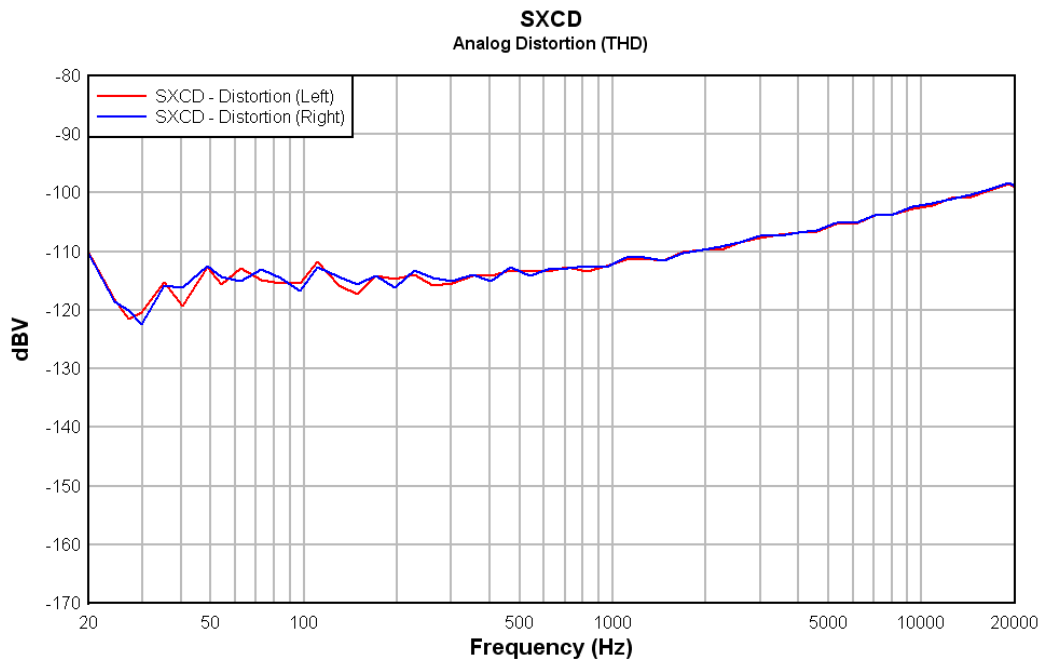


FIGURE 5.11: ANALOGUE TOTAL HARMONIC DISTORTION

$$THD_{dB} = 20 \log \left(\frac{THD_{\%}}{100} \right) \quad \text{Equation 5-1}$$

The graph shows the THD rising in the upper frequencies, with the worst case value of -98 dBV at around 20 kHz. This equates to a percentage of 0.00125 % using Equation 5-1. However a figure for THD is traditionally taken at a frequency of 1 kHz. From the graph, the THD at 1 kHz is -112 dBV, which equates to a percentage of 0.00025 %. This is an extraordinarily low value for

distortion. Comparatively, it is an order of magnitude lower than other products that were investigated, with most products struggling to fall below 0.0025 % THD at 1 kHz.

5.4.3 SIGNAL TO NOISE RATIO

The SNR is the ratio between a reference signal and the background noise and is a measure of how noisy a system is. A typical SNR curve for an audio application increases with frequency (as the noise energy increases).

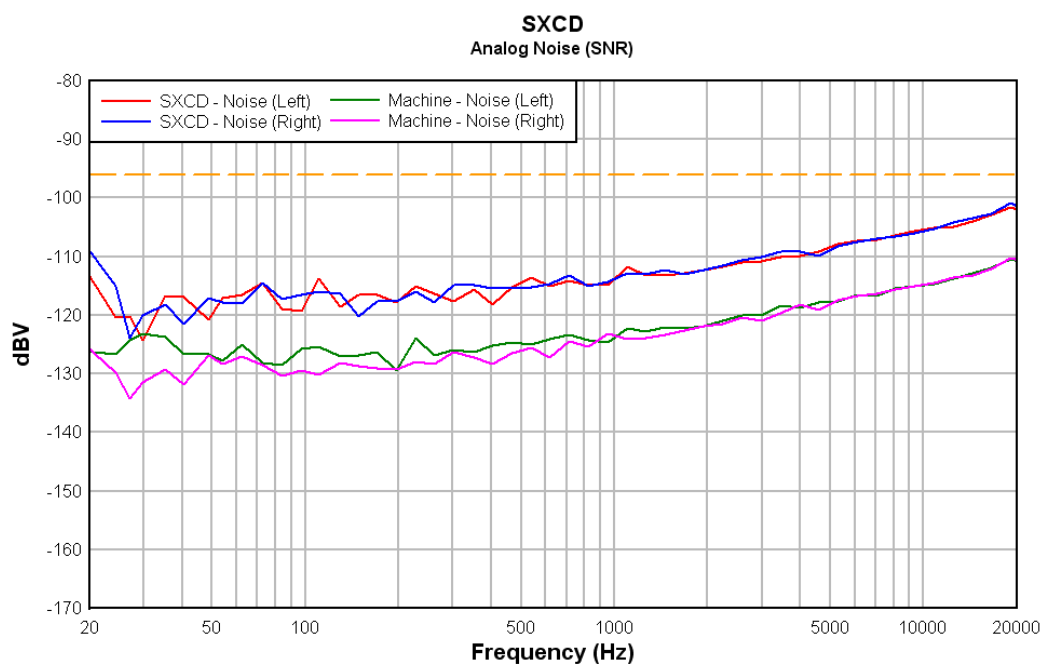


FIGURE 5.12: ANALOGUE SNR WITH CD-DA AND MACHINE LIMITS

The SXCD's SNR for both the left and right channels (red and blue curves) over the audio frequency range is shown in Figure 5.12, where a lower value represents a better SNR. The SNR that is typically stated is the average value over the audio range. From the resulting graph this is approximately -110 dB. However the worst case SNR over the audio range is approximately -102 dBV. It is found that most products state a SNR value of -110 dB, with the average value being -107 dB. Relatively, the SNR of the SXCD is consistent with other products in a similar price range. However, the SNR of the competitive products have not been stated at a particular frequency, nor whether they are worst case or average values, and so a direct comparison is difficult.

Also shown in the resulting SNR graph are two other comparisons; the first being a reference line (yellow dashed line). This reference shows the theoretical maximum SNR possible for an audio disc containing CD-DA audio. This limit comes about because each audio sample is stored using a resolution of 16-bits. This bit-depth of a digital sample results in a maximum SNR governed by the relationship in Equation 5-2, where n is the number of bits. Given the bit-depth of 16-bits for CD-DA audio, the best SNR possible is -96 dB. As shown in the resulting graph, the SNR for the analogue output is below this limit over the entire audio range.

$$SNR = 20\log(2^n) \qquad \text{Equation 5-2}$$

The last comparison shown on the graph is the measured SNR for the left and right channels of the ATS-2 measurement equipment (green and pink lines respectively). This is measured by looping the output of the ATS-2 signal generator to the input of the analyser, and recording the results in the same manner as with the SXCD. The curves sit approximately (and consistently) 10 dB below the results from the CD player. The implications of this are twofold; first, the SXCD results are very close to the noise floor of the ATS-2 equipment. Secondly, the similarities between the two sets of results indicate that the results from the SXCD are influenced by the limits of the measurement system. These results show that in testing of the SXCD the ATS-2 measurement system is not adequate and a more accurate system is required.

5.4.4 MASTER CLOCK JITTER

Any jitter in the master clock signal that drives the DAC stage ultimately results in both distortion of the analogue signal and a higher noise floor in the system. The relationship between jitter and SNR is shown by Equation 5-3 [37], where a particular RMS jitter results in a degradation in noise performance with increasing signal frequencies. The equation can be rearranged and used to find the maximum jitter present in the master clock signal driving the DAC. This is achieved by

plotting the line for the equation with various jitter figures against the SNR graph and choosing the maximum jitter value resulting in a line below the SNR curve.

$$SNR_{\max} = 20 \log(2\pi f_{\text{analog}} t_{\text{jitter}}) \quad \text{Equation 5-3}$$

Because this equation assumes an infinite resolution ADC where jitter is the only factor determining the SNR, t_{jitter} is a value for the maximum jitter present. In reality, many other factors (such as EMI radiation, thermal noise and power supply noise) also contribute to the reduction in the SNR, and therefore the actual jitter value is much less.

By fitting many curves to the SNR of the analogue output, the plot of Equation 5-3 (green dashed line) shown in Figure 5.13 is chosen. It shows that, given the performance of the SNR, the jitter must be less than 70 ps. If more than this were present in the master clock signal driving the DAC stage, then the SNR results would be worse in the upper frequencies.

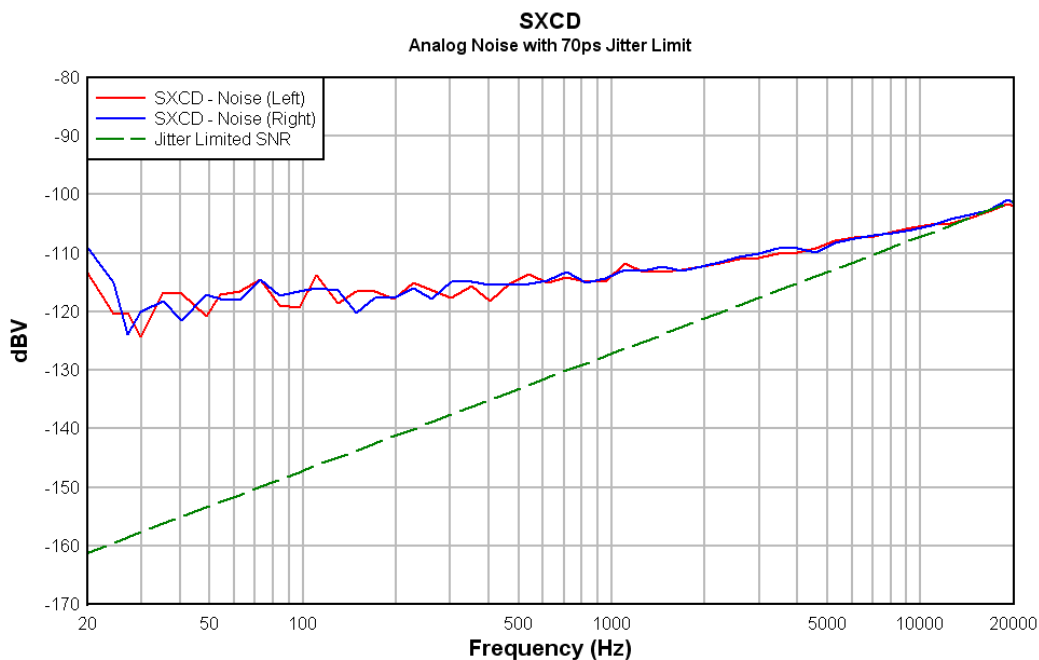


FIGURE 5.13: ANALOGUE SNR WITH 70 PS JITTER LIMIT

This worst case jitter value of 70 ps can be compared with the other products that gave figures for jitter. The lowest value provided being a clock jitter of less than

100 ps (by the Cyrus CD6s), showing that the jitter present in the SXCD master clock is particularly low compared to other products in the target range.

5.5 DIGITAL OUTPUT

The SPDIF output is connected to the ATS-2 test equipment with a coaxial cable, and the ATS-2 receiver configured to provide a termination of 75 ohms. The quality of the digital signal total harmonic distortion and signal to noise ratio can be obtained as with the analogue output. However, due to the digital nature of the transmission and large bandwidth available compared to the audio range, the frequency response is completely flat. For this reason, a frequency response graph is not required. The digital integrity of the signal can also be analysed in terms of the rise and fall times, pulse width, amplitude and jitter.

5.5.1 TOTAL HARMONIC DISTORTION

The total harmonic distortion of the digital output is measured in a similar manner to the analogue output and the resulting graph provided by the ATS-2 test equipment is shown in Figure 5.14.

It is important to note that there are two sets of results shown in this graph. Due to the nature of the digital output, the left and right channels of each set are almost identical. The result is that the left and right channels appear as a single curve for each of the sets of results. The right channel (blue) sits on top of the left channel (red) for the SXCD digital output THD. Similarly, the right channel (pink) sits on top of the left channel (green) for the distortion measured in the ATS-2 equipment.

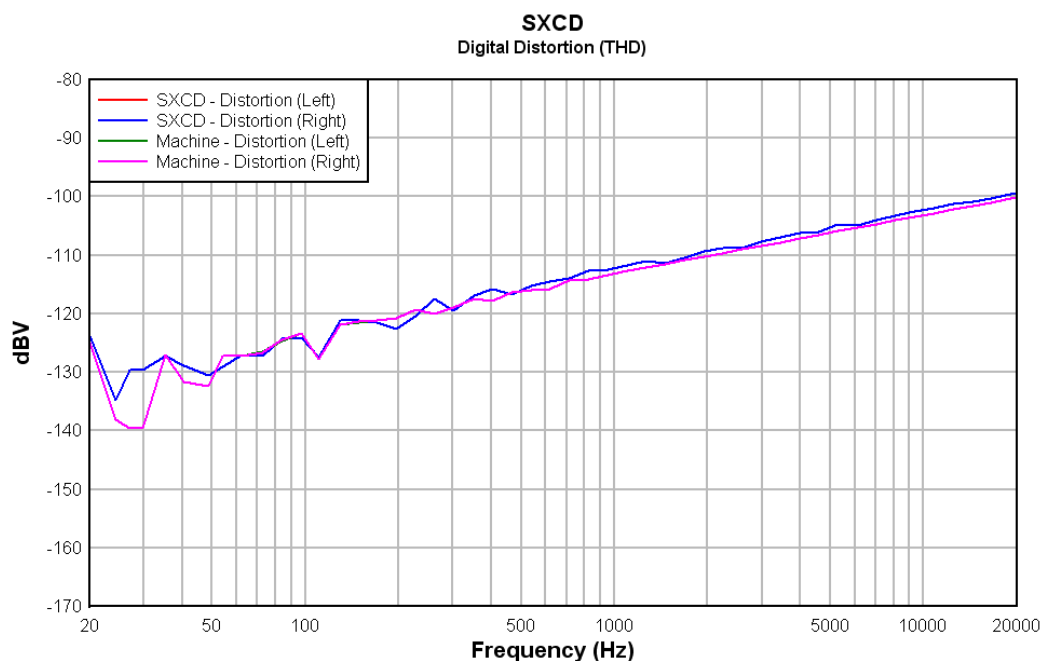


FIGURE 5.14: DIGITAL TOTAL HARMONIC DISTORTION

As with the analogue output, the distortion can be measured at a frequency of 1 kHz to be -112 dBV which equates to 0.00025 %. The maximum THD across the entire audio range is approximately -99 dBV (0.001 %) however, the distortion at lower frequencies is much lower, with a maximum THD below 200 Hz of -120 dBV (0.0001 %).

The digital output results (like the analogue output) are very acceptable by themselves. However, the important features to note from the graph are the values of THD measured when the output of the ATS-2 signal generator is connected directly with the analyser. The two results are shown to be extremely similar, indicating that the distortion of the SXCD digital output is effectively limited by the ATS-2 measuring equipment and will be well below any audible detection limit.

5.5.2 SIGNAL TO NOISE RATIO

The signal to noise ratio of the digital output is also measured, where the graph in Figure 5.15 is produced. Again, it must be noted that this graph contains two sets of results and similarly to the digital THD results, the left and right channels are

practically identical. The SNR result for the final SXCD prototype is shown by the blue curve.

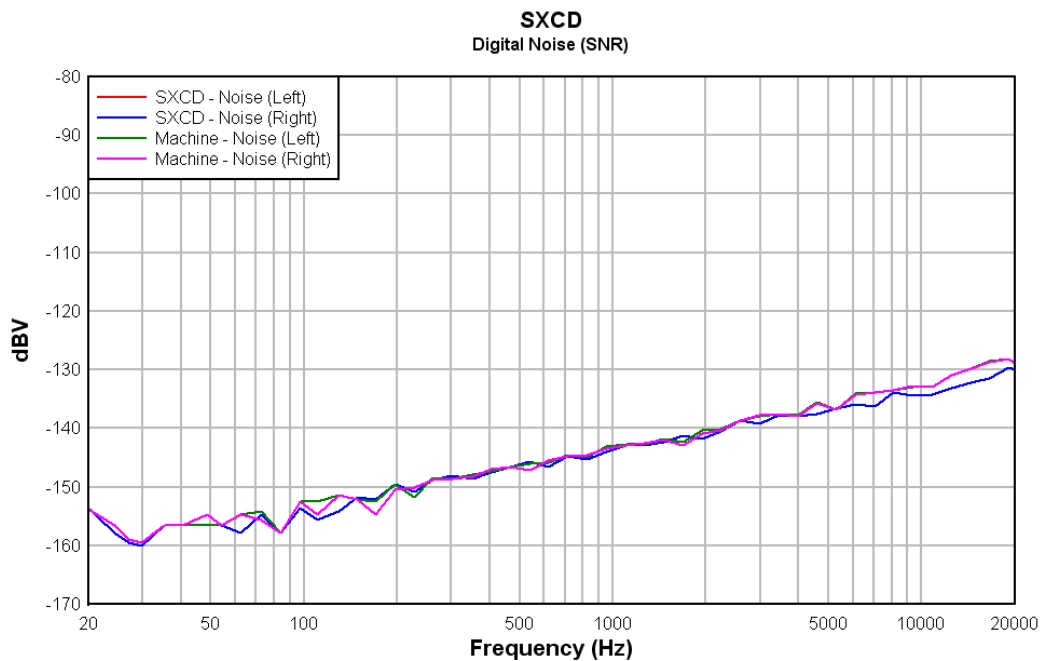


FIGURE 5.15: DIGITAL SIGNAL TO NOISE RATIO

The maximum signal to noise ratio across the audio range is the SNR at 20 kHz, which measures -130 dBV. Again the curve increases with frequency and the average SNR is much lower at below -140 dBV. This figure is extremely low and in fact is again virtually at the SNR limit of the ATS-S measuring equipment.

5.5.3 DIGITAL SIGNAL QUALITY

One method of checking the signal quality of the SPDIF output is to look at an eye diagram of the signal received at the ATS-2 test equipment. The diagram is generated by repetitively sampling the signal and displaying it in the time domain. This gives a good visual indication of timing and level errors that may be present. The minimum height and width between the curves are called the “eye height” and “eye width” respectively.

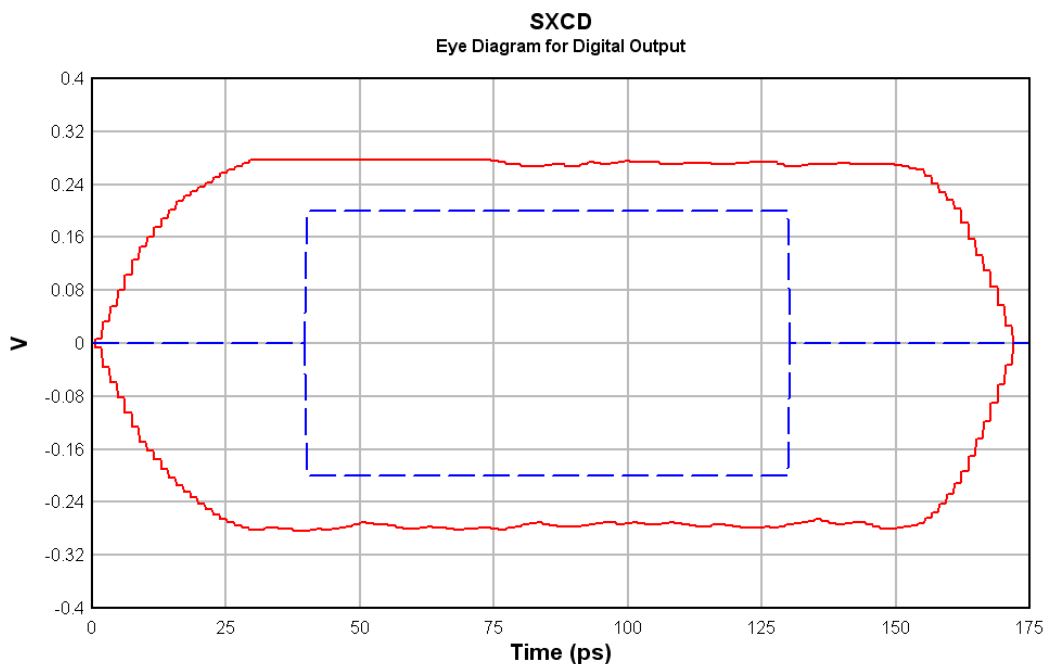


FIGURE 5.16: EYE DIAGRAM FOR THE DIGITAL OUTPUT SIGNAL

The eye diagram provided by the ATS-2 test equipment is shown in Figure 5.16. It is not a comprehensive diagram as it is only showing the minimum values that are measured. This is because the diagram is intended to provide an indication of the signal integrity and show whether the signal is conforming to the Audio Engineering Society AES-3 requirements [9]. As described in Section 3.4, the SPDIF specification states the minimum requirements for an SPDIF signal. The amplitude of the signal at the receiver must be at least ± 200 mV (but no more than ± 300 mV), and the width must be a minimum of 50 % of the nominal width as shown by the blue dashed lines in Figure 5.16. The resulting minimum eye height and eye width both meet the requirements. The rise and fall times can also be read from the eye diagram in Figure 5.16, which range between 12 ns and 25 ns. This is satisfactory as the SPDIF specification states that the rise and fall times must be between 5 ns and 30 ns.

5.5.4 DIGITAL SIGNAL JITTER

As described in Section 2.5.3, the jitter present in a digital SPDIF signal is broken down into two types: deterministic jitter and random jitter. Two types of graphs are produced to measure how much of each type of jitter is present in the signal.

The first is a Fourier transform of the jitter showing the components in the frequency domain. The first graph was produced using the ATS-2 high bandwidth analyser (up to a frequency of 1.4 MHz) and is shown in Figure 5.17.

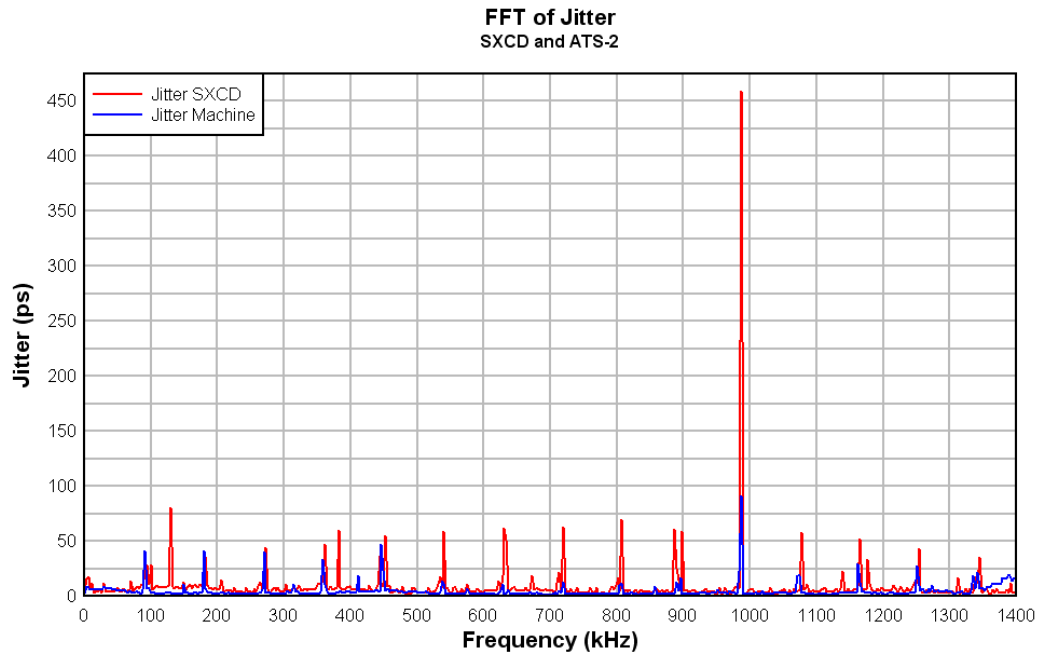


FIGURE 5.17: FFT OF DIGITAL SIGNAL JITTER, HIGH BANDWIDTH

The graph shows that both random jitter and deterministic jitter are present in the signal. The random jitter is spread evenly across the frequency range and is primarily below 10 ps in amplitude. The deterministic jitter consists of one main spike at 990 kHz which is 460 ps in amplitude. The other components are mainly harmonics of this spike spaced at approximately 90 kHz intervals (all below 70 ps). A few other components also exist at 129 kHz (81 ps), 381 kHz (60 ps) and 886 kHz (60 ps).

It is difficult to determine the cause of these last three spikes, as they are outliers and the frequencies at which they occur do not directly relate to any clock frequencies present in the SXCD system. The main component (at 990 kHz) could be explained to some degree by the fact that the ATS-2 equipment also has the same component and harmonics. The jitter present in the ATS-2 equipment is shown by the blue line. Although the component at 990 kHz is much smaller in amplitude than the one measured from the SXCD, it indicates that the cause of the

spike may be inherent in the measurement equipment. This is the case with many of the other spikes apart from the three outliers.

Often, to analyse the audibility of jitter, only the components that are present in the audio range are inspected. In a study performed by Julian Dunn [38] a graph of maximum inaudible jitter amplitude against frequency was derived as shown in Figure 5.18. It shows that at 20 kHz the jitter must be less than 20 ps, increasing at 6 dB per octave for lower frequencies until approximately 500 Hz where the limit is 1 ns. Below 200 Hz the jitter may be up to 800 ns in amplitude before it becomes audible.

A Fourier transform of the jitter in the audio range is shown in Figure 5.19 which is obtained using the low bandwidth analyser in the ATS-2 equipment. It shows that the jitter is below 20 ps over the entire audio range, which is well below this audible limit.

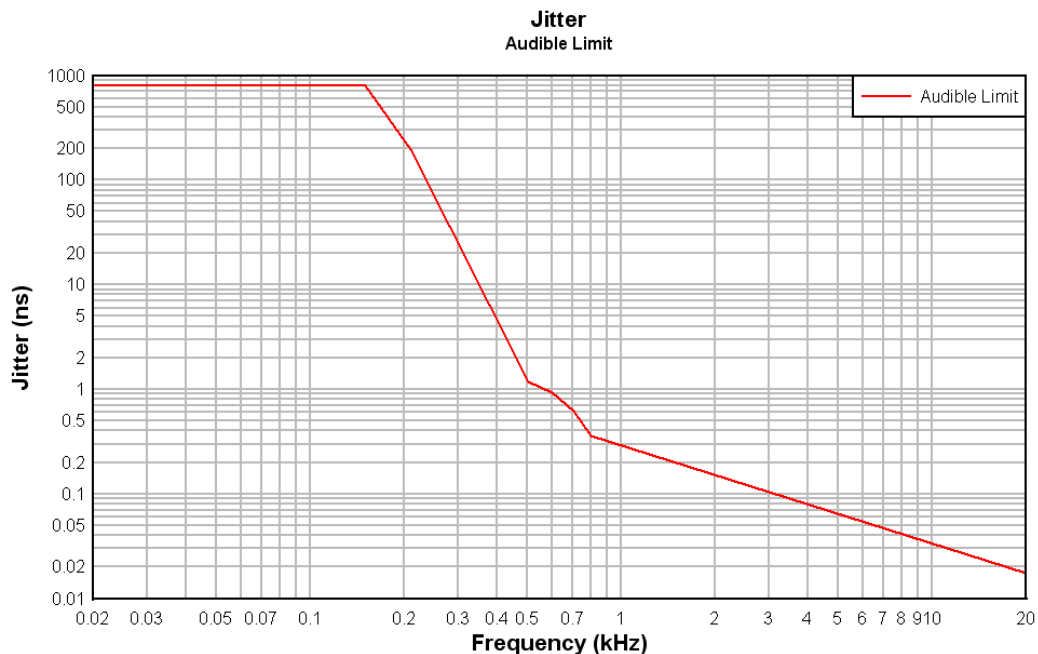


FIGURE 5.18: MAXIMUM INAUDIBLE JITTER AMPLITUDES

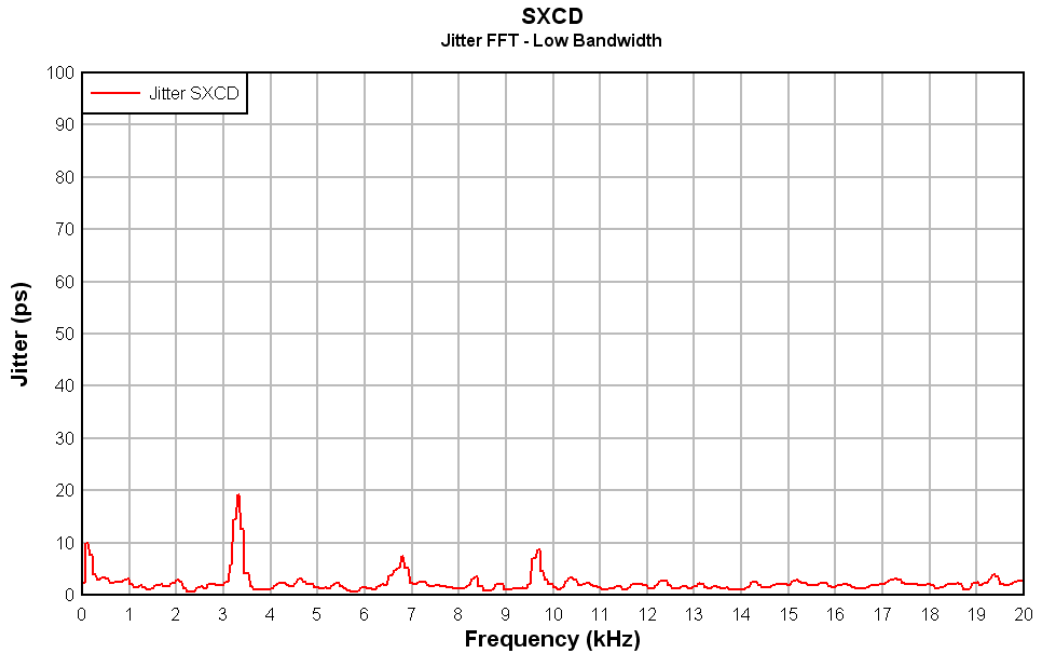


FIGURE 5.19: FFT OF DIGITAL SIGNAL JITTER, LOW BANDWIDTH

The second type of graph produced is a probability curve of the jitter as shown in Figure 5.20. Any random jitter present in the signal results in a Gaussian distribution in the probability curve. However, the curve produced does not have the shape of a Gaussian distribution. It is made up of several peaks which indicate the presence of several constant frequency sources of deterministic jitter.

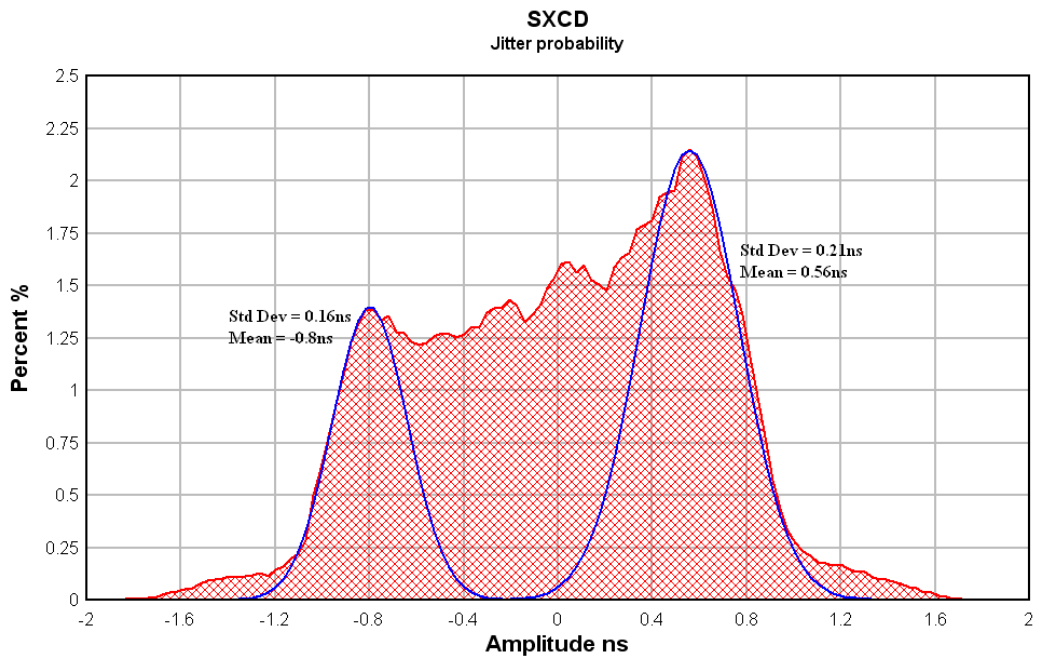


FIGURE 5.20: PROBABILITY CURVE FOR THE DIGITAL SIGNAL JITTER

By using the TailFit™ method [39] (where the tails of the distribution are fitted with a Gaussian curve) the random and deterministic jitter components can be quantified. The method states that the random jitter is given by the Gaussian distributions and the deterministic jitter component is the difference between the means of the two Gaussian curves. From the graph, the average of the two Gaussian standard deviations is 185 ps which is a value for the approximate RMS random jitter component. The deterministic jitter component (difference between the two means) is approximately 1.3 ns pk-pk and is well below the 4.4 ns limit imposed by the AES specification (page 21 of [9]). The analysis above also shows that these figures mainly consist of high frequency components that are outside the audio range. However, these values are difficult to compare as other products do not provide these details.

6 CONCLUSION

6.1 REVIEW

The project to design a high quality CD player (SXCD) emerged due to the need for a source component for the Silhouette series of products. The SXCD would complement the existing Silhouette series of products, providing a potential customer with a complete audio solution. The project involved the entire circuit design and controller firmware development. Other Perreux engineers were responsible for the chassis and front panel design.

It was decided to use a computer optical drive instead of a proprietary mechanism due to cost and minimum order quantity limitations imposed by manufacturers. This decision resulted in a flexible design implementation, where a simple low cost ATmega8515 micro was used as the controller for the system. An on-board DAC was included so the unit can function as a stand-alone CD player and a high quality SPDIF output was included so it could be upgraded with a dedicated DAC module such as the SXD2. The product is interfaced by a circular array of buttons, a remote input was implemented to allow for remote control, and a RJ45 connector provided for future upgrade purposes.

The complete system is synchronised from a single high quality crystal oscillator providing a low jitter clock. Considerable effort was invested in the controller firmware to maximise efficiency and to provide a robust operation. The circuit board was designed to reduce noise in the power supply, clock signals and output circuits so that the output signals (SPDIF and analogue) were not compromised.

6.2 ANALYSIS OF RESULTS

In terms of audio quality, the purpose of a CD player is to reproduce the audio signal (contained in digital form on the CD) whilst having as little affect as possible on the signal itself. Therefore, the specification of an ideal player would have a perfectly flat frequency response, zero THD and an infinitely low SNR.

However, in reality this is an impossible task and so a design engineer can only endeavour to get as close to these ideals as possible. The results for the final revision of the SXCD have demonstrated that the system is able to provide an audio signal with less THD and with a lower SNR than many other products currently available.

6.2.1 SIGNAL TO NOISE RATIO

Typically a digital output will perform very well with low SNR as the data are contained in binary form and an analogue noise element is generally not of any concern until the digital to analogue conversion process. The digital output proves to be of high quality, with a SNR of lower than -130 dB over the entire audio range of 20 Hz to 20 kHz.

The SNR of the analogue output is of similar or better quality than many of the competitive products currently in the market. It is well below the limit of the CD-DA format. However, with a worst case value of -102 dB, the SNR of the analogue output could be improved. The SNRs of the products being compared, along with the SXCD, are shown in Table 6.1.

6.2.2 TOTAL HARMONIC DISTORTION

The quality of the digital SPDIF output is high, with the THD a mere 0.00025 % at 1 kHz. Traditionally the analogue output does not provide the same THD benefits that the digital output does with worse SNR and THD values. For the SXCD, the analogue output is of exceedingly high quality, given the THD is at the same quality as that of the digital. This low 0.00025 % THD is compared with that of the other products both in Table 6.1 and in the form of a bar graph in Figure 6.1. In the graph a lower bar represents a lower THD figure while a higher bar shows a higher THD. The SXCD is shown at the right hand side with the red bar, which is much lower than any other product, and in fact, all other products are more than eight times the distortion of the SXCD.

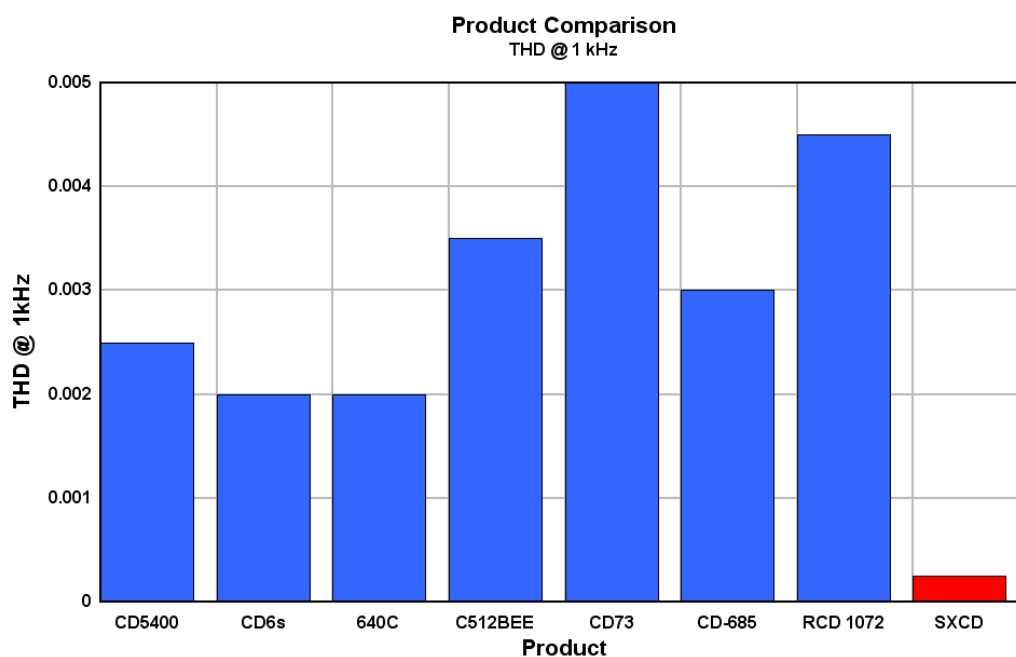


FIGURE 6.1: A COMPARISON OF COMPETITIVE PRODUCTS ANALOGUE THD

6.2.3 OTHER DIGITAL PARAMETERS

The digital output exceeds the required specifications (as outlined in Section 3.4.2). The loaded peak-to-peak amplitude of the signal sits just below the maximum at ± 275 mV, and the ratio of unloaded to loaded signal amplitude of 2:1 shows that the output impedance is correctly matched to the 75 Ohm line impedance. The peak digital signal jitter is only 1.3 ns (refer [9]) while the jitter within the audio range of 20 Hz to 20 kHz is far less than the audible limit.

The effort invested in the oscillator as a master clock source has proven to be beneficial. The result is a master clock with extremely low clock jitter, with less than 70 ps present within the circuit. This value is lower than any of the products that were compared in Table 6.1.

These results are compared with the competitive products that were discussed in Section 2.1 to establish how the design performs compared to the current market. The specifications of each product, along with the final SXCD revision are summarised in Table 6.1. As shown, many of the products have not provided the response deviation or a figure for clock jitter, and the Azur 640C has not provided a value for the SNR.

	Response 20 Hz–20 kHz	THD @ 1 kHz	SNR	Clock Jitter	Price \$NZ
SXCD	0.2dB	0.00025 %	110 dB	<70 ps	~\$1000
Marantz CD5400	-	0.0025 %	110 dB	-	\$599
Cyrus CD6s	-	0.002 %	110 dB	<100 ps	\$3295
Cambridge Audio Azur 640C	1 dB	0.002 %	-	<260 ps	\$899
NAD C521BEE	0.5dB	0.0035 %	108 dB	-	\$499
Arcam Diva CD73	-	0.005 %	112 dB	-	\$1499
Denon CD-685	-	0.003 %	110 dB	-	\$699
Rotel RCD 1072	0.5 dB	0.0045 %	100 dB	-	\$1495

TABLE 6.1: COMPARISON FOR THE CD PLAYERS DISCUSSED IN SECTION 2.1

Overall, the results have been extremely promising and have set a benchmark for future CD player developments at Perreux. Many of the results have shown to be at the limit of the available measurement equipment capabilities indicating exceptionally high quality performance.

6.3 CIRCUIT OPERATION

The controller firmware was customised to provide the required functionality and performance. The speed of operation as mentioned in Section 5.2 is certainly acceptable, and the mechanical noise level created by the actual CD drive was limited to a desirable level by limiting the CD spin speed. This resulted in great approval from Perreux as there was an initial concern regarding the level of mechanical noise generated by the high spin speeds.

The final revision is shown in its most complete form at the conclusion of the project in Figure 6.2. The look of the final product and CD drive tray will be a result of Perreux's mechanical design of the front panel and chassis.

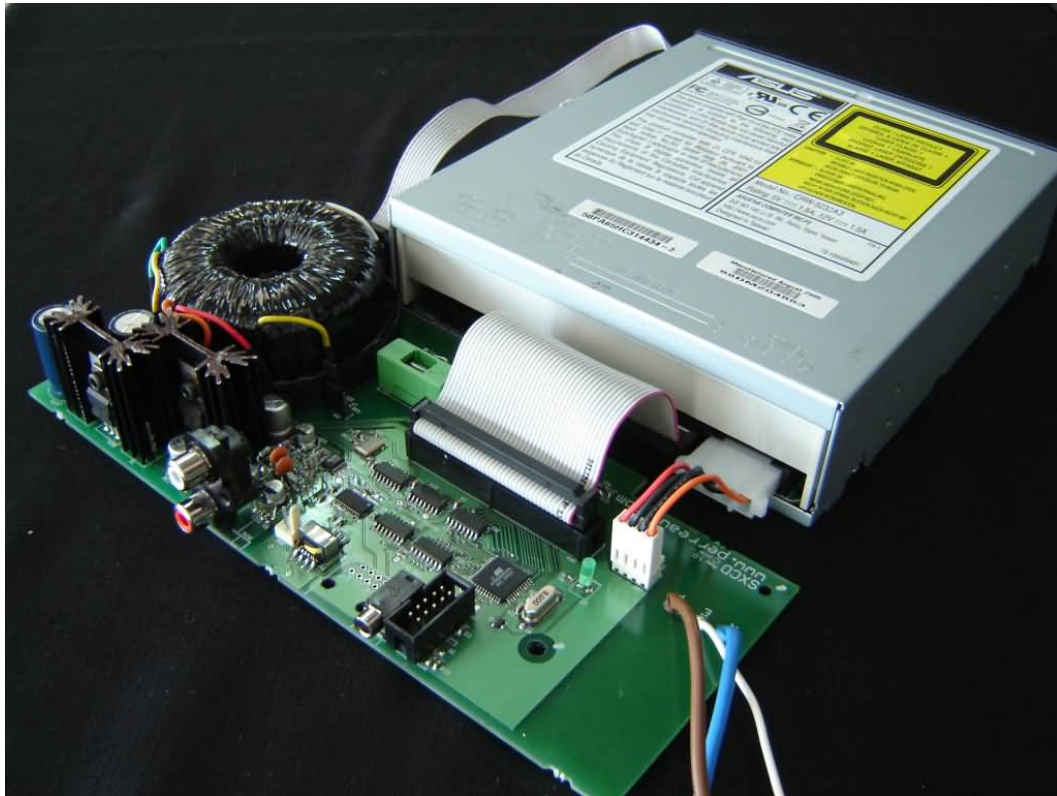


FIGURE 6.2: COMPLETED PROTOTYPE FROM REAR

6.4 FUTURE IMPROVEMENTS

The concept of using a common computer CD drive as the mechanism for a CD player is a versatile one. The current project can be easily upgraded to a larger product with more functionality. This is the exact intent of Perreux, where another model will be designed based on this project. The new model would include more functionality such as a front panel display, a remote control, an optical digital output and extra features (such as random and repeat play). However, when such a product is designed, it would be wise to upgrade the micro controller to the faster 16 MHz version, or maybe a new part altogether which provides more functionality. The ATmega8515 used in this project is fully utilised, and would not be capable of providing any more functionality.

Areas do exist where more work can be invested to enhance the performance of this product. The quality of the digital SPDIF output signal is satisfactory, and the signal jitter within the frequency range is not audibly significant. However, the high frequency components of deterministic jitter that are present could be better understood and possibly eliminated. The physical size limitations imposed in this

project did have an effect on the PCB layout and circuit design. If the project was to be developed further (or into a larger product) it would be wise to eliminate these limitations. This would enable circuit enhancements such as larger heatsinks on the power supply regulators for improved heat dissipation, more power supply smoothing capacitors and a larger transformer. Such adjustments would provide a more regulated supply which would improve the performance of the analogue output.

Research has proven that mechanical vibrations within a chassis can have a negative effect on an analogue signal [40]. The vibrations can generate electrical signals within a system which can result in the distortion of the audio output. In the future, this area could be investigated, possibly increasing the quality and performance of the analogue output.

6.5 SUMMARY

The requirements set out in Chapter 1.4 were that the product be designed with both analogue and digital outputs. The analogue output was intended to be inexpensive and of moderate quality (below 0.003 % THD) so that the SXCD could be used as a stand alone CD player. The digital output was to be of high quality (below 0.001 % THD) to provide superior performance by upgrading the system with a dedicated DAC module. A simple user interface was requested with the addition of a remote input, whilst maintaining the Silhouette series theme and profile.

The final project revision exceeded all of these initial requirements. The analogue output quality exceeded all expectations whilst maintaining a simple and low cost design. Both the analogue and digital outputs surpass the quality of many of the competitive products in the market.

Overall the project was not only a great success, but it was also an immense learning curve for both parties involved. It has been shown that it did not require a huge budget, a large design team or a complicated implementation to produce a good quality and competitive design. Both the digital and analogue outputs of this

product are in contention with many of the products currently on the market, with the analogue output far exceeding the initial product specifications.

The success of this project can best be expressed by the following quote from the managing director of Perreux Industries Ltd (a full letter of recommendation is included in Appendix D):

“We are extremely satisfied with Carl's project. Being of a complex technical nature it is doubtful that our company would have had the time or resources to attempt such an ambitious undertaking. Not only has Carl satisfied our brief, he has brought it through several iterations to a fully prepared production version. In our opinion he has exceeded all expectations of a Master of Engineering student and we wish to express our sincere thanks to him for his excellent work.”

- Martin Van Rooyen

7 REFERENCES

- [1] Perreux Industries Ltd, URL: www.perreux.com last accessed 05/05/2006
- [2] Perreux Industries Ltd, "Silhouette Series", URL: www.perreux.com/seroverview.php?idcat=4 last accessed 23/04/2006
- [3] Cyrus Audio, "CD 6s", URL: www.cyrusaudio.com/product.asp?ProductID=23 last accessed 24/05/2006
- [4] Marantz, "CD5400", URL: www.marantz.com/new/index.cfm?fuseaction=Products.ArchProduct&cont=eu&bus=hf&prod_id=2471&type=cd last accessed 24/05/2006
- [5] M. Jones, "Three CD Players", *AudioEnz Magazine*, 2005, URL: www.audioenz.co.nz/2005/affordablecd.shtml last accessed 25/05/2006
- [6] Cambridge Audio, "Azur 640C", 1999, URL: [www.cambridgeaudio.com/summary.php?PID=99&Title=Azur+640C+\(Version+2\)](http://www.cambridgeaudio.com/summary.php?PID=99&Title=Azur+640C+(Version+2)) last accessed 25/05/2006
- [7] NAD, "C521BEE", 2003, URL: www.nadelectronics.com/cd_players/C521BEE_framset.htm last accessed 25/05/2006
- [8] Daisy Laser, "L1210 Audio-dedicated tray-loading CD module", URL: www.daisy-laser.com/products/CD/modules/L121065/1121065.htm last accessed 16/09/2005
- [9] Audio Engineering Society Inc, "REVISED AES standard for digital audio", *revision of AES-3 1992*, pp. 29, 2003, www.mtsu.edu/~jakins/4200/aes3-2003.pdf
- [10] LessLoss Audio Devices, "Types of Digital Configurations in DACs", 2004, URL: www.lessloss.com/types.html last accessed 26/01/2006

- [11] Atmel, "ATmega8515 Microcontroller", URL:
www.atmel.com/dyn/products/product_card.asp?part_id=2006 last
accessed 08/11/2005

- [12] Texas Instruments, "DIT4192 Digital Audio Transmitter", 2003, URL:
<http://focus.ti.com/docs/prod/folders/print/dit4192.html> last accessed
07/04/2006

- [13] Scientific Conversion Inc, "SC982 Pulse Transformer", 2005, URL:
www.scientificconversion.com last accessed 12/04/2006

- [14] Cirrus Logic, "192 kHz Digital Audio Interface Transmitter", 2005,
http://www.cirrus.com/en/pubs/proDatasheet/CS8406_F3.pdf

- [15] Wikipedia, "SPDIF", URL: <http://en.wikipedia.org/wiki/SPDIF> last
accessed 06/04/2006

- [16] J. D. Paul, "The Effects and Reduction of Common-Mode Noise and
Electromagnetic Interference in High Resolution Digital Audio
Transmission Systems", in *Proceedings of the 115th Audio Engineering
Society Convention, New York, 10-13 October 2003*,

- [17] Texas Instruments, "Filter Pro", 2005, URL:
<http://focus.ti.com/docs/toolsw/folders/print/filterpro.html> last accessed
03/03/2006

- [18] Philips Semiconductors, "I2S bus specification", pp. 7, 1996,
www.semiconductors.philips.com/acrobat_download/various/I2SBUS.pdf

- [19] ST Microelectronics, "M74HC165 8 bit PSIO Shift Register", pp. 12,
2001, www.st.com

- [20] CMAC, "CFPS-72, 73", 2005, URL:
www.cmac.com/mt/databook/oscillators/smd/spxo/cfps-72,-73.pdf last
accessed 18/05/2006

-
- [21] R. Poore, "Phase Noise and Jitter", pp. 7, 2001, URL:
http://eesof.tm.agilent.com/pdf/jitter_phasenoise.pdf last accessed
29/05/2006
- [22] Ralton, "Convert SSB phase noise to jitter", URL:
www.raltron.com/cust/tools/osc.asp last accessed 16/05/2006
- [23] Toroid International, 2006, URL: www.toroid.se/english/index.html last
accessed 17/04/2006
- [24] ST Microelectronics, "L7800 Series Positive Voltage Regulators", *Rev. 12*,
pp. 34, 2004, www.st.com
- [25] Aavid Thermalloy, "TO220 Heatsink", 2006, URL:
www.aavidthermalloy.com/cgi-bin/stdisp.pl?Pnum=531002b02500g last
accessed 20/04/2006
- [26] SB Projects, "Philips RC5 Protocol", 2005, URL:
www.xs4all.nl/~sbp/knowledge/ir/rc5.htm last accessed 18/05/2006
- [27] Wikipedia, "Manchester Code", URL:
http://en.wikipedia.org/wiki/Manchester_encoding last accessed
10/04/2006
- [28] M. I. Montrose, "Right Angle Corners on PCB Traces", *Montrose
Compliance Services, Inc. 2352 Mission Glen Dr. Santa Clara, CA 95051-
1214, USA*, 1999,
- [29] Munro Cable Fabrication LTD, *Unit 23, 298 Nelson St, Te Papapa,
Auckland, NZ*, 2006,
- [30] Wikipedia, "C programming language", URL:
http://en.wikipedia.org/wiki/C_programming_language last accessed
21/04/2006
- [31] Wikipedia, "Assembly language", URL:
http://en.wikipedia.org/wiki/Assembly_language last accessed 21/04/2006

- [32] Wikipedia, "Event-driven programming", URL:
http://en.wikipedia.org/wiki/Event-driven_programming last accessed
03/04/2006
- [33] T13.org, "AT Attachment with Packet Interface - 6 (ATA/ATAPI-6)", pp.
415, 2000, www.t13.org
- [34] Small Form Factor Committee, "ATA Packet Interface for CD-ROMs
SFF-8020i", pp. 224, 1996, www.t13.org
- [35] Audio Precision, "ATS-2 Audio Analyzer", 2005, URL:
[www.audioprecision.com/index.php?page=products&category=&id=1000
001075](http://www.audioprecision.com/index.php?page=products&category=&id=1000001075) last accessed 22/05/2006
- [36] Audio Precision, "Compact Disc Player Tests", *Application Note #1*, vol.
1.0, 1998,
[www.audioprecision.com/index.php?page=support&category=&id=10000
01047](http://www.audioprecision.com/index.php?page=support&category=&id=1000001047)
- [37] B. Brannon, "Understand the effects of clock jitter and phase noise on
sampled systems", pp. 87-96, 2004,
- [38] J. N. Dunn, "Considerations for Interfacing Digital Audio Equipment to
the Standards AES-3, AES-5, AES-11", *Published in "Images of Audio",
the Proceedings of the 10th International AES conference, London,
September 1991*, pp. 115-126,
- [39] M. P. Li, J. Wilstrup, R. Jessen, and D. Petrich, "A New Method for Jitter
Decomposition Through Its Distribution Tail Fitting", *ITC Proceeding*,
1999,
- [40] Barry Kohan, "Ultimate Isolation Theory", 2002, URL:
www.brightstaraudio.com/technical_talk.htm last accessed 02/06/2006

8 ACRONYMS

ATA

The Advanced Technology Attachment (ATA) is a standard hardware interface for connecting storage devices such as hard disks and CDROM drives inside a computer. An earlier version of the interface was dubbed Integrated Drive Electronics (IDE) due to the controller being contained in the drive itself instead of having a separate controller on the motherboard. The ATA interface is still frequently called IDE which is incorrect. A new serial form of the ATA interface has been developed, called Serial ATA (SATA), which has led to ATA being retroactively renamed to Parallel ATA (PATA). SATA offers a lower number of connectors and faster transfer speeds.

ATAPI

The ATA Packet Interface defines a more general purpose interface than the ATA task file and was developed for use with removable media drives. It uses the ATA hardware interface at the physical level, but includes the necessary extra commands for interfacing devices other than hard disk drives and issues the commands through the use of a packet mode.

CD-DA

Compact Disc - Digital Audio (CD-DA) is the standardised medium for recording digital/audio information. The Red book (the standard for audio CD's) defines CD-DA media. The Red book is named after one of a set of colour-bound books that contain the technical specifications for all CD and CDROM formats.

I²S

The Inter-IC Sound (I2S) is a serial bus protocol developed by Philips for transmitting digital audio data within different parts of a circuit. The bus consists of 3 wires; a serial data line to carry the audio data, a clock line used to clock in

the data, and a word select line used to synchronise the receiver to the left and right samples. Because the data and clock signals are separated, time-related errors such as jitter do not occur.

MFB

The Multiple Feedback (MFB) filter is a two pole filter topology offering low sensitivity to component variation. Although a MFB filter contains one more passive component than a Sallen-Key topology, the stop band rejection is better, making this topology more attractive to low cost solutions.

MIPS

Spoken as one word, “MIPS” stands for Million Instructions Per Second. It is usually used in describing the speed of a micro processor such as: “The Atmel ATmega8515 can run up to a speed of 16 MIPS from a 16 MHz crystal”, meaning that it can execute 16 million (16,000,000) single cycle instructions per second.

MSF

MSF stand for Minutes, Seconds, Frames, and is one of the formats used for indexing a position on an audio CD. If a MSF address is used, the desired position is stored as the number of Minutes, Seconds, and Frames from the start of the CD. One Minute unit consists of 60 Second units, one Second unit consists of 75 Frame units, and one Frame unit consists of 2352 bytes of digital audio data. Therefore, one 74 minute audio CD has a capacity of 746.9 MB. A data CD only uses 2048 bytes per Frame unit of the available 2352 bytes, and so a 74 minute data CD has a capacity of 650.4 MB.

RAM

Random Access Memory (RAM) refers to a data storage format that allows the data to be accessed in any order. RAM is the most common form of computer memory, and the contents of the memory are typically lost in the absence of power.

SNR

Signal to Noise Ratio (SNR) is a measure of the power ratio between a reference signal and the background noise. SNR is often expressed in terms of the logarithmic decibel (dB) scale. However, because SNR (and the decibel scale) is a relative term, the level of the reference signal must be stated. In this project, it is expressed as dBV, which is a decibel scale relative to a 1 V RMS reference signal.

SPDIF

SPDIF stands for Sony/Philips Digital Interface Format. It is a 1-wire protocol for transmitting pulse code modulated digital audio data multiplexed with user data over a single signal wire. SPDIF is the consumer version of the standard known as AES/EBU and it contains small differences in the protocol and requires less expensive hardware. The SPDIF format is primarily used on CD players, but has become common on other audio components such as computer sound cards.

THD

The Total Harmonic Distortion (THD) of a signal is a measure of the harmonic distortion present and is defined as the ratio of the sum of the powers of all harmonic components to the power of the fundamental. THD is calculated as follows:

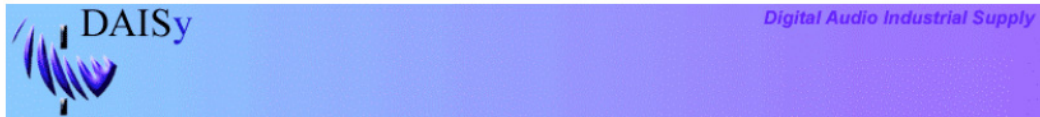
$$THD = \frac{\sum \text{harmonic powers}}{\text{fundamental frequency power}} = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + \dots + V_n^2}}{V_1}$$

TOC

The Table of Contents (TOC) on a compact disc contains information on the type of disc and the starting address of the tracks. This information is encoded in the Q sub-channel in an area reserved at the start of the disc called the lean-in area.

9 APPENDICES

9.1 APPENDIX A - PHILIPS L1210 QUOTATION:



Z.A.C. de la Baudinière
Rue de la Baudinière
44470 Thouare sur Loire
FRANCE

CUSTOMER : PERREAUX INDUSTRIES
NEW ZEALAND
Att: Carl Benton

Date : 23/08/2005

Quotation Number : - PERREAUX1-0805/1

Subject : PHILIPS/ L1210.

Dear Mr Benton,

Thank you very much for your inquiry. We are pleased to quote, according to our general terms of sales and supplies, as follows :

L1210 module:

The price of this module is based on qty per batch.

Qty per batch	Price in USD
30pcs	49.50
90pcs	46.30
240pcs and above	43.85

Minimum order qty : 30pcs
Multiple order Qty: 10pcs.
Sample price : as for 30pcs +8%.

Delivery :

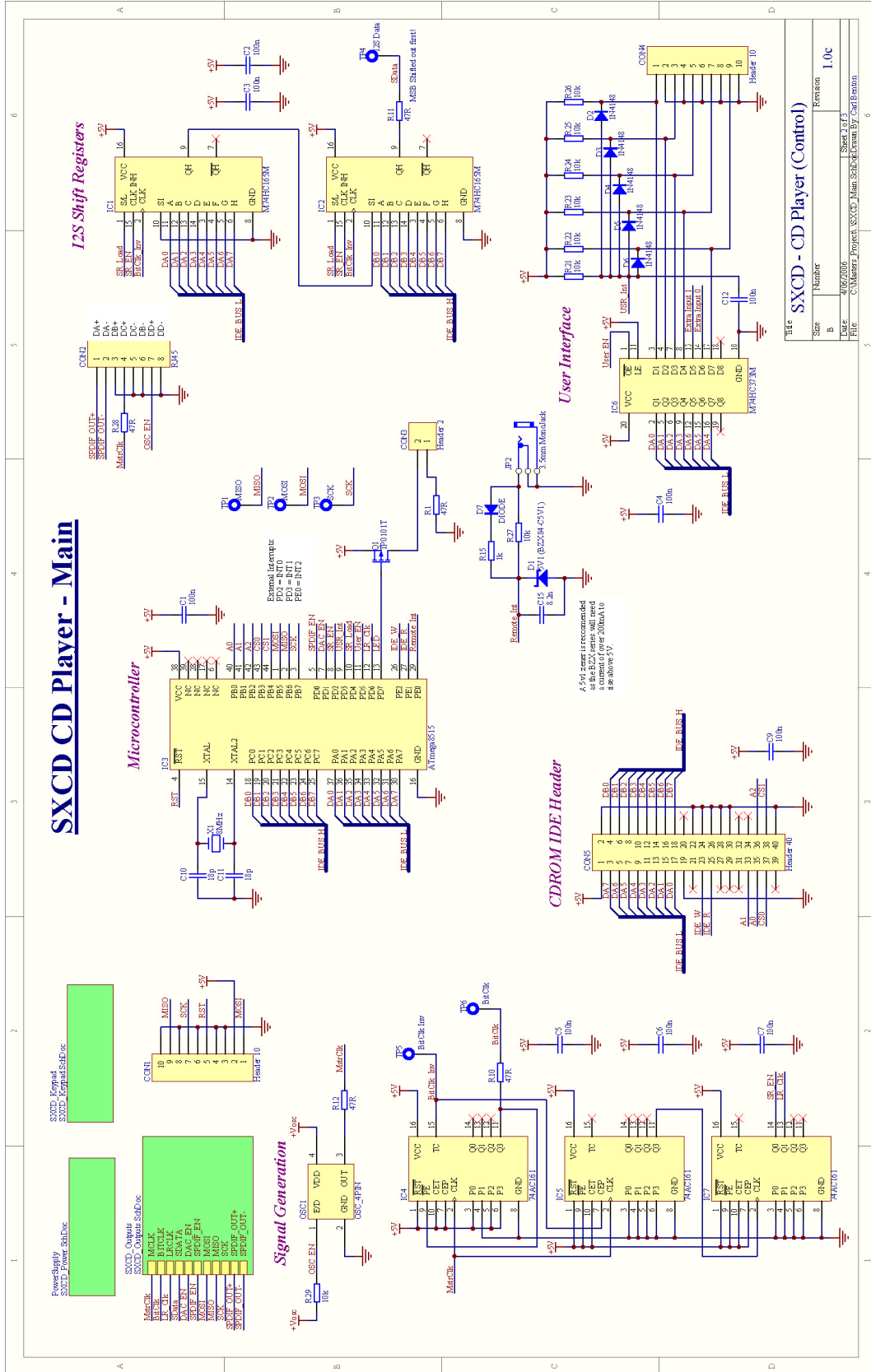
Ex-Works Thouare sur Loire, France.
Parts are sent via UPS Expedited.
If you want to collect parts using your own forwarder, please mention it on purchase order.

Payment terms :

Prepayment by bank transfer, Irrevocable letter of credit or Bank guarantee.

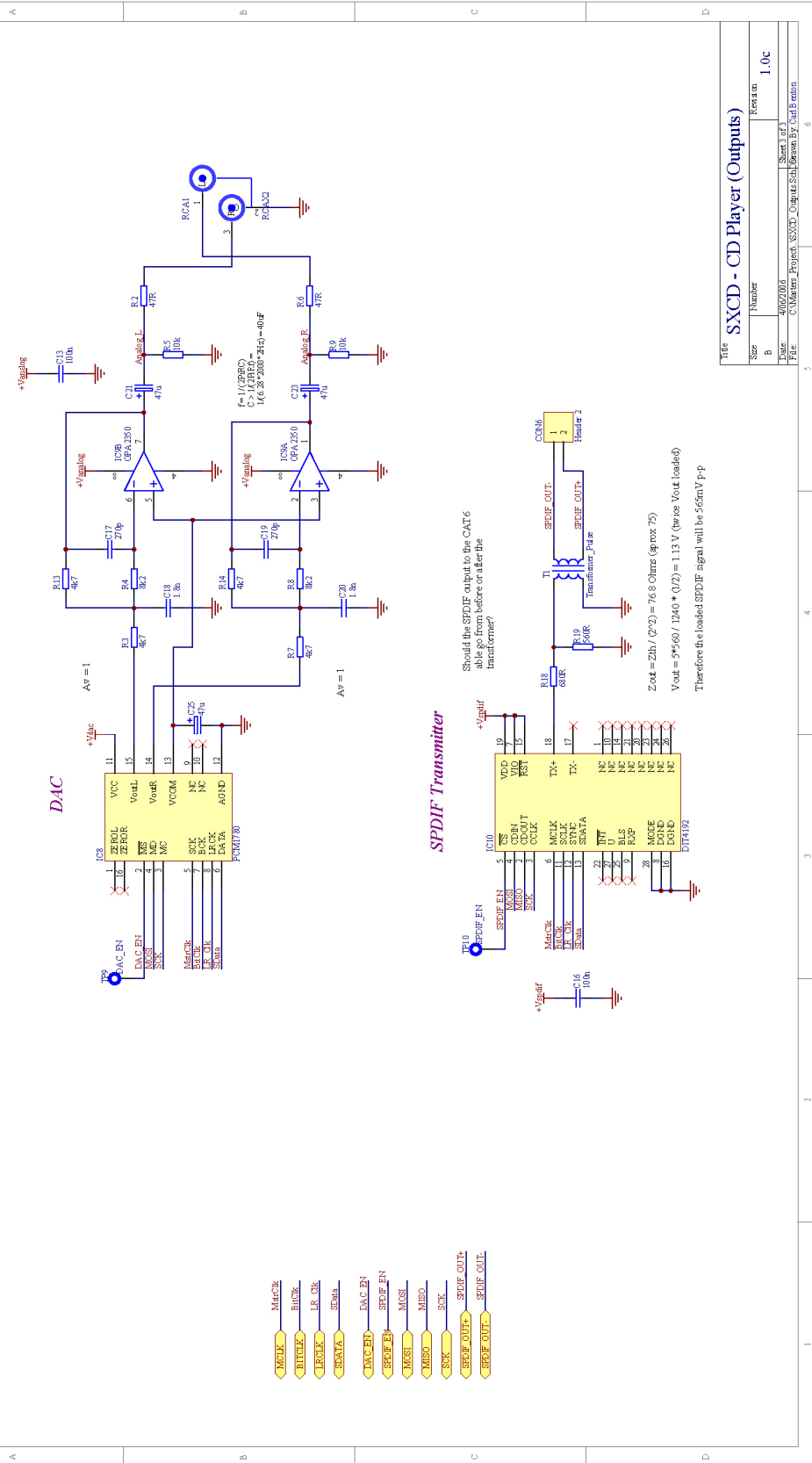
9.2 APPENDIX B - SCHEMATICS

9.2.1 MAIN SCHEMATIC



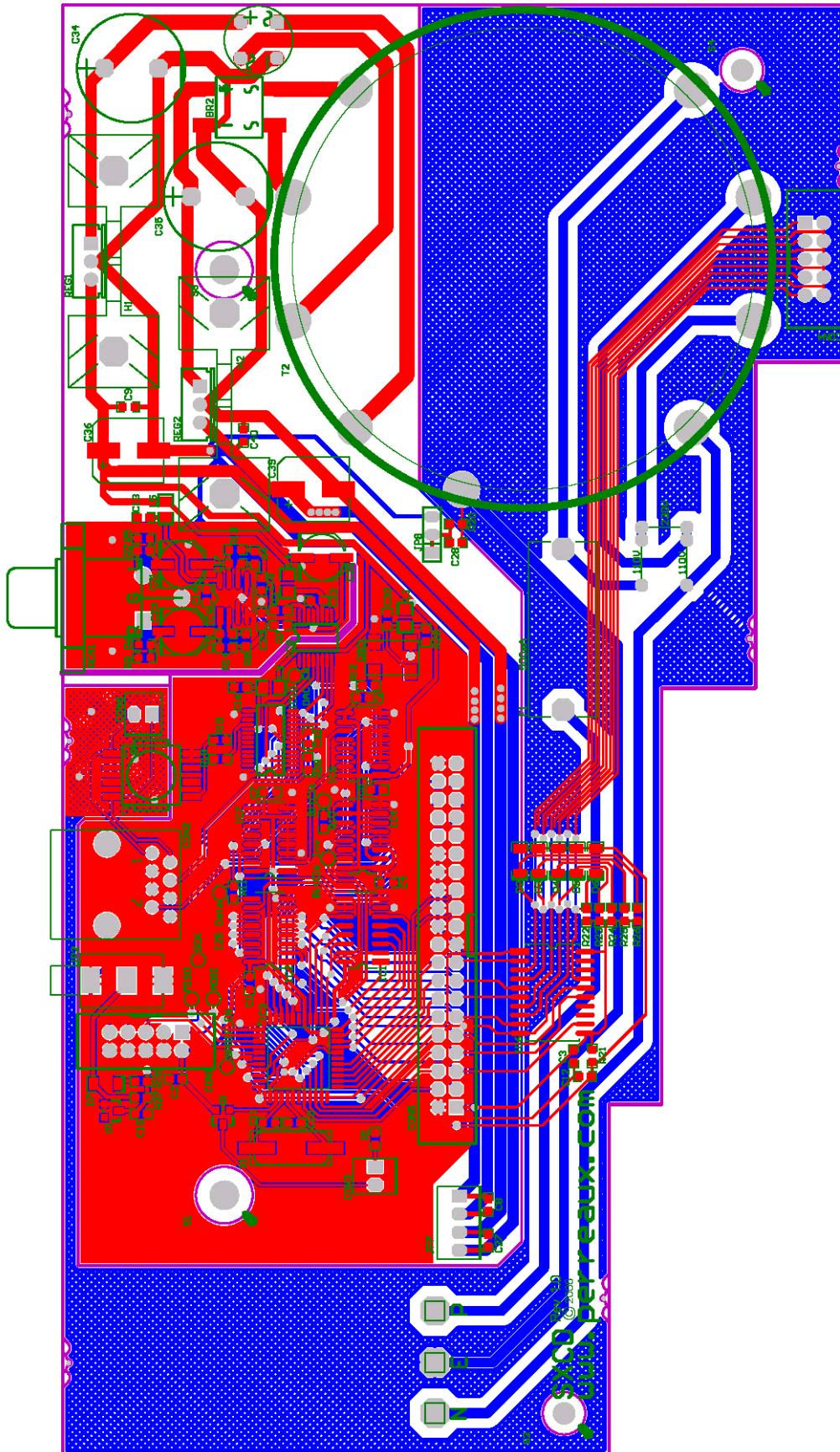
9.2.3 OUTPUTS

SXCD CD Player - Outputs



Title	
Size	Number
b	Revision
Date	Sheet 1 of
File	C:\MyDrawings\Projects\SXCD_CD_Player_Sch\Outputs_Sch1B.dwg

9.3 APPENDIX C – PCB LAYOUT



9.4 APPENDIX D – LETTER OF RECOMMENDATION



PERREAUX INDUSTRIES LTD

9a Gladstone Rd, Mosgiel, Dunedin 9024, New Zealand
PO Box 305, Mosgiel, Dunedin 9053, New Zealand
Phone: +64 3 489 2975 • Fax: +64 3 489 2976
Email: info@perreaux.com • Web: www.perreaux.com

To whom it may concern

08 June 2006

RE: Carl Benton Master of Engineering Research Project

Carl Benton has just completed the development and pre-production designs of a new CD player design for our company. Perreaux is a long established and reputable international niche manufacturer and exporter of high end audio componentry for domestic applications. Our lineup includes integrated, pre and power stereo and multi-channel amplifiers, digital to analogue converters, CD players and other peripheral products. Our expertise centers primarily on the design of and production of a wide variety amplifiers.

Our international clients have a strong preference for suppliers that can offer a complete electronic audio solution. The one weakness in our current lineup is our CD player offering which is not actually manufactured by our company. Lacking expertise in this area, we have up until this point found it easier to simply strip down, repackage and re-badge an existing product.

This has proved unsatisfactory and we are very excited through Carl's project to shortly be able to offer an original engineering solution that befits the Perreaux brand.

Carl took our design brief, developed our concept to a full working prototype design. Mechanical drawings have been produced and we anticipate the commercial release to be in July 2006. Carl was able to work alongside our staff and contractors to coordinate them to achieve a satisfactory result. The responsibility now rests with us to ensure that our mechanical sub-contractors produce a solution that fits the finished electronic design. Prototype mechanical versions look very promising.

We are extremely satisfied with Carl's project. Being of a complex technical nature it is doubtful that our company would have had the time or resources to attempt such an ambitious undertaking. Not only has Carl satisfied our brief, he has brought it through several iterations to a fully prepared production version. In our opinion he has exceeded all expectations of a Master of Engineering student and we wish to express our sincere thanks to him for his excellent work.

Yours sincerely

Martin van Rooyen

Managing Director
Perreaux Industries Ltd
Ph: +64 3 489 2975
Fax: +64 3 489 2976
E-Mail: martin@perreaux.com
Web: www.perreaux.com

9.5 APPENDIX E – PROJECT CD CONTENTS

- AUDIO PRECISION DOCUMENTS
- AUDIO TEST FILES
 - AUDIO PRECISION
 - CUSTOM
- CODEVISION FILES
- DATASHEETS
 - TRANSFORMER
- INITIAL MECHANICAL DRAWINGS
- PHOTOS
- PROTEL FILES
 - SXCD 0.1
 - SXCD 1.0B
 - SXCD 1.0C
 - SXCD 1.0D
- RESULTS FILES
- SPECIFICATIONS
- TIF DOCUMENTS