USB Power Delivery PHY for Domestic Applications

Liam McSherry
40313040

Submitted in partial fulfilment of the requirements for a
B.Eng (Hons) in Electronic and Electrical Engineering

Supervised by
Jay Hoy

School of Engineering and the Built Environment
Edinburgh Napier University

March 2020

Abstract

The USB Type-C connector is an increasingly common choice for devices of all kinds and, while it enables transmission of up to 100 watts, a consumer must still carry a mains power adapter wherever they wish to power a Type-C device. This negatively impacts convenience and, with many devices bundling an adapter, is a contributor to an increase in electronic waste. This report considers the design of a physical layer (PHY) for a USB Power Delivery plug socket which, integrated into a home, could replace many disparate mains power adapters.

In this report it is demonstrated that a semi-discrete PHY which supports both legacy and current revisions of USB Power Delivery can offer a cost-competitive alternative to monolithic PHYs supporting only current revisions, with a unit cost only 17% greater than that of a monolithic PHY. The design of the PHY is partially verified by simulation, and prototype-derived estimations indicate that the area the semi-discrete solution demands is less than half that available on the front face of a BS 4662 standard plug socket. Overall, it is concluded that a design of this kind is both economically and technologically viable.

USB Power Delivery PHY for Domestic Applications

A report, submitted in partial fulfilment of the requirements for a bachelor's degree with honours in electronic and electrical engineering, on the theory, design, and construction of a physical layer for a USB Power Delivery plug socket.

Set in IBM Plex and Rasmus Andersson's Inter.

With acknowledgement to Jay Hoy, for his guidance and supervision, and to Jim Gordon, for his prompt and high-quality work in assembling prototype hardware.

Background

The proliferation of the Universal Serial Bus (USB) has fundamentally changed how consumers interact with their computers. Although becoming more complex as it has developed, the largely singular connector for power and data and the simplification of device set-up has greatly expanded the functionality available to the consumer whilst maintaining a consistent level of convenience.

As this expansion took place, the demands of the consumer and the devices they used also expanded. However, while rates of data transfer have improved more than exponentially since USB's release in 1996, the basic level of power available to a USB device has always increased more conservatively. In fact, that basic level remained at 2.5 watts from 1996 until the 2008 publication of the USB 3.0 specification, which provided for an increase to 4.5 watts. In the time between, likely prompted by the development of the smartphone and the *de facto* standards to enable faster charging which consequently appeared, a USB Battery Charging Specification was published to provide a formalised method of accessing up to 7.5 watts of power, depending on the capabilities of the charger.

In 2012, recognising USB's shift from an interface used for data first to one used for power first, the USB Promoter Group published the USB Power Delivery Specification (USB-PD), which supports the provision of power up to 100 watts in either direction. The Promoter Group intended to obviate the need for external power supplies and enable devices which also required a data interface—such as printers, displays, and hard disk drives—to consolidate onto a single interface.

With this power-first approach, USB is increasingly becoming the all-purpose interface for portable computing. For example, a laptop might now use USB to provide a video signal to a monitor that includes a USB-PD controller and a USB hub, charging the laptop and providing access to a keyboard and mouse at the same time. Once finished, the laptop's user would need only to unplug a single cable. If this consolidation continues and USB becomes even more common, it will likely become desirable to further streamline usage and avoid power bricks in favour of a single kind of cable. In this situation, a space in the market is likely to open for a USB-PD source which replaces a conventional plug socket.

Abbreviated Contents

Abstract 2
Background 3
Abbreviated Contents 4
Contents 5
Figures 10
Tables 12

CHAPTER 1
Project Brief 13

CHAPTER 2
Fundamental Theory 14

CHAPTER 3
Requirements Specification 26

CHAPTER 4
Summary of Findings 33

CHAPTER 5
Biphase Mark Code Transceiver 35

CHAPTER 6
Binary Frequency-Shift Keying Transceiver 47

CHAPTER 7
USB Type-C Signalling 58

CHAPTER 8
System Realisation 60

CHAPTER 9
Review 66

APPENDIX A
Circuit Realisation 82

APPENDIX B
System Controller 104

References 127

Contents

	Abstract	2
	Background	3
	Abbreviated Contents	4
	Contents	5
	Figures	10
	Tables	12
CHAPTER 1	Project Brief	13
1.1	Context	13
1.2	Aim	13
1.3	Objectives	13
CHAPTER 2	Fundamental Theory	14
2.1	Design feasibility	14
2.2	USB Power Delivery	15
2.2.1	The USB-PD protocol layer	16
2.2.2	The USB-PD physical layer	18
2.2.3	Biphase mark coding (BMC)	20
2.2.4	Binary frequency-shift keying (BFSK)	21
2.3	USB Type-C	21
2.4	USB Battery Charging	23
CHAPTER 3	Requirements Specification	26
3.1	Power supply requirements	26
3.2	Physical layer behavioural requirements	27
3.3	BMC transceiver requirements	28
3.4	BFSK transceiver requirements	30
3.5	Type-C signalling requirements	32

CHAPTER 4 Summary of Findings 33

CHAPTER 5 Biphase Mark Code Transceiver 35

5.1 Key concerns 35

5.1.1 DC bias in received signals 35

5.1.2 Synchronisation 35

5.2 Transmitter design 36

5.2.1 Transmitter control logic 36

5.2.2 Transmitter line driver 37

5.3 Receiver design 39

5.3.1 Synchronisation logic 39

5.3.2 Decoding logic 41

5.3.3 Line instrumentation 42

CHAPTER 6 Binary Frequency-Shift Keying Transceiver 47

6.1 Key concerns 47

6.1.1 Speed of operation 47

6.1.2 Synchronisation 47

6.2 Transmitter design 47

6.2.1 Modulator design 47

6.2.2 Control logic: four-oscillator design 51

6.2.3 Control logic: Colpitts design 52

6.3 Receiver design 53

6.3.1 Demodulator design 53

6.3.2 Automatic gain controller 55

6.3.3 Timing and decoding logic: time-averaging design 55

6.3.4 Timing and decoding logic: frequency–voltage design 56

6.3.5 Squelch detection 57

CHAPTER 7 USB Type-C Signalling 58

7.1 Rp–Rd signalling 58

7.2 Cable power 58

CHAPTER 8 System Realisation 60
8.1 Design for excellence 60
8.1.1 Design for manufacturing 60
8.1.2 Design for test 61
8.1.3 Design for safety 62
8.2 Test-driven development 62
8.2.1 Integration testing 64
8.2.2 Unit testing 64
8.2.3 Formal verification 64

CHAPTER 9 Review 66
9.1 Project schedule 66
9.2 Designs 67
9.2.1 BMC transceiver circuit 67
9.2.2 BFSK transceiver circuit 68
9.2.3 System controller: BMC transmitter 68
9.2.4 System controller: BMC receiver 69
9.2.5 System controller: FPGA resource utilisation 70
9.3 Objectives 72
9.3.1 Implementing the USB Power Delivery physical layer 72
9.3.2 Supporting legacy communication modes 73
9.3.3 Offering a cost-competitive solution 74
9.3.4 Supporting installation in place of a plug socket 75
9.3.5 Complying with standards and law 77
9.4 Professional competence 78
9.5 Conclusion 80

APPENDIX A
Circuit Realisation 82

CHAPTER A1 Schematic Diagrams 83

CHAPTER A2 Circuit Layout 89

CHAPTER A3 Bill of Materials 90

A3.1 System controller 93
A3.2 BMC transmitter 94
A3.2.1 Slew rate-limited buffer 94
A3.2.2 Transmission gate 95
A3.3 BMC receiver 95
A3.3.1 Subtractor 95
A3.3.2 Schmitt trigger 96
A3.4 BFSK four-oscillator transmitter 96
A3.5 BFSK Colpitts transmitter 97
A3.5.1 Colpitts oscillator 97
A3.5.2 Varactor control 98
A3.6 BFSK automatic gain controller 99
A3.7 BFSK squelch detector 100
A3.8 BFSK time-averaging receiver 100
A3.9 BFSK frequency–voltage receiver 101
A3.10 BFSK bandpass filter 101
A3.11 Type-C signalling 101
A3.11.1 Rp switches 101
A3.11.2 Attachment detectors 101
A3.11.3 Cable power and discharge drivers 102
A3.12 Power supply 102
A3.12.1 Protection 102
A3.12.2 Positive-supply regulators 102
A3.12.3 Negative-supply regulators 103

APPENDIX B
System Controller 104

CHAPTER B1 Overview 105

CHAPTER B2 Biphase Mark Code Transmitter 106
B2.1 Transmitter line-driving block 106
B2.2 Preamble generator 108
B2.3 4b5b encoder 110
B2.4 4b5b decoder 112
B2.5 Control unit 113
B2.6 5-bit Gray code generator 116
B2.7 FIFO9 block 117

CHAPTER B3 Biphase Mark Code Receiver 119
B3.1 Binary search block 119
B3.2 Cyclic redundancy check engine 121
B3.3 End-of-packet detector 122
B3.4 Control unit 124

References 127

Figures

No.	Caption	Page
Figure 1	An example USB-PD negotiation.	17
Figure 2	The general format of USB-PD packets.	19
Figure 3	Biphase mark coding (BMC) of example binary data.	21
Figure 4	Presence and role resistances (Rp–Rd) on the USB Type-C ‘CC’ wire.	22
Figure 5	Block diagram of the project system.	26
Figure 6	State diagram of basic BMC transmitter operation.	36
Figure 7	State diagram of a USB-PD BMC transmitter’s operation.	36
Figure 8	High-level design of the BMC receiver synchronisation logic.	41
Figure 9	State diagram of the BMC receiver decoding logic’s operation.	42
Figure 10	The output of the USB-PD receiver subtraction scheme for various levels of noise-affected input.	43
Figure 11	Operation of a comparator and a Schmitt trigger on a noisy signal.	44
Figure 12	Schmitt trigger operation on BMC receiver subtractor output.	45
Figure 13	The phase difference of 23.2MHz±450kHz waves returning to zero at the end of a data period of 3.33µs.	48
Figure 14	Comparison of topologies of LC sine wave oscillators.	50
Figure 15	High-level design of the four-oscillator BFSK transmitter.	51
Figure 16	State diagram for the four-oscillator BFSK transmitter line logic.	52
Figure 17	FSK demodulation with cascaded multivibrators.	54
Figure 18	Block diagram of the automatic gain controller.	55
Figure 19	High-level design of the time-averaging BFSK receiver.	56
Figure 20	Annotated diagram of the BFSK printed circuit board.	76
Figure 21	Annotated diagram of the BMC printed circuit board.	76

No.	Caption	Page
Figure A1	BMC transceiver circuit layout.	89
Figure A2	BFSK transceiver circuit layout.	89
Figure B1	A high-level block diagram of the system controller.	105
Figure B2	Biphase mark-coded transmission of sequence *10*.	107
Figure B3	Biphase mark-coded transmission of sequence *11*.	107
Figure B4	The output waveform for the USB-PD preamble generator.	108
Figure B5	State diagram for the end-of-packet detector.	123

Tables

No.	Caption	Page
Table 1	USB-PD physical layer ordered sets.	19
Table 2	PLL divisors and multipliers for a four-oscillator BFSK modulator.	49
Table 3	System controller FPGA resource utilisation.	71
Table 4	Cost comparison of semi-discrete and monolithic solutions.	74
Table 5	Review of project against CEng standard.	79
Table A1	Bill of materials for the BMC transceiver circuit.	90
Table A2	Bill of materials for the BFSK transceiver circuit.	92
Table B1	Map of 4b5b encoder input to K-codes.	111
Table B2	Register map for the BMC transmitter control unit.	114
Table B3	Error codes for the BMC transmitter control unit.	115
Table B4	Register map for the BMC receiver control unit.	125
Table B5	Error codes for the BMC receiver control unit.	126

1. Project Brief

1.1 Context

A power supply which conforms to the Universal Serial Bus Power Delivery Specification (USB-PD) and which can be installed in the place of a conventional plug socket is required to meet anticipated market demand. Current offerings for monolithic USB-PD controllers are limited in their flexibility and support for legacy interfaces, and so it is to be investigated whether a non-monolithic design can offer a viable and cost-competitive alternative.

1.2 Aim

The aim of the project is to design and build with semi-discrete components a USB Power Delivery physical layer suitable for integration into a power supply in a conventional plug socket form factor.

1.3 Objectives

1.3.1 Implementing the USB Power Delivery physical layer

The project should produce a design which, as far as necessary in the context, implements the USB-PD physical layer (PHY) and hence enables communication using the biphase mark coding scheme (BMC) required to implement USB-PD.

1.3.2 Supporting legacy communication modes

The project should investigate the viability of including in the produced design support for communication using the legacy binary frequency-shift keying scheme (BFSK) specified in USB-PD revision 2.0.

1.3.3 Offering a cost-competitive solution

The design produced by the project should offer a unit cost at reasonable volume which compares favourably to a monolithic USB-PD controller solution.

1.3.4 Supporting installation in place of a plug socket

The design produced by the project should be suitable, with minimal further work, for integration with other components in a housing that allows installation in a space intended for a conventional plug socket.

1.3.5 Complying with standards and law

The project should identify relevant requirements in standards and law and should produce a design which, as far as necessary in the context, complies with the identified requirements.

2. Fundamental Theory

The objectives in chapter 1 set the scope of the project but are not sufficient *per se* to guide a design; they are abstract and describe the requirements of the user rather than the requirements of the designer. This chapter discusses the theory and background that are the foundation for the concrete design and functional requirements defined in chapter 3.

2.1 Design feasibility

The concept of a USB Power Delivery-based plug socket is not a new one—proof of the concept was presented by Reydams and Lauwereys *et al.* [1] in 2014 and, in their 2018 analysis, Vossos *et al.* [2] considered USB-PD as a candidate standard for use in DC power distribution throughout commercial premises.

The work by Reydams and Lauwereys *et al.* indicates that space constraints are unlikely to be prohibitive even when designing to supply the 100-watt maximum the USB Power Delivery Specification standardises, with their unhoused design measuring in the region of 45×45×90mm (W×D×H). Although their design exceeds BS 4662 [3] plug socket size limits[1] and a housing will add to product size, the transition from a prototype to a commercial design is likely to enable miniaturisation. For example, the circuit board shown in their figure 5 has large areas which are visibly bare. That said, the work Reydams and Lauwereys *et al.* present focuses on the power electronics aspects of the design—at the time of their writing, they note, USB Power Delivery was relatively fresh to the market and so no commercial devices were available for purchase. Consequently, as their design emulated rather than implemented USB-PD, the practicalities of its implementation are not discussed.

Narrowing from the general feasibility of a USB-PD plug socket, it is obvious that implementing the Power Delivery physical layer is feasible. The uncertainty lies in the feasibility of a cost-competitive semi-discrete implementation.

At the lowest end of the cost range, monolithic ‘port controllers’ provide a PHY and a partial implementation of the protocol layer. A separate system controller is required to carry out most USB-PD tasks but, that aside, a first impression might suggest that components such as the FUSB302 [4] (£0.359 at 3ku), the PTN5110 [5] (£0.564 at 4ku), or the UPD350A [6] (£0.72 at 5ku) are the ideal tools for the application. However, their use is limited—these devices support only the biphase mark code scheme, and so additional hardware would be required to

[1] A flush-to-wall one-gang box has a square cavity 68.3mm wide and up to 47mm deep if compliant with BS 4662, fig. 1.

support the binary frequency-shift keying scheme. As the PHY included in these port controllers is integral and not exposed, any additional hardware would have to duplicate the PHY implementation as well as providing co-ordination between the two independent PHYs. A semi-discrete solution with a single PHY attached to two transceivers would reduce redundancy and allow the cost of additional hardware to be distributed between BMC and BFSK transceivers. This has the potential to enable cost competitiveness.

Moving from somewhat barebones USB Type-C port controllers, full-featured charging controllers are available at increased cost. These devices integrate the controller needed to perform protocol-layer tasks and often support standards which supplement USB-PD. The AP43770 [7] (£1.08 at 4ku), for example, offers support for USB Battery Charging, Qualcomm QuickCharge,[2] and USB-PD's programmable power supply features. Devices of this kind share limitations with the more basic port controllers, but more pressing is their flexibility—these parts are fixed to a relatively narrow range of functionality, and any deviation outside this range would require a further controller. This might necessitate a high-cost processing element if more than basic power converter interfacing is required or if the networking that Reydams and Lauwereys *et al.* envisage is implemented.

Added to this, as a turnkey solution, any derived product is entirely dependent on the utilities a manufacturer provides to program the device. For a product which, compared to a microcontroller, is relatively obscure, the support life of these tools is a concern that cannot be ignored. Notably, the AP43770 datasheet provides no information on programming. Brief examination of the devices available shows that support and quality of documentation follow price. The TPS65988 [8] (£2.81 at 2.5ku) offers far better documentation and integrates V_{CONN} and V_{BUS} power switches, but sacrifices QuickCharge support and uses an external memory. The extent of the integration it offers is likely to make it the benchmark in assessing whether a semi-discrete design is cost-competitive.

Overall, these observations suggest that—while it is far from clear-cut—there is the potential for a semi-discrete solution to offer improved compatibility and value as compared to a monolithic solution.

2.2 USB Power Delivery

Although it comes under the broad 'USB' label, USB Power Delivery as a standard is largely independent of the other components of USB. Negotiation between Power Delivery devices happens over a channel entirely separate from that used by USB devices to communicate with their hosts and, except under special

[2] Qualcomm markets QuickCharge as its own high-power USB-based charging specification. The text of the specification is not publicly available, but press coverage indicates that version 4 is compatible with USB-PD.

circumstances,[3] USB data flow and the USB-PD negotiation flow do not influence each other. Indeed, USB data support is intentionally an optional part of USB-PD.[4]

Like USB, the USB-PD specification defines both a physical layer—dealing with the generation, transmission, and interpretation of electrical signals—and a protocol layer, which describes how transmitted data is used to form meaningful messages and how connected devices interact. It has three major revisions, with two continuing to be published by the USB Implementers Forum.

This project will primarily work to USB-PD revision 3.0 as adopted in the 2018 edition of BS EN IEC 62680-1-2 [9]. However, where necessary for objective 1.3.2, reference will be made to revision 2.0 [10].

2.2.1 The USB-PD protocol layer

At the highest level, the Power Delivery protocol layer relies on establishing a contract between the source (power supply) and sink (power consumer)—a fixed agreement on how power is to be delivered and used. The connected devices operate within the bounds of the contract until a reset condition occurs or until they agree to renegotiate and adopt a different power level.

To establish a contract, connected devices progress through four stages:

- **Detection**, where the source and sink detect that a cable has been attached between them and prepare to begin communication;
- **Cable identification**, where the source determines which kind of cable is connected to it either by reference to the standard minimum capabilities[5] or by interrogating electronics integrated into the cable;
- **Offer**, where the source lists to the sink the voltage levels it supports and the currents it can supply at each level, excluding the capabilities which exceed the ratings of the cable;[6]
- **Agreement**, where the sink requests one of the listed capabilities and the source confirms agreement, immediately followed by the source adjusting its supplied power to the requested level.

[3] The USB-PD specification [9] defines (at § 6.3.9) a 'swap data roles' message, used to agree a role reversal where the downstream-facing port (the host) becomes the upstream-facing port (the device) and *vice versa* while maintaining the direction of power flow.

[4] See, in particular, USB-PD revision 3.0 [9] at § 6.4.1.2.2.4.

[5] All Type-C cables can carry 3 amps. If a USB-PD source includes Type-A or Type-B receptacles, it uses a special 'PD Detect' contact to identify high-current cables: regular cables carry 1.5 amps, micro-USB-PD cables carry 3 amps, and full-size USB-PD cables carry 5 amps. See USB-PD revision 2.0 [10] at § 3.1.1, 3.1.6, 3.2.2, and 3.4.

[6] See USB-PD revision 3.0 [9] at § 4.4.

This is done using messages transmitted over the physical layer (see 2.2.2), and most communication is done using 'control' and 'data' messages (which are used to manage the connection and provide device information, respectively). USB-PD does define a third type—the extended message—with an increased payload size, but this type is not relevant to the most common USB-PD tasks.

A message can be addressed to one of three recipients: the opposite Type-C port, the 'electronic marker' chip in the cable-end nearest the source, or the marker chip in the cable-end nearest the sink (if present). The specification uses the somewhat opaque terms SOP, SOP′, and SOP″, respectively, for these addresses. It is this addressing that enables a USB-PD source to interrogate a cable for the current it supports. A typical negotiation involves a simple and relatively short exchange of messages. An example is illustrated in figure 1.

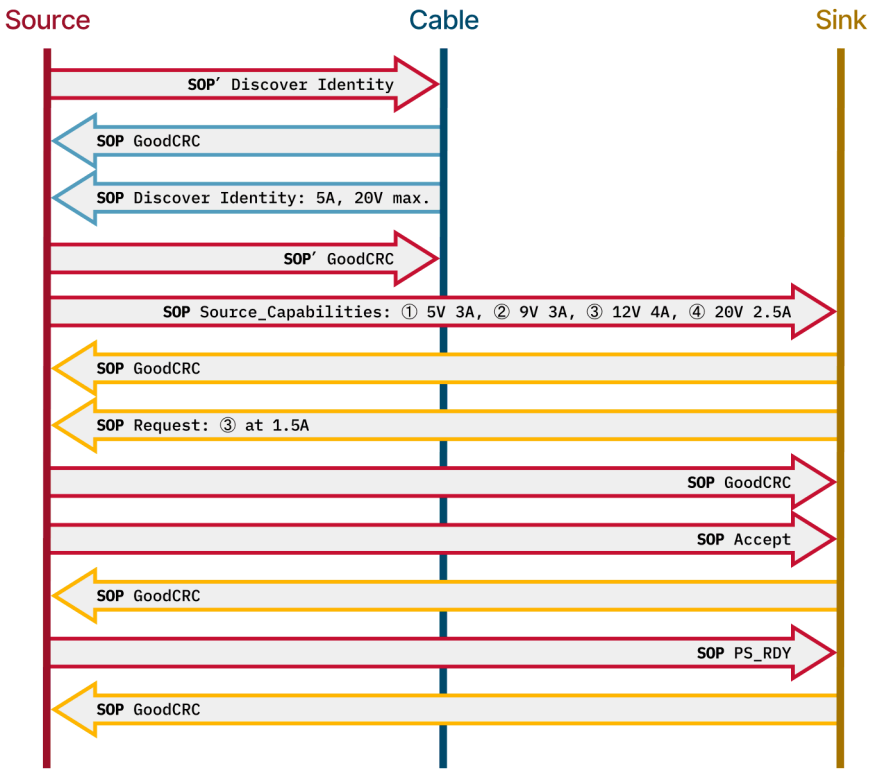


Figure 1. An example USB-PD negotiation.

For each message in this exchange, the protocol layer defines failure conditions which, if met, can substantially alter the message flow. The most general failure conditions include the failure to receive acknowledgement[7] or an anticipated response, although more complex messages have other conditions related to

[7] An acknowledgement *GoodCRC* message must be fully received within 1ms: [9] at § 6.6.1.

their meaning—the *Request* message, for example, might fail if the sink sends an invalid request or if the source is unable to meet the request.[8] The response a failure elicits also differs depending on the message—in the general case, a 'soft reset' first clears protocol-layer state but leaves contracts in place and is followed by a 'hard reset' and ultimately USB Type-C error recovery if any failure cannot be resolved.[9] Hard resets invalidate contracts, while Type-C error recovery electrically disconnects a sink in a non-graceful manner.

More complex failures, and message sequences more generally, are driven by the policy engine—a standardised state machine dictating how interactions between the source, sink, and cable proceed.[10] A core concept used by the policy engine is the 'atomic message sequence' (AMS),[11] which is a sequence of messages which represents a single operation and which is logically indivisible. The use of a message sequence which is not a defined AMS is inherently invalid as the policy engine has no behaviour specified for that sequence. Within the set of defined AMS, each is either interruptible or non-interruptible—that is, a protocol error that occurs during the AMS either causes a general-case failure as above or causes a return to the appropriate ready state.[12]

A policy engine exists for each USB port, and each policy engine in a device is co-ordinated by a 'device policy manager' with knowledge of the entire system. The device policy manager can also implement wider 'system' policies it receives from a USB host over the USB data bus. This has limited relevance to the project.

2.2.2 The USB-PD physical layer

At a lower level in the stack, the physical layer provides three relatively simple services to the protocol layer: packet framing and de-framing, message integrity verification, and media conversion.

The essential unit of transmission is the packet and, although other signalling between USB-PD devices takes place,[13] packets are the only means by which protocol layers can request and send information. Packets are also used to a more limited extent for basic control transmissions, and this difference in use is

8 See USB-PD revision 3.0 [9] at § 6.3.4.

9 See USB-PD revision 3.0 [9] at § 4.5.2.2.2 and 6.8.

10 See USB-PD revision 3.0 [9] at § 8.3 generally and 8.3.3.2 for USB-PD source state diagrams.

11 Per USB-PD revision 3.0 [9] at § 8.3.2, an AMS is any sequence of messages which starts, ends, or both starts and ends in the 'source ready', 'sink ready', or 'cable ready' states (as appropriate for the device on which the policy engine is operating).

12 See 'interruptible' and 'non-interruptible' in USB-PD revision 3.0 [9] at § 1.6.

13 For example, resistances are used to signal whether a sink is or is not permitted to transmit on the USB-PD data channel. These resistances are controlled by the source and are intended as a means of collision avoidance: USB-PD revision 3.0 [9] at § 5.7. Note that, for revision 2.0 [10], collision avoidance relies instead on voltage level detection (for Type-A and Type-B: *ibid.* at § 5.8.2.6.4) or a timeout since last signal transition (for Type-C: *ibid.* at § 5.8.3.6.1).

also the line between two different forms of packet. The general format of a packet is shown in figure 2.

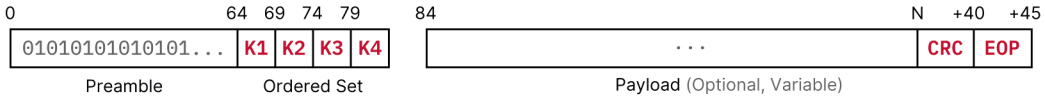


Figure 2. The general format of USB-PD packets.

Common to each format are the preamble—an alternating sequence of *1*s and *0*s to be used to synchronise clocks—and the 'ordered set', which identifies the kind of packet and hence whether it includes a payload. The ordered set is a resilient data structure composed of four 5-bit codes known as 'K-codes', and its valid values are defined in such a way that they can be identified when three of four K-codes are present and correct.[14] Table 1 lists relevant ordered sets.[15]

Table 1. USB-PD physical layer ordered sets.

Set	K-codes				Purpose
Control transmissions					
Hard_Reset	RST-1	RST-1	RST-1	RST-2	Hard-resets devices and cable.
Cable_Reset	RST-1	Sync-1	RST-1	Sync-3	Hard-resets cable.
Data transmissions					
SOP	Sync-1	Sync-1	Sync-1	Sync-2	Messages for ports.
SOP′	Sync-1	Sync-1	Sync-3	Sync-3	Messages for source eMarker.
SOP″	Sync-1	Sync-3	Sync-1	Sync-3	Messages for sink eMarker.

A control transmission is terminated by its ordered set.[16] Data transmissions, on the other hand, include a variable-length payload, a check code, and a terminator after their ordered set.[17] The payload is simply the message forwarded by the protocol layer, while the 'CRC' (cyclic redundancy check) field is a check code calculated from the payload and used to verify that it has not been damaged or corrupted in transit.[18] The 'EOP' field is a K-code used to indicate that the end of

[14] See USB-PD revision 3.0 [9] at § 5.4.

[15] Two further ordered sets, SOP′_Debug and SOP″_Debug, are defined but have no specified purpose or meaning: see USB-PD revision 3.0 [9] at § 5.6.1.2.4 and 5.6.1.2.5.

[16] Note that this separation is not expressly reflected in the USB-PD specification [9], and so each control transmission is independently specified in this format: *ibid.* at § 5.6.4 and 5.6.5.

[17] See USB-PD revision 3.0 [9] at Figure 5-3.

[18] See USB-PD revision 3.0 [9] at § 5.6.1.4, 5.6.2, and 5.6.3.

the data has been reached. An in-band terminator like 'EOP' would normally impact the ability of a channel to carry arbitrary data by preventing the use of whichever value represents the terminator. However, the design of USB-PD's media conversion service makes this a non-issue.

The conversion to the USB-PD data channel line format involves two steps: the mapping of payload data to a '4b5b' line-coded form,[19] and then final conversion to the electrical signals transmitted between devices. That line code—which uses 5-bit symbols to represent 4 bits of data—is intended to ensure that there are sufficient transitions between *0* and *1* to allow a receiver to remain synchronised with the transmitter, but its use has the added benefit of enabling use of in-band signalling that does not interfere with data transmission; a 5-bit code can represent 32 symbols, but only 16 symbols are needed to represent all 4-bit values. USB-PD uses the remaining symbols to represent its K-codes, including the packet terminator sent with data transmissions.

Once line coding has been applied, the resultant symbols are converted to electrical signals in one of two ways. For USB Type-C cables, the biphase mark coding scheme (see 2.2.3) is used and the output is transmitted over the 'CC' wire in the cable.[20] Where instead a Type-A or Type-B cable is attached, a binary frequency-shift keying scheme (see 2.2.4) is used and is transmitted over the USB power conductor V_{BUS}.[21] Formally, the latter scheme is deprecated in USB-PD revision 3.0. Its use is permitted by revision 2.0 with the caveat that, where used by a source with a USB Type-C receptacle, the BMC scheme must be supported and must be used for initial attempts at communication.[22]

Irrespective of the scheme used, data is sent over the line in little-endian order.[23]

2.2.3 Biphase mark coding (BMC)

The USB Power Delivery specification sources its definition of biphase mark coding from BS EN 60958-1 [11],[24] a standard for an audio interface implemented in S/PDIF. Onto this definition it adds its own electrical parameters: a signalling voltage swing of 1.125 volts nominal, a data rate of 300 kilobits per second, and a host of timing parameters which dictate how quickly and slowly the signal can swing from one voltage to another.

The scheme itself is relatively simple. A clock operates at twice the data rate, and

[19] See USB-PD revision 3.0 [9] at § 5.3.

[20] See USB-PD revision 3.0 [9] at § 5.8 and revision 2.0 [10] at § 5.8.3.

[21] See USB-PD revision 2.0 [10] at § 5.8.2.

[22] See USB-PD revision 2.0 [10] at § 5.8.4. That section also imposes other similar requirements, such as that any hard reset must be signalled using both schemes if both are supported.

[23] See USB-PD revision 3.0 [9] at § 5.5 and revision 2.0 [10] at § 5.5.

[24] *Op. cit.* at § 4.2 *per* USB-PD revision 3.0 [9] at § 5.8.

each bit is encoded in a ‘unit interval’ of two clock pulses. The line starts at zero and its state inverts at the beginning of each unit interval. A measurement of the line is taken on each clock pulse and the value represented in a unit interval is determined by comparing the two measurements that occur during the interval: equal represents *0*, and inequal represents *1*. An example is shown in figure 3.

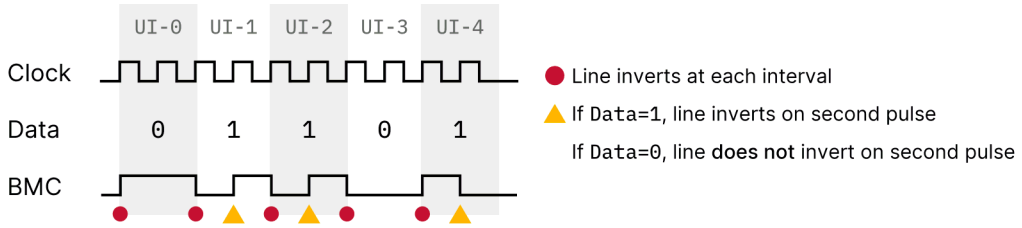


Figure 3. Biphase mark coding (BMC) of example binary data.

The result of this is that line state does not, in isolation, map to data. This is especially clear in comparing *UI-0* and *UI-3*—each represents *0*, but the line is high in *UI-0* and low in *UI-3*. Similarly, when encoding *1*, it can be seen in *UI-2* and *UI-4* that the line state may move through either low-to-high or high-to-low depending on the previous state of the line.

2.2.4 Binary frequency-shift keying (BFSK)

Conceptually, USB-PD revision 2.0’s binary frequency-shift keying scheme is even simpler than BMC. A carrier wave (nominally 23.2MHz) is injected onto the USB power conductor V_{BUS} and its frequency is modulated up or down by 500kHz to transmit either a *1* or *0*, respectively.[25] Practically, the complexity of the BFSK scheme lies in its use of high-frequency signals and in the requirement for transmitters to use ‘continuous phase’ BFSK, where no abrupt change in phase is permitted when switching between frequencies.[26]

2.3 USB Type-C

Much as USB was intended to be a universal interface for devices with data and power connections, USB Type-C is intended to be a universal connector that can provide a USB data connection, source enough power for most purposes through

[25] See USB-PD revision 2.0 [10] at § 5.8.2. The wave function is not normatively stated, but a sine wave is what would be conventionally expected and the informative Appendix C (*ibid.* at C.1.2) suggests the use of a sine wave carrier in testing for power converter noise.

[26] If, for example, two independent oscillators are running at 23.7MHz and 22.7MHz, the two will not be in phase. This can be seen by comparing $y_1 = \sin(23.7x)$ and $y_2 = \sin(22.7x)$. At $x = 0.3314$, y_1 is at 100% amplitude and y_2 is at 94.57% amplitude. Simply switching between the two oscillators would distort the output wave as the amplitude would instantaneously drop 5.43%. As the signal is no longer a pure sine wave, it introduces undesirable frequency components.

USB-PD, and host other interfaces as required with well-defined extensibility.[27]

The Type-C specification was standardised in BS EN IEC 62680-1-3. This project will work to the 2018 edition [12] of the standard.

Before Type-C's definition, USB connectors were generally asymmetric and indicated the role the device would assume:[28] a USB host always had a Type-A receptacle and a USB device always had a Type-B receptacle. This simplified connection and ensured a connection was always valid. The Type-C connector does neither—it is horizontally symmetric and so can be inserted 'upside-down', and the same connector is used for USB hosts and devices. As a result, a given pin on a receptacle may become connected to either of two wires in a cable, and a user may invalidly attach a host to a host or a device to a device.

To enable this to work, BS EN IEC 62680-1-3 defines a system which provides means of detecting role, orientation, and power-sourcing capability. Using pairs of resistances connected to each of the two CC wires, a source can verify that it is attached to a sink, determine how the cable is oriented, establish whether the cable requires power, and advertise a basic power level to the sink. Although not especially closely related logically, each of the means is dealt with in this section because of their close electrical relationship. Figure 4 illustrates this scheme.[29]

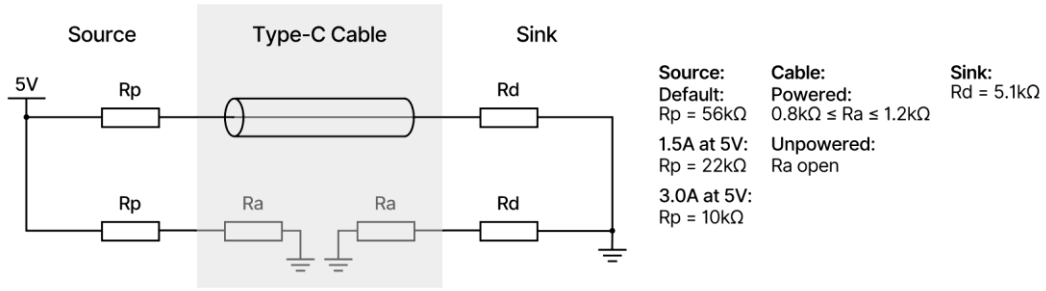


Figure 4. Presence and role resistances (Rp–Rd) on the USB Type-C 'CC' wire.

In this figure, Ra is the impedance presented by the cable electronics, if present.

As discussed in 2.2, each port has independent power and data roles: a source provides power to a sink and is, on attachment, the downstream-facing port (or the host). To enable power role detection, a source independently pulls each CC

[27] For example, USB Type-C can host DisplayPort (*per* [12] at § 5.1.4), the analogue signals normally transmitted over 3.5mm jacks (*ibid.* at Appendix A), Thunderbolt [50], and HDMI [51].

[28] Note that special USB 'On-The-Go' connectors exist and allow exchanging roles, but that the cable type continues to determine role until an exchange is negotiated.

[29] This diagram is based on Figure 4-5 in BS EN IEC 62680-1-3 [12].

wire up to 5 volts, and a sink independently pulls each wire down.[30] In the cable, one CC wire connects the two cable-ends and the other may be open-circuited or connected to the cable's marker electronics.

These resistances form voltage dividers on each CC wire, and so by measuring the voltage at each midpoint node a source can determine both whether a device is connected and, if it is, how it is oriented. The source uses this to connect its transceiver to the correct CC wire and, if it detects Ra, to attach V_{CONN} to the wire that will power the cable electronics. The source then energises V_{BUS} to power the sink and begin USB-PD negotiation.[31]

As noted in 2.2.1, detecting the presence of an electronic marker (as indicated by the presence of Ra) is important in USB-PD negotiation—a source cannot know whether a Type-C cable is capable of carrying more than 3 amps unless it can interrogate the electronic marker to request the cable's rated current.

2.4 USB Battery Charging

The first revisions of the USB specification presented USB as a data interface which also provided power. As such, USB hosts were able to supply 2.5 watts in most cases, down to as little as 0.5 watts for low-power ports.[32] This was enough for basic peripherals, but too little to quickly charge the smartphones that were becoming increasingly prevalent in the mid to late 2000s. Manufacturers solved this by introducing their own indication schemes,[33] and it was likely this that prompted the 2007 publication of the USB Battery Charging specification.

The most recent revision of the Battery Charging specification (USB-BC), revision 1.2, was published in 2010 [13]. It is the closest precursor to USB-PD and is the revision this project will refer to.

In contrast with USB-PD, USB-BC is a 'dumb' protocol—no exchange of messages occurs, and all capabilities are instead advertised by adjusting the electrical characteristics of the USB data lines D+ and D−. This is used to allow the device receiving power to distinguish between and adjust its behaviour to accommodate

[30] Note that BS EN IEC 62680-1-3 [12] permits other implementations. A source may use a current source instead of a resistor or might pull CC up to a voltage other than 5 volts, while a sink might use a voltage clamp (such as a Zener diode or TL431) rather than a pull-down: see *ibid.* at § 4.11.1 and 4.11.3 with Tables 4-20, 4-21, and 4-28 to 4-32.

[31] See BS EN IEC 62680-1-3 [12] at § 4.5.1.2.1.

[32] See USB revision 1.1 [52] at § 7.2.1, which defines a unit load as 100mA and states that low-power devices can draw one unit load and high-power devices five unit loads. Further, although seldom implemented, *ibid.* at § 7.1.7.4 and 7.2.3 requires a host to suspend a device which is idle for more than 3ms. Suspension limits current draw to 0.5mA or 2.5mA (2.5 to 12.5mW).

[33] Examples of such schemes can be found in datasheets for 'Dedicated Charging Port Controllers', such as that for the TPS2513A [63] at § 8.3.

five kinds of port:

- **Standard Downstream Ports** (SDPs), which provide the default USB power level and expect a device to limit the current it draws accordingly;[34]
- **Charging Downstream Ports** (CDPs), which are largely equivalent to SDPs but permit a device to draw increased current without negotiation;[35]
- **Dedicated Charging Ports** (DCPs), which have no USB data connection and looser limits on permitted V_{BUS} voltage;[36]
- **Accessory Charging Adapters** (ACAs), which are used to connect the device being powered to a CDP or DCP and another device, enabling it both to receive power and act as a USB host for the other device;[37]
- **ACA-Docks**, which are conceptually similar to ACAs but use an external non-USB power source and must meet the power requirements of CDPs.[38]

Once attached, the device being powered drives the D+ and D− lines and observes the response: SDPs and DCPs do not respond, but resistances connect the data lines together and the mode of connection changes the voltage measurable on the D− line;[39] a CDP drives D− until it detects the device driving D+, prompting the device to drive D− and measure the voltage on D+;[40] and ACA-Docks both bias the data lines and connect a specified resistance to a dedicated 'ID' contact, each of which the device detects and which are sufficient to unambiguously identify the port as an ACA-Dock.[41] ACA detection relies solely on the resistance on the

[34] See USB-BC [13] at § 1.4.13. The pull-down resistances mentioned in that section are the standard resistances specified in USB revision 2.0 [53] at § 7.1.1.3 and 7.1.5.1 with Table 7-7.

[35] A CDP must be capable of sourcing at least 1.5 amps, and a device being powered can draw this before a USB connection is established: see USB-BC [13] at § 4.2.1 with Table 5-2.

[36] DCPs must be able to source at least 500mA, after which their V_{BUS} may droop. If more than 1.5 amps is sourced, or if V_{BUS} droops below 2 volts, they may shut down: see USB-BC [13] at § 4.4.1.

[37] See USB-BC [13] at § 6.1.

[38] See USB-BC [13] at § 4.3. An ACA takes its sourcing behaviour from its attached charging port, which may be a CDP or DCP. As an ACA-Dock does not use a USB charging port, the standard sets out that the CDP requirements are those that apply.

[39] An SDP pulls both lines low while a DCP shorts the two lines together: see USB-BC [13] at § 3.2.4.1 and 3.2.4.3. This means that an SDP will present a near-ground voltage on D− while a DCP will present a voltage at almost the same level as was driven.

[40] As the voltage the CDP uses to drive D− is the same as that a DCP would present, a device proceeds to 'secondary detection' where it establishes if a device is a CDP or a DCP. The CDP does not short the data lines in the way a DCP does, and so D+ measures near ground without the device driving it: see USB-BC [13] at § 3.2.4.2 and 3.2.5.2 with Table 5-1.

[41] See USB-BC [13] at § 3.2.4.4.

'ID' pin,[42] while Micro ACA detection[43] relies on the 'ID' pin and measuring voltages on the data lines connected to the accessory.[44]

Most relevant to the project is the Dedicated Charging Port behaviour—a USB-PD source which can supply 100 watts will be more than capable of supplying the currents necessary to comply with USB-BC, and in a plug socket form factor such a source is unlikely to require a USB data connection.

[42] See USB-BC [13] at § 3.2.6.

[43] A Micro ACA is an ACA with a USB On-The-Go port for the device being powered as well as a USB micro-AB receptacle for the accessory. As On-The-Go enables role swapping, detection is necessary to avoid a violation of the USB specification resulting from two hosts attempting to assert themselves: see USB-BC [13] at § 6.2 and, in particular, 6.2.5.

[44] See USB-BC [13] at § 3.2.4.5.

3. Requirements Specification

This chapter sets out design and functional requirements for the project. Each is derived from the objectives in chapter 1 and based on the foundations set out in chapter 2. For the purposes of this specification, the structure of the system that the project aims to implement is that shown in figure 5.

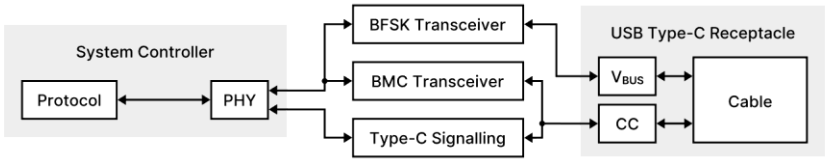


Figure 5. Block diagram of the project system.

In a full USB-PD plug socket system, the system controller would also control or interface with an isolated power converter. This is outwith the project's scope.

3.1 Power supply requirements

The PHY must short USB data lines D+ and D− together through a resistance of no more than 200Ω (**3.1.1**).

> Rationale: This is required by USB-BC [13] at § 4.4.3 and indicates to a USB device that the source is capable of supplying between 500mA and 5 amps without the need for negotiation, subject to voltage droop limits. This will enable fast charging even when a non-USB-PD device is attached.

The PHY must limit V_{BUS} capacitance to 10µF (**3.1.2**) and secure that the impedance between V_{BUS} and GND exceeds 72.4kΩ (**3.1.3**) until it detects the attachment of a sink. The PHY must return V_{BUS} to this state when the sink is detached (**3.1.4**).

> Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.4.2. See also the relevant requirement 3.5.2, which relates to Rp–Rd signalling.

If operating in either of the Disabled or Error Recovery states, the PHY must not drive V_{BUS} or V_{CONN} (**3.1.5**).

> Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.5.2.2.1 and 4.5.2.2.2. See also the related requirement 3.5.3.

The PHY must energise V_{BUS} to 5±0.25 volts within 275ms of attachment (**3.1.6**) and de-energise V_{BUS} to 0.0–0.8 volts within 650ms of detachment (**3.1.7**).

Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.11.2 and 4.5.2.2.1 with voltage levels given in USB-PD revision 3.0 [9] in Table 7-21. That Table also restates the requirements as to timing from BS EN IEC 62680-1-3. See also requirements 3.1.5 and 3.2.4, which restrict this requirement.

The PHY must be capable of connecting V_{CONN} to either CC wire (**3.1.8**).

Rationale: This is necessary to power Type-C powered cables, which use one of the CC wires to communicate and one as a power source. As Type-C is reversible, it cannot be known prior to attachment which CC wire will be which. Refer to BS EN IEC 62680-1-3 [12] at § 4.5.1.2.1 and 4.5.1.3.1.

The PHY must energise V_{CONN} to 3.0–5.5 volts no more than 2ms after energising V_{BUS} (**3.1.9**), must be capable of sourcing at least 1 watt on V_{CONN} (**3.1.10**), and must disconnect V_{CONN} within 35ms of detachment (**3.1.11**).

Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.4.3, 4.5.2.2.9, and 4.11.2 with the parameters given in Tables 4-5 and 4-25. Note that V_{CONN} may be energised before V_{BUS}.

The PHY must connect a discharge resistance Rdch of 30–6120Ω between V_{CONN} and ground if the sink is detached when V_{CONN} is energised and until V_{CONN} is at a voltage of 0.8 volts or less (**3.1.12**).

Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.4.3 and 4.5.2.2.6 with Tables 4-5 and 4-6 and Figure 4-12. See also requirement 3.2.4, which relates to PHY behaviour during V_{CONN} discharging.

3.2 Physical layer behavioural requirements

The PHY must not consider an attachment to be made unless, after 100–200ms of first detecting the attachment, the states of the CC wires are unchanged (**3.2.1**).

Rationale: This required by BS EN IEC 62680-1-3 [12] at § 4.5.2.2.8.2 with parameters taken from Table 4-27. The waiting period is intended to allow values to settle after initial attachment. See also requirement 3.5.2, which relates to the detection of attachment and orientation.

The PHY must monitor for detachment (**3.2.2**) and must consider detachment to have occurred if the CC wire initially connected to Rd is open-circuited (**3.2.3**).

Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.5.2.2.9.2.

Whilst discharging V_{CONN} in fulfilment of requirement 3.1.12, the PHY must not heed attachments until V_{CONN} has discharged below 0.8 volts (**3.2.4**).

Rationale: This is a requirement of BS EN IEC 62680-1-3 [12] at § 4.5.2.2.6 with the voltage level given in Table 4-6.

The PHY must not begin USB-PD configuration before V_{BUS} has reached the voltage specified in requirement 3.1.6 (**3.2.5**).

Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.5.2.2.9.1.

The PHY must not attempt to communicate with a cable's electronic marker unless at least 50ms has elapsed since V_{CONN} was energised (**3.2.6**).

Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.9.1.

3.3 BMC transceiver requirements

The BMC transceiver must be capable of operating in separate modes for USB-PD revision 2.0 and USB-PD revision 3.0 (**3.3.1**).

Rationale: This is necessary to support relatively minor differences in BMC transmission between the two revisions.

The BMC transceiver must be capable of connecting to either CC wire (**3.3.2**).

Rationale: This is necessary to accommodate the reversibility of Type-C. Refer to BS EN IEC 62680-1-3 [12] at § 4.5.1.2.1 and 4.5.1.3.1.

The BMC receiver must present an input impedance of at least 1MΩ (**3.3.3**).

Rationale: This is a requirement of USB-PD revision 3.0 [9] at § 5.8.6.

The BMC receiver must be capable of synchronising its clock with that of a transmitter using the preamble prepended to each transmission (**3.3.4**) and must tolerate a clock frequency in the range 540–660kHz (**3.3.5**).

Rationale: BMC relies on taking line measurements in time with a clock but does not provide a clock line between devices, and so the preamble must be used instead to derive a clock. The USB-PD specification also permits data rates in the range 270–330kbps, and so the clock frequencies given above must be supported.

The BMC receiver must recognise a line voltage of 1.05–1.20 volts as a logic high and a line voltage of 0.6 volts or less as a logic low (**3.3.6**). If the line is idle, the BMC receiver must ignore noise with an amplitude of 0.3 volts or less (**3.3.7**).

Rationale: USB-PD revision 3.0 [9] at § 5.8.1 and 5.8.4 specifies 1.05–1.20 volts as the swing to be produced by the transmitter's driver. A logic low voltage is not given, but it is reasonable to assume a target of 0 volts and reasonable to take 0.6 volts (half the maximum logic high voltage) as an upper bound. This is supported by BS EN IEC 62680-1-3 [12] in Figure 4-36. The noise level requirement is derived from *ibid.* at § 5.8.2.2.

The BMC receiver must measure and decode transmissions on the line using the biphase mark coding scheme defined in USB-PD revision 3.0 [9] (**3.3.8**).

Rationale: This is a requirement of the USB-PD specification, *ibid.* at § 5.8.

The BMC receiver must be capable of receiving a signal which complies with the limits and masks in USB-PD revision 3.0 [9] at § 5.8.2.2 (**3.3.9**).

Rationale: This is necessary to support the rise and fall times a transmitter is permitted to exhibit.

The BMC transmitter must be capable of signalling at 600kHz (**3.3.10**).

Rationale: This is necessary to give the 300kbps nominal data rate specified in USB-PD revision 3.0 [9] at § 5.8.4.1.

The BMC transmitter must begin a transmission by driving the line low (**3.3.11**).

Rationale: This is a requirement of USB-PD revision 3.0 [9] at § 5.8.1.

The BMC transmitter must not begin a transmission unless a period of at least 25µs has elapsed since the line became idle, since the transmitter concluded its previous transmission, or since the previous frame was received (**3.3.12**).

Rationale: This is required by USB-PD revision 3.0 [9] at § 5.7, 5.8.5.4, and 5.8.6.1, and parameters are given in Tables 5-18 and 5-19. The 'line idle' condition occurs when there are fewer than three transitions on the line within the last 12 to 20µs. Note that the 25µs includes the periods in requirements 3.3.16 to 3.3.20. See also requirement 3.3.13 for collision-avoidance signalling timing. This requirement is mirrored for BFSK by requirement 3.4.19.

In USB-PD revision 3.0 mode, the BMC transmitter must signal 'transmit no go' at least 16 to 20ms before beginning a transmission (**3.3.13**) and must otherwise signal 'transmit ok' (**3.3.14**).

Rationale: This is required by USB-PD revision 3.0 [9] at § 5.7 and is used to avoid collisions on the line. These states are signalled using the CC wire pull-up Rp, which indicates that a device is a source and advertises its current capability. It can be seen from Table 5-13 (*ibid.*) with BS EN IEC 62680-1-3 [12] at Table 4-20 that 'no go' is signalled by Rp = 22kΩ ±5% and 'ok' by Rp = 10kΩ ±5%. The initial value of Rp to be used is specified in requirement 3.5.1. Note that no equivalent provision is made for USB-PD revision 2.0. See also requirement 3.3.12 for timing around when a transmission may begin.

After completing a transmission, the BMC transmitter must invert the state of the line (**3.3.15**). If this causes the transmitter to drive the line low, the transmitter must hold the line low for at least 1µs (**3.3.16**) and present a high impedance within 23µs (**3.3.17**). If this causes the transmitter to drive the line high, the transmitter must hold the line high for one unit interval (**3.3.18**), drive the line low for at least 1µs (**3.3.19**), and present a high impedance within 23µs (**3.3.20**).

Rationale: This is a requirement of USB-PD revision 3.0 [9] at § 5.8.1, illustrated in Figures 5-11 through 5-14. The final transition is intended to ensure the final bit is correctly received. For the avoidance of doubt, the period in 3.3.17 includes the period in 3.3.16. The same is true of 3.3.20 with 3.3.18 and 3.3.19.

The BMC transmitter must transmit logic high using a line voltage of 1.125 volts and logic low using a line voltage of 0 volts (**3.3.21**). When not transmitting, the transmitter must present a high impedance to the line (**3.3.22**).

Rationale: The logic high voltage is the nominal voltage swing to be produced by the transmitter's driver *per* USB-PD revision 3.0 [9] at § 5.8.1 and 5.8.4. The logic low level is not specified but, given the phrasing 'swing' used for logic high, it is reasonable to assume a swing from reference potential is intended. Presenting a high impedance is implicit in the requirement to use a tristate driver. Also see requirements 3.3.16 to 3.3.20 relating to timing on transmission completion.

The rise and fall times of the BMC transmitter output must comply with the limits and masks in USB-PD revision 3.0 [9] at § 5.8.2.1 (**3.3.23**).

Rationale: This is necessary, *per op. cit.*, to limit coupling to other conductors in the USB cable. The nature of this requirement is such that it makes the most sense to reference the diagrams rather than describe the requirement.

The BMC transmitter must present a capacitance of 200 to 600pF to the line when not transmitting (**3.3.24**).

Rationale: This is a requirement of USB-PD revision 3.0 [9] at § 5.8.5.1.

The BMC transmitter must present an output impedance of 33 to 75Ω at a frequency of 750kHz (**3.3.25**).

Rationale: This is a requirement of USB-PD revision 3.0 [9] at § 5.8.5.2.

The BMC transmitter must be capable of withstanding without damage a short to ground lasting 120µs (**3.3.26**).

Rationale: This is required by USB-PD revision 3.0 [9] at § 5.8.5.5. This is intended to support the 'fast role swap' functionality of hubs, which is not relevant here.

3.4 BFSK transceiver requirements

The BFSK transceiver must be isolated from the V_{BUS} bulk capacitance and the power supply by an impedance of at least 80Ω at frequencies 20.4–26MHz (**3.4.1**).

Rationale: This is required by USB-PD revision 2.0 [10] at § 5.8.2.2 with the parameters given in Table 5-17. *Ibid.* at C.1 suggests the use of a 1µH inductance, which provides a 128–163Ω impedance, and discusses potential issues in the implementation of the isolation impedance.

The BFSK transceiver must be AC-coupled to V_{BUS}, or must be rated to withstand the maximum voltage which may become present on that conductor (**3.4.2**).

Rationale: This is required by USB-PD revision 2.0 [10] at § 5.8.2.3.

The BFSK receiver must present an input impedance of at least 15kΩ ±15% at any frequency in the range 19–27MHz (**3.4.3**).

> Rationale: This is a requirement of USB-PD revision 2.0 [10] at § 5.8.2.6.1. Note that the input impedance between V_{BUS} and ground is determined by the transmitter termination resistance (see requirement 3.4.10).

The BFSK receiver must be capable of synchronising its clock with that of a transmitter using either the carrier transmitted prior to transmission (refer to requirement 3.4.12) or the preamble prepended to each transmission (**3.4.4**) and must be able to synchronise to frequencies in the range 21.8–24.6MHz (**3.4.5**).

> Rationale: BFSK uses differing frequencies to encode logic high and logic low. As no marker event occurs in each symbol period (compare with BMC), to decode a sequence of one state (*e.g.* 11_2 or 00_2) it is necessary to rely on timing. A data rate of 300kbps produces symbol periods of 3.33µs, and so a logic-low frequency which persists for 6.66µs can be interpreted as 00_2.

The BFSK receiver must support a carrier frequency of 22.4–24MHz (**3.4.6**) and must interpret a frequency 450–600kHz slower as a logic low (**3.4.7**) and a frequency 450–600kHz faster as a logic high (**3.4.8**), provided that the amplitude of each signal is 55–300mV$_{RMS}$ (**3.4.9**).

> Rationale: These frequencies are set out in USB-PD revision 2.0 [10] at § 5.8.2 and in Table 5-16. The signal amplitude is given in *ibid.* in Table 5-20.

The BFSK transmitter must have a termination resistance of 62±10Ω (**3.4.10**).

> Rationale: This is required by USB-PD revision 2.0 [10] at § 5.8.2.5.1.

The BFSK transmitter must comply with the frequency deviation and transmit spectral masks given in USB-PD revision 2.0 [10] at § 5.8.2.5.2 and shown in Figures 5-11 and 5-12 (**3.4.11**).

> Rationale: This is necessary to comply with standard limits on emitted noise.

The BFSK transmitter must transmit a carrier-frequency signal for 1µs before transmitting a frame (**3.4.12**) and must not transmit the carrier for more than 4µs after the end of the last bit of its transmission (**3.4.13**).

> Rationale: This is a requirement of USB-PD revision 2.0 [10] at § 5.8.2.5.2 with the parameters in Tables 5-14 and 5-18. See also requirements 3.4.14 to 3.4.17.

The BFSK transmitter must use a carrier frequency of 22.4–24MHz (**3.4.14**), must transmit logic low with a frequency 450–600kHz slower than the carrier (**3.4.15**), must transmit a logic high with a frequency 450–600kHz faster (**3.4.16**), and must transmit with a signal amplitude of 100–200mV$_{RMS}$ (**3.4.17**).

> Rationale: This is required by USB-PD revision 2.0 [10] at § 5.8.2.4.1 and 5.8.2.5.1 with the parameters in Tables 5-16 and 5-18.

The BFSK transmitter must be capable of signalling at 300kbps (**3.4.18**).

Rationale: This is required by USB-PD revision 2.0 [10] at § 5.8.1.1.

The BFSK transmitter must not begin a transmission unless a period of at least 25µs has elapsed since the line became idle, since the transmitter concluded its previous transmission, or since the previous frame was received (**3.4.19**).

Rationale: This is a requirement of USB-PD revision 2.0 [10] at § 5.7, 5.8.1.4, and 5.8.2.6.4. Relevant parameters are given in Tables 5-14, 5-17, and 5-20. The 'line idle' condition occurs when no signal in the frequency range 20.4–26MHz is detected on V_{BUS} with amplitude of or above 35–55mV_{RMS}. Note that the 25µs includes the periods in requirements 3.4.12 and 3.4.13. This requirement is mirrored for BMC by requirement 3.3.12.

3.5 Type-C signalling requirements

The PHY must pull each USB Type-C CC wire up to 5 volts using independent resistances Rp of 10kΩ ±5% (**3.5.1**).

Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.11.1 to indicate that the device is a source capable of sourcing at least 3 amps at 5 volts. Relevant to this requirement is requirement 3.3.14, which requires varying Rp, and requirement 3.5.4, which restricts when Rp may be presented.

The PHY must be capable of detecting whether a CC wire is open-circuited, attached to a sink pull-down resistance Rd of 5.1kΩ ±20%, or attached to a cable marker electronics resistance Ra of 800–1200Ω (**3.5.2**).

Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.5.1.2.1 with Tables 4-20 to 4-22 and is necessary to detect whether a sink is attached and, if it is, how the attaching cable is oriented. See also section 2.3. *Per ibid.* in Table 4-30, a CC voltage of 0.00–0.75 volts is expected for Ra, 0.85–2.45 volts for Rd, and a voltage of 2.75 volts or above for an open circuit. See also requirement 3.2.1, which relates to the process of confirming attachment.

If operating in either of the Disabled or Error Recovery states, the PHY must present an impedance to ground of at least 126kΩ on each CC wire (**3.5.3**).

Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.5.2.2.1 and 4.5.2.2.2 with the minimum impedance figure given in Table 4-23. See also the related requirement 3.1.5.

Whilst discharging V_{CONN} in fulfilment of requirement 3.1.12, the PHY must not present a pull-up Rp on V_{CONN} (**3.5.4**).

Rationale: This is required by BS EN IEC 62680-1-3 [12] at § 4.5.2.2.6.

4. Summary of Findings

This chapter summarises the findings made in chapters 5 to 7, which describe the research done to fulfil the requirements in chapter 3.

In summary:

- The biphase mark code transmitter is most easily and reliably implemented with a slew rate-limiting RC filter, buffered and connected to the line through a transmission gate (section 5.2.2);
- The biphase mark code receiver can be implemented largely in digital logic using a reference counter to measure pulse duration, where counting rate is successively approximated using the USB-PD preamble (section 5.3.1);
- The analogue portion of the biphase mark code receiver can use a reference design from USB-PD, where DC bias tolerance is obtained by taking the difference between two cascaded filters and extracting a square wave using a hysteresis comparator fed with the output (section 5.3.3);
- The binary frequency-shift keying transmitter must maintain continuous phase in its output and so can be viably implemented with four orthogonal-frequency oscillators or a tuneable LC oscillator (section 6.2.1);
- The four-oscillator transmitter would use a high-speed switch to connect the appropriate signalling frequency to the line at the start of each data period, minimising distortion but at relatively high cost (section 6.2.2);
- The tuneable LC oscillator transmitter would use a varactor—a DC-controlled variable capacitance—to vary output frequency, with an analogue control signal produced by filtering pulse-width-modulated output (section 6.2.3);
- The binary frequency-shift keying receiver implementation is restricted by loose tolerances (a 0.9–1.2MHz band at any point from 21.8–24.6MHz), and so it could be implemented either by relying on high-speed logic to count pulses or by measuring a phase-locked loop's control signal (section 6.3.1);
- The low signal amplitude permitted by the binary frequency-shift keying scheme would require an automatic gain controller to produce a fixed output suitable for processing (section 6.3.2);
- The digital logic method counts reference clock cycles over a given period and uses changes between counts to detect frequency shifts (section 6.3.3);

- The phase-locked loop-based method divides the linear control signal into eight 350kHz bands and so, with comparators and a priority encoder, can detect frequency shifting by a change in encoder output (section 6.3.4);
- In each case, the output from the binary frequency-shift keying receiver can be decoded by taking the time between frequency shifts and dividing by the number of counts at the carrier frequency, relying on USB-PD's guaranteed maximum durations at any frequency (sections 6.3.3 and 6.3.4);
- A pair of signalling resistances are required for full USB-PD compliance, and transmission gates can be used to control line connection (section 7.1);
- Up to 1 watt of power must be provided to USB Type-C cables and a fast-discharge system must be provided for disconnection, which can be done using a push–pull driver with logic-level MOSFETs (section 7.2).

5. Biphase Mark Code Transceiver

This chapter discusses the design and implementation of the biphase mark code (BMC) transceiver required to implement the USB-PD physical layer. Its requirements are set out in section 3.3.

5.1 Key concerns

5.1.1 DC bias in received signals

In the general case, communication will be with a device that has no ground other than the ground provided by its USB cable. This cable may stretch several metres, and so may have significant resistance—enough to drop 250mV over its length.[45] This causes ground potential at the sink's end to rise, and so when the sink drives 1.125 volts (logic high) above its ground it may be driving at as much as 1.375 volts above the source's ground. This is noted in Appendix D to USB-PD revision 3.0 [9], which discusses DC bias-immune receiver implementations.

Mitigating this bias is essential to ensuring that received signals are interpreted correctly. While doing so is straightforward for logic high—simply increasing any maximum voltage limit would be sufficient—the same cannot be said for logic low: a DC bias is already present as a result of Rp and Rd, and so the sink driving the line low to begin transmission will not be detectable as a transition to zero or a transition from zero. In addition, as the current supplied to the sink changes, so too will the DC bias, and so any receiver must have compensation which can dynamically adjust. The easiest solution is to use either of the examples given in USB-PD revision 3.0. However, as those examples are intended to function at either end of the cable (and so with positive or negative bias), it may be possible to make cost- or complexity-reducing modifications.

5.1.2 Synchronisation

Although a USB Type-C cable has 20 wires, none carry a clock. As interpreting BMC uses line state at set intervals, a receiver must be able to synchronise itself with a transmitter opposite in the absence of a clock. Every USB-PD frame includes a preamble for this purpose, and transmitters are limited to deviations in bit rate of 0.25% within a frame.[46] Even so, a receiver design will need to be carefully considered to avoid frequency and phase differencing accumulating to cause desynchronisation over frames of hundreds or thousands of symbols.

[45] This is permitted by BS EN IEC 62680-1-3 [12] at § 4.4.1.

[46] See USB-PD revision 3.0 [9] at § 5.8.5.

5.2 Transmitter design

There is little complexity to a BMC transmitter. Unlike the receiver, it need not be concerned with raised ground potentials or synchronising with its opposite at the sink end of the cable. Although there are limits on implementation details such as slew rate and bit rate deviation, in large part the transmitter is a simple machine shuffling data from a buffer to a driver.

The design has two key parts: the control logic and the line driver.

5.2.1 Transmitter control logic

The BMC scheme is straightforward, as 2.2.3 shows. Although the additions made by USB-PD reduce its simplicity somewhat, those changes are minor and require little new logic. A state diagram showing the basic operation of a BMC transmitter is given in figure 6.

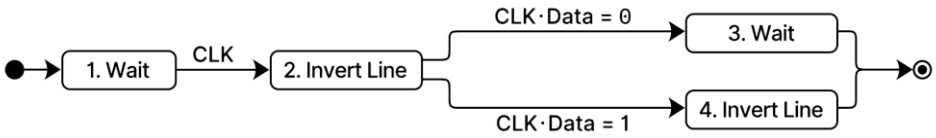


Figure 6. State diagram of basic BMC transmitter operation.

This is the idealised 'core' logic of a BMC transmitter, but is not enough alone to implement a transmitter. USB-PD requires that the line is first driven low, and that it holds its state for a period at the end of a transmission as described in requirements 3.3.15 to 3.3.20. Figure 7 gives a more complete representation, where `Transmit` is a composite state containing those shown in figure 6.

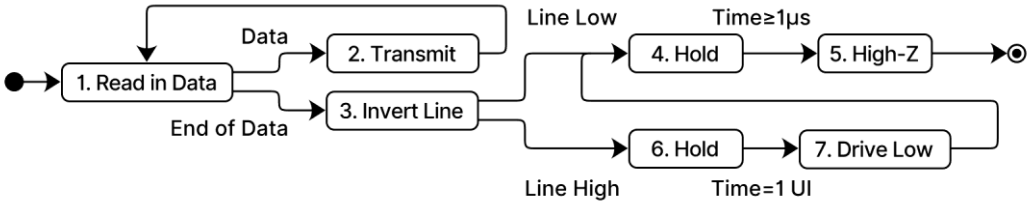


Figure 7. State diagram of a USB-PD BMC transmitter's operation.

In a concrete implementation, the machine in figure 7 would be part of an even larger state machine responsible for interfacing with data-producing parts of the system, a Type-C signalling system (see chapter 7), and a line state manager which ultimately controls the line driver in 5.2.2.

In component terms, the line state can be implemented with two simple logical operations: at the first clock pulse in a unit interval the new line state is the

inversion of the old and, at the second pulse, it is the exclusive-or of the line state and the data to be transmitted.[47] Data might be fed to the transmitter by a first-in first-out (FIFO) buffer, and a simple counter might implement the timing for the end-of-transmission logic.

In terms of control signals, a 'buffer empty' signal would be needed from the FIFO to indicate the end of the data stream, and the line state manager would require 'toggle' and 'set high-impedance' control signals. Depending on the wider system, a 'line idle' signal may also be needed to ensure transmissions do not collide. Organisationally, handling collision avoidance at a higher level may make more sense—if the transmitter is instead only instructed to transmit when the line is idle, its logic can be simplified.

5.2.2 Transmitter line driver

As a single wire is used for communication, the line driver must have three output states: logic high, logic low, and high impedance. This allows the source to stop driving the line when the sink is transmitting. The transmitter must also limit its slew rate to no faster than 3 volts/µs,[48] and present specified capacitance and impedance characteristics to the line.

The most obvious choice is to use a dedicated tristate buffer such as the 74AUP1G126 [14] (6.49p at 3ku). This provides the necessary functionality in a single component, simplifying circuit design. Next most obvious might be a push–pull driver with complementary transistors. Paired P- and N-channel FETs are available in a single package in products such as the Rohm VT6M1 [15] (4.52p at 8ku) and ON NTZD3155C [16] (5.21p at 4ku). Trade-offs exist for each: the complementary FETs are less expensive but require additional resistors, while the dedicated buffer's tolerances mean it is not guaranteed to drive at the logic-high voltage USB-PD requires.[49] These tolerances remained even for higher-cost components. Comparatively, FET voltage drop is easily calculated as $I_D \cdot R_{DS(ON)}$, where drain current I_D will be low due to the receiver input impedance.

Unfortunately, several factors make a push–pull driver unsuitable. If gate signal timing is not precisely controlled, both push and pull FETs may conduct at once and produce a short circuit. Current-limiting resistors could mitigate this, but

[47] Consider the exclusive-or $A \oplus B$: $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$. If A is taken as the previous state of the line and B the data, it can be seen that when the data is zero the previous state is maintained and when the data is one the new state is the inversion of the previous. This is the same logic used to determine the line state in the middle of a BMC unit interval.

[48] The time between driver output reaching 10% amplitude and 90% amplitude is required to be at least 300ns. The same is true in reverse (*i.e.* 90% to 10%). As nominal logic high is 1.125 volts, this means a rise from 0.1125 volts to 1.0125 volts in 300ns ($\Delta = 0.9$ volts). The rate 900mV/300ns is equivalent to 3 volts/µs: see USB-PD revision 3.0 [9] at § 5.8.5 and requirement 3.3.23.

[49] See the datasheet [14] at Table 7, where high-level output is '$V_{CC} - 0.1$' for a 20µA drive current. A drop of 100mV from 1.125 volts reduces output below the 1.05-volt minimum.

their addition risks violating the transmitter impedance requirements as well as forming unintended voltage dividers. Added to this, control of the slew rate for a push–pull arrangement is not easily attained: an RC output filter cannot meet the impedance requirements,[50] and slowing FET turn-on time is complicated by gate charge Q_G which is inconsistent and, at low drain–source voltage, non-linear [17].

To remedy issues with output impedance, a typical solution would be to buffer the output. Applied here, this would require a switching element after the buffer to disconnect the buffer's output from the line. A simple element such as a single MOSFET is not suitable as neither the source nor the drain will be held constant, complicating the driving arrangements.[51] However, if a pair of MOSFETs were used—one of each polarity—the polarity of the voltage between the buffer and the line would always be suited to one of the pair. Monolithic devices of this kind are available and are known variously as transmission gates, linear switches, or analogue switches [18]. The 74AHC1G66 [19] (3.63p at 6ku) has a logic-level gate, a low on-resistance,[52] and a low unit cost. Further cost reduction may be possible if the buffer were a spare op-amp from an array used elsewhere.

The buffer and transmission gate facilitate slew rate limiting, but to achieve a limited slew rate requires circuitry before the buffer. The simplest choice is an RC filter, exploiting a capacitor's charging characteristic to limit slew rate. The design of an appropriate filter is not an involved process but does require a method different from that typical—where a low-pass RC filter is designed for cut-off frequency, a slew rate limiter instead ensures the capacitor charges to a desired voltage within a specified time (or longer, within reason). This requires a rearrangement of the capacitor voltage formula shown below as equation 1.

$$V_C = V_S\left(1 - \mathrm{e}^{-t/RC}\right) \tag{1}$$

The supply voltage V_S is 1.125 volts, and V_C is the voltage the rise time to which is to be limited. Using the 90% point of 1.0125 volts enables easy alignment with the requirements of USB-PD. The time t is the slew rate-limited time to reach V_C, which is 337.5ns for 1.0125 volts. This leaves R and C, either of which could be chosen to enable calculation of the other. A resistance of 680 ohms was chosen arbitrarily from the preferred series,[53] giving a capacitance of 216pF or 220pF if rounding to the preferred series. However, the curve of a capacitor's voltage during charging is nonlinear, and so equation 1 must be rearranged for t to allow

[50] To stay within the 600pF capacitance limit, it would be necessary to use a 560pF capacitor and a 262-ohm resistor. At 750kHz, this is an output impedance of 215.6 ohms for a 33–75-ohm limit.

[51] The output of the buffer will swing between ground and logic high, while the line voltage will rest at the level fixed by the Rp–Rd resistances. Any gate voltage would need to change with these.

[52] *Per op. cit.* at Table 8, 15 ohms typical and 40 ohms maximum when supplied at 5 volts.

[53] See BS EN 60063 [54]. A value of 680 ohms is included in series E6.

verification of the time between the 10% and 90% points. For values of 680 ohms and 220pF and a tolerance of ±5%, worst-case rise time and slew rate are 297ns and 3.03 volts/µs. This does not comply with USB-PD and so charging must be slowed. The next available preferred-series value is 750 ohms, which would give a worst-case rise time of 327ns and so a worst-case slew rate of 2.75 volts/µs.

In simulation, this gave the desired results: the 0.1125-volt 10% point is reached around 18ns and the 1.0125-volt 90% around 380ns, giving a rise time of 362ns and slew rate of 2.5 volts/µs between those points. Using the same ±5% tolerance for both R and C, the full range is 327–400ns and 2.25–2.75 volts/µs.

5.3 Receiver design

The receiver is the more complex part of a BMC transceiver. Not only must it cope with transmitted noise, offset grounds, and line coding, it must also establish and maintain synchronisation over messages of thousands of symbols.

The receiver can be split into three general components: the synchronisation logic, the decoding logic, and the line instrumentation.

5.3.1 Synchronisation logic

The synchronisation of a receiver to an incoming signal is essential for ensuring that data can be decoded correctly. Various methods exist, depending on the interface: UARTs typically have a pre-agreed data rate and frames short enough that phase difference does not accumulate, and so simple oversampling may be used;[54] for a higher-speed interface such as PCI Express, oversampling is not feasible. The same is true where the data rate is arbitrary and not pre-agreed. In these cases, a phase-locked loop (PLL) may be used instead. A PLL adjusts the frequency of a local oscillator to reduce phase difference with an incoming signal to near-zero and continually retunes to maintain the relationship [18].

Neither technique is particularly suited to USB-PD. Simple oversampling would see phase difference accumulate over long messages unless a greater cost is paid for a more stable oscillator, while even a low-cost PLL will add significantly to the cost of the receiver. The most cost-effective solution would be entirely digital, exploiting the additional processing hardware facilitating the BFSK scheme.

Literature specific to USB-PD is scarce but, as 2.2.3 notes, the S/PDIF audio standard shares use of the BMC scheme. A technique useful for one standard is likely to be useful for the other. This in mind, it was possible to identify an expired patent by Adams [20] which uses entirely digital techniques to lock to an incoming S/PDIF data stream. Based around four components—an edge detector,

[54] Universal asynchronous receiver–transmitters (UARTs) commonly provide line-coding for RS-232 line drivers such as the MAX232 [55]. They often oversample the line and use the most common value observed (a ‘majority vote’ scheme): [56] at § 19.6, [57] at § 16.3.2.4.3, and [64] at § 22.3.9.2.

a shift register, a counter, and a 'digitally-controlled oscillator' (DCO)—the design relies on S/PDIF packet preambles having an invalid overlong BMC sequence to tune the DCO.[55] The frequency of the DCO is controlled by the counter, which counts up periodically and down when an overlong sequence is detected, or *vice versa*. If the shift register is cleared on each input edge and a *1* is clocked in on each DCO cycle, it becomes possible to measure the length of pulses by the extent to which the shift register fills between edges.

Building on this, if an overlong sequence is detected by observing when an arbitrary position within the register becomes *1*, too fast a DCO frequency will produce more down-counts, slowing the DCO, while too slow will result in DCO frequency gradually increasing. The system will reach equilibrium where the shift register fills to the overlong position and no more for an overlong sequence, and so the DCO frequency will be related to the data rate.

The overlong sequence the system is now detecting covers three clock cycles, while the unit intervals (UIs) it needs to detect cover two. In effect, the system is detecting periods of 1.5 UI when it needs to detect 1 and 0.5 UI. If the overlong position is bit 15 within the shift register, it is clear that the shorter periods can be detected by observing bits 10 and 5. If the state of each position is evaluated whenever an edge is detected, the system can determine the length of the last pulse and hence what data was transmitted. Advantageously, the system should be entirely insensitive to phase difference accumulation in the input if the clock used to sample for edges is fast enough that multiple cycles occur within 0.5 UI.

In applying this design to USB-PD, some changes are required. Unlike S/PDIF, a USB-PD packet does not include an overlong sequence. Instead, its preamble is an alternating sequence of *0* and *1*, and it is this that would be used in tuning.

If the overlong position were moved closer to the 1 UI position, the frequency of the DCO would be tuned so that 1 UI filled the shift register past the 1 UI position without passing the overlong position. This would move the 0.5 UI position, but it would remain generally halfway between 1 UI and bit 0 in the register. However, while this solves one problem, it leaves another open: S/PDIF packets are short and so overlong sequences occur regularly, meaning the feedback constraining the DCO frequency is also regular. USB-PD packets, on the other hand, are long and relatively infrequent, requiring a different DCO-tuning mechanism.

An advantage of USB-PD in this regard is that its long preamble of 64 UIs allows a long time for tuning before data is received. Added to this, its repeating pattern affords a receiver many opportunities to adjust, compare, and adjust again. With a frequency space restricted to 540–660kHz, a binary search through the space would be a straightforward and inexpensive solution. The essence of a binary

[55] See BS EN 60958-1 [11] at § 4.3.

search is that a comparison is made such that the search space can be halved on each comparison. The result is that, for a space of N frequencies, the number of comparisons required for a successful search C_N is given by equation 2.[56]

$$C_N = 1 + {}^{L_{intl}}/_{N} \qquad L_{intl} = \sum_{k=1}^{N} \log_2 k \tag{2}$$

This is equivalent to the technique applied by successive-approximation-register analogue-to-digital converters (SAR ADCs). Similarly, it fixes worst-case tuning time based on the search space. For USB-PD, assuming a comparison only every second UI of the preamble (*i.e.* 32 comparisons), the potential frequency space is small enough that no time pressure will exist. For example, N = 1,048,576 gives $C_N \approx 19$. This confirms the technique's applicability here.

Figure 8 illustrates the general design of the system with these changes.[57]

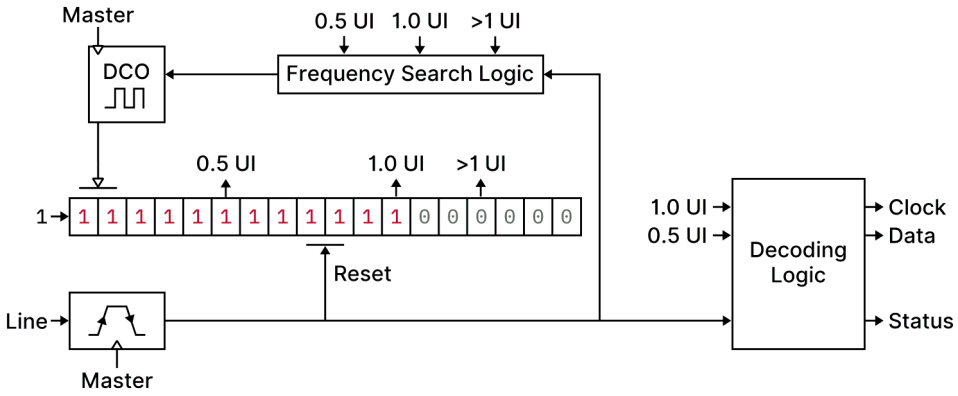


Figure 8. High-level design of the BMC receiver synchronisation logic.

In a practical system, the '>1 UI' position might be moved forwards to provide additional tolerance for frequency deviation. For the same reason, the 0.5 UI position might be moved backwards from its halfway point. As Adams discusses, an alternative implementation could replace the shift register with a counter.

5.3.2 Decoding logic

Once synchronisation completes, decoding the output of the system in 5.3.1 is simple. Both the data and the data clock can be recovered by evaluating the 0.5 and 1.0 unit interval taps on each transition in the manner shown in figure 9.

[56] See Knuth [65] at p. 413 for the average number of comparisons in a binary search and Knuth [66] at p. 401 for the calculation of a minimum value of internal path length. The equation above gives the number for a successful search. Knuth also discusses the case of an unsuccessful search, but this is not relevant here—the BMC receiver requires only the closest frequency possible.

[57] This diagram is derived from that given by Adams [20] at Figure 5.

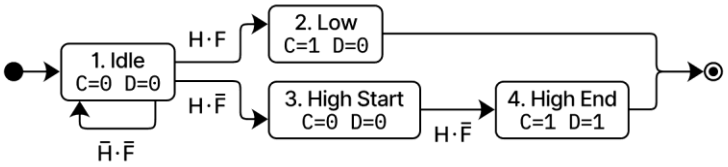


Figure 9. State diagram of the BMC receiver decoding logic's operation.

In this diagram, H represents the 0.5 unit interval tap and F the 1.0 tap. In the states, D is the data output and C indicates when D is valid. A combination of H and F other than one given as a transition condition results in a transition to an error state (for example, $\bar{H} \cdot F$ in state 1 and $H \cdot F$ in state 3).

5.3.3 Line instrumentation

The USB-PD specification [9] discusses in its Appendix D two implementations of a BMC receiver. In each case, the general concept is to rely on relative changes in the signal so that transitions can be detected without DC bias having an impact. If the change exceeds a positive or negative threshold, it is indicative of a signal edge. The difference between the two methods is in how they determine the relative change in the received signal.

The first method, called the 'finite difference scheme' by USB-PD, samples the received signal at a fixed interval (50ns) and keeps a rolling record of samples. It subtracts from its current sample's value the value of a sample taken a specified period earlier, as in equation 3.

$$\Delta V_{CC} = V_{CC}(t) - V_{CC}(t - \Delta t) \tag{3}$$

In equation 3, Δt is that specified period. USB-PD recommends 300–500ns.

The second method—referred to as the 'subtraction scheme' by USB-PD—works in a similar manner, using a pair of low-pass filters feeding into a difference amplifier to produce its output. The differing time constants give a slight phase difference between the two filters, imitating the finite difference scheme. As can be seen in Figures D-5 and D-9(b) from USB-PD, the subtraction scheme's results are highly similar to those of the finite difference scheme.

Given this similarity and that the subtraction scheme appears the simpler of the two to implement, it was selected for further investigation. In simulation, the filters and subtractor were implemented with three op-amps, six resistors, and two capacitors. Two op-amps served as unity-gain buffers, while the third was used to implement a unity-gain difference amplifier (that is, a subtractor). The output of the system is shown in figure 10, which illustrates its resilience to malformed signals and compares favourably with illustrations in USB-PD.

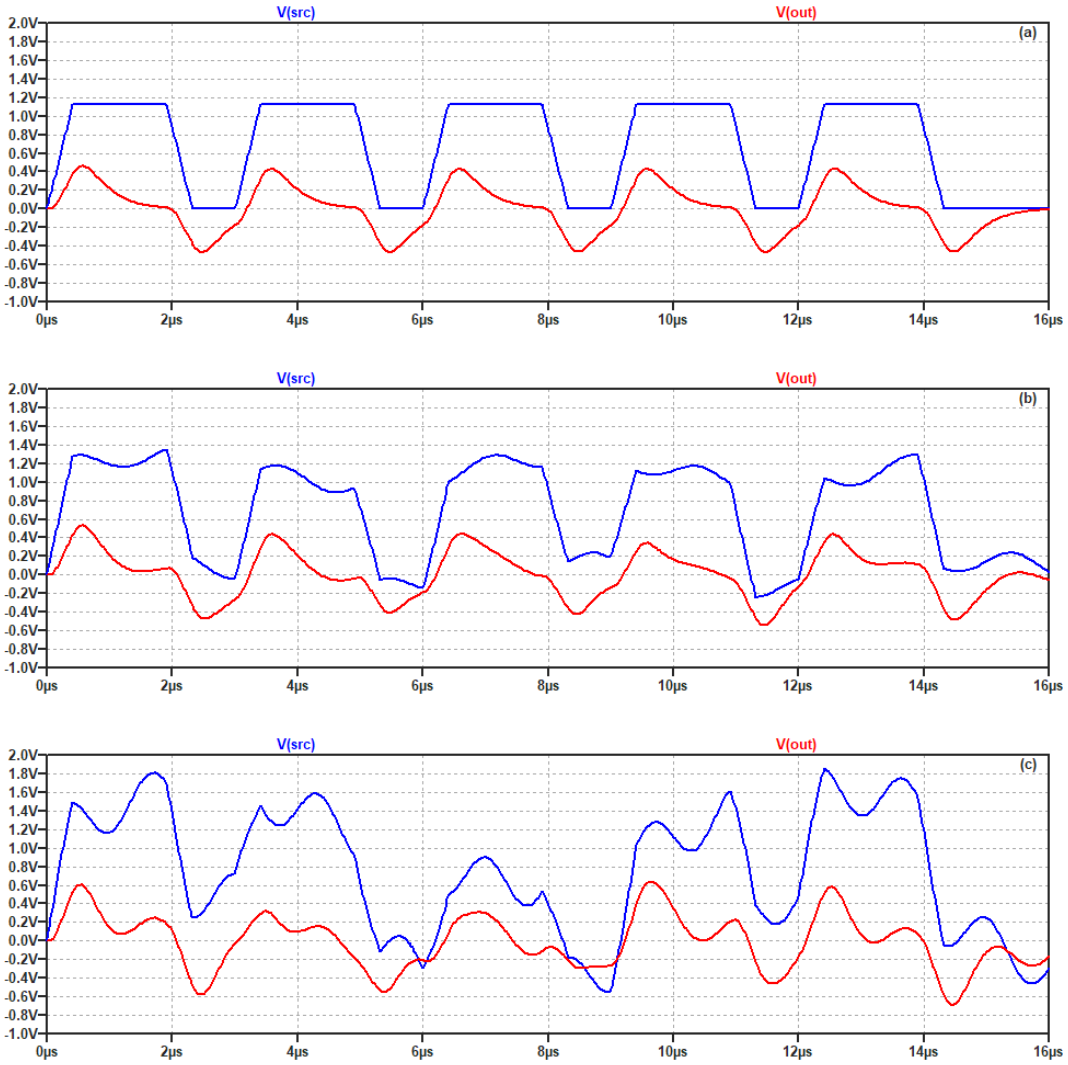


Figure 10. *The output of the USB-PD receiver subtraction scheme for various levels of noise-affected input.*

Subtractor output (red) is shown for various inputs (blue), where input is (a) a 1.125-volt square wave of period 3µs; (b) that square wave modulated with 0.15sin(2π·150kHz·t) + 0.1sin(2π·600kHz·t) volts; (c) that square wave modulated with 0.5sin(2π·100kHz·t) + 0.25sin(2π·750kHz·t) volts.

Comparing figure 10(a) and (b) shows that, at relatively minor noise levels, the subtractor's output changes little. In figure 10(c), although the more pronounced noise means output is no longer recognisably a triangular or sawtooth shape, the basic information can still be extracted—at its most muted, output crests above 300mV during square wave high[58] and below −280mV during square wave low. At these levels, with the thresholds of ±250mV used in the USB-PD examples, the

[58] That is, the period where the unmodulated square wave would be at logic-high voltage.

voltage level comparator would produce the correct square wave output. As the noise in figure 10(c) is far beyond the levels expected, the subtraction scheme appears a suitable implementation method as-is.

As the subtractor output is based on relative change, a fixed voltage comparison would not work. While the output might peak above a threshold, the decrease observed at level voltage would require an impractically low threshold to keep the comparator's output high. More appropriate, as USB-PD suggests, would be separate thresholds where a change in state requires the opposite threshold to be crossed. This dependence on previous state, or hysteresis, makes a Schmitt trigger the device of choice. Its operation is illustrated in figure 11.[59]

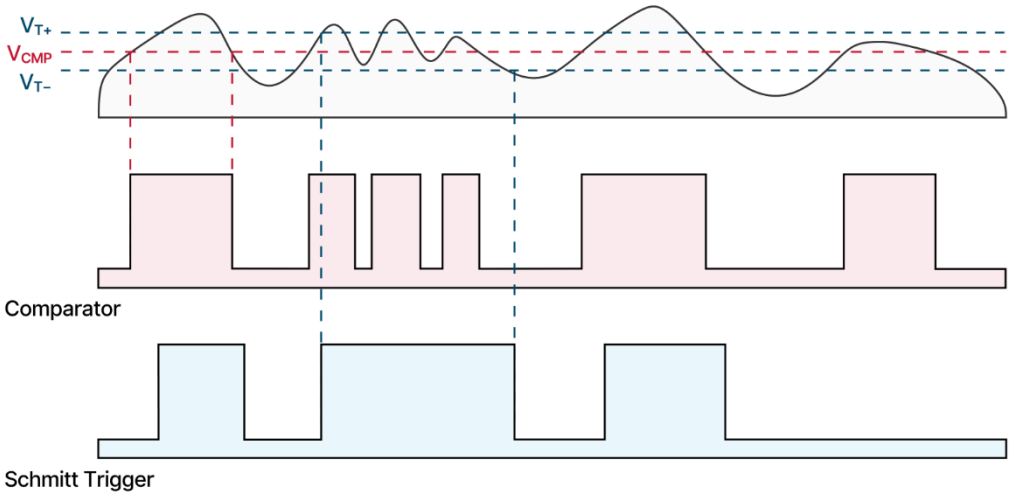


Figure 11. Operation of a comparator and a Schmitt trigger on a noisy signal.

In figure 11, V_{CMP} is the comparator threshold voltage. V_{T+} and V_{T-} are the Schmitt trigger's positive-going and negative-going threshold voltages, respectively.

Hysteresis is induced by feeding comparator output into the reference voltage node, introducing a bias dependent on output. When the output is low, it forces the reference and hence the comparator threshold down. When high, it does the opposite and pulls the threshold up.[60] This action produces the negative-going and positive-going threshold voltages for the Schmitt trigger.

In simulation, applying the simple hysteresis-setting method described by Rohm Semiconductor [21] for targets of +249mV and −241mV, resistances of 47kΩ, 220 ohms, and 810 ohms gave thresholds of +246mV and −238mV. Increasing these

[59] This figure is based on Figures 4.31 and 4.32 from Horowitz and Hill [18].

[60] See Horowitz and Hill [18] at p. 237.

resistances would reduce the extent to which voltage is dropped but could result in instability if the capacitances inherent to the chosen device form a filter with a cut-off frequency below the maximum signal frequency. Figure 12 shows the output of that Schmitt trigger for the input waves from figure 10(a) and (b).

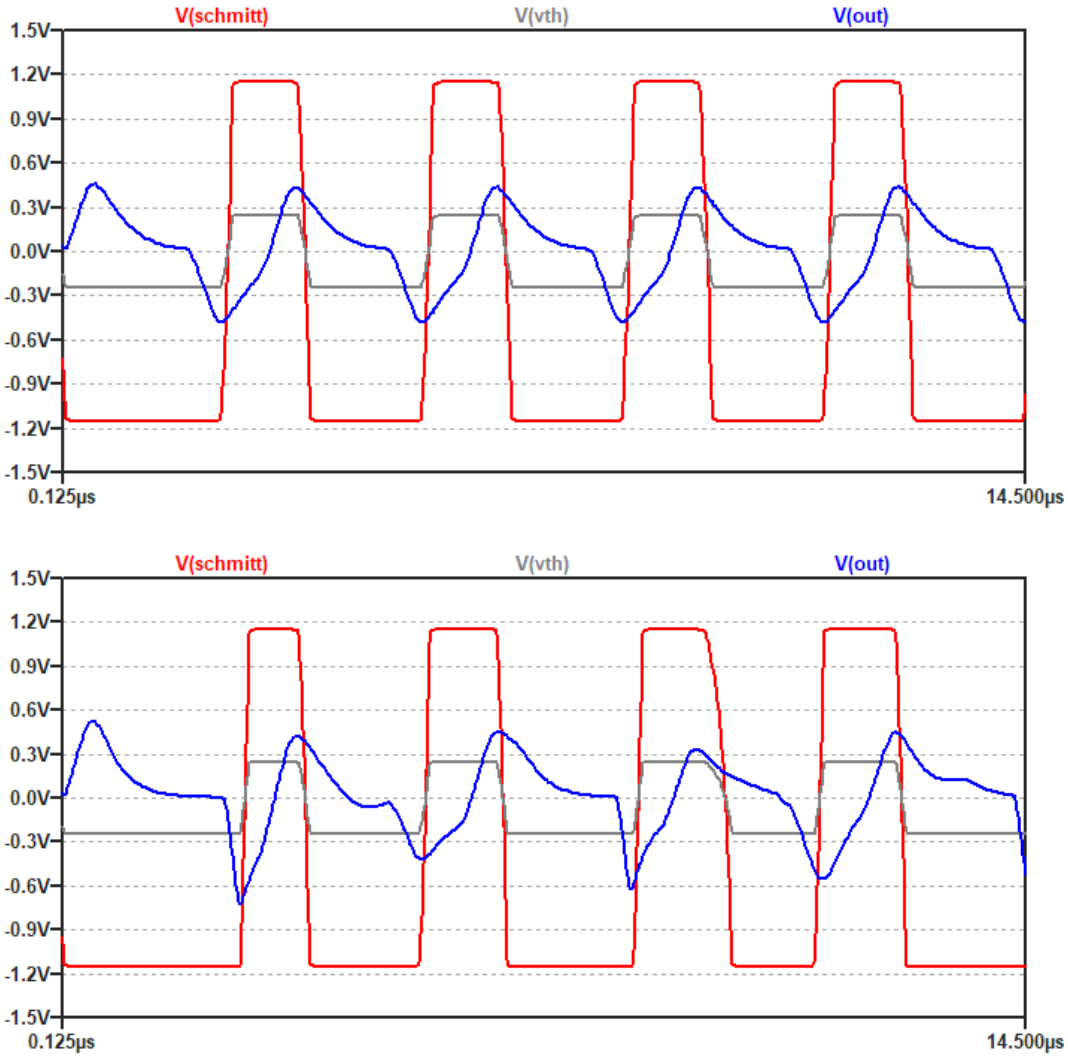


Figure 12. Schmitt trigger operation on BMC receiver subtractor output.

The output of the Schmitt trigger (red) is shown for various subtractor outputs (blue), where the input to the subtractor is the same as in figure 10(a) for the top plot and figure 10(b) for the bottom. Schmitt trigger threshold voltage (grey) is also shown.

This figure demonstrates a Schmitt trigger's application and shows how the threshold voltage is biased by the output. The wave from figure 10(c) was also fed into the trigger, but with limited success. As that input is thought to represent noise far greater than will be encountered, its failure has limited relevance.

In addition, figure 12 shows the importance of slew rate to the circuit's correct operation. The threshold changes with the output, and so the output must change fast enough that a steady state is reached before the input passes its peak. This ensures that the device can be triggered by the peak and that it is not triggered mid-transition. As simulated, the shortest peak persists for 378ns. An output voltage of ±1.15 volts (for 2.3 volts swing) would necessitate a slew rate no lower than 6.1 volts/µs, all other effects ignored. Simulating again, reliable operation was attained at 20 volts/µs. Lower than this, and the output either resembled a triangle wave—indicative of limited a slew rate—or showed that the comparator occasionally failed to trigger when positive peaks were brief.

That said, a high slew rate is only necessary for the Schmitt trigger portion of the line instrumentation. The first low-pass filter limits bandwidth to 882kHz in the worst case which, for a sine wave at that frequency and with amplitude 1.2 volts, results in a maximum slew rate of 6.7 volts/µs. The USB-PD specification limits slew rate to 3 volts/µs, and so a real signal has relatively lax requirements. Given this disparity in requirements, the Schmitt trigger may be more economically implemented with a different device, allowing the devices selected for use in the subtractor to be less performant and correspondingly less expensive.

Figure 12 also shows the inverting nature of the Schmitt trigger—crossing the more positive threshold results in it driving low, and *vice versa*. As simulated, the result of this is that the first pulse is lost. Although a receiver must tolerate the loss of the first edge[61] and so should not be affected, this is not expected to occur in a real circuit. The USB-PD pull-up resistors will cause the line to hold at 0.85 to 2.45 volts when undriven, and so the sink initially driving the line low will produce a falling edge sufficient (in simulation) to pass the more negative threshold. If a DC bias of 250mV is introduced, this remains true provided the line idles at least 0.93 volts.

Note that the mark–space asymmetry in figure 12 is largely the result of slow rise times in the input signal, visible in figure 10(a). When inverted, the wider high level in the input gives a wider low level in the output.

[61] See USB-PD revision 3.0 [9] at § 5.8.1.

6. Binary Frequency-Shift Keying Transceiver

This chapter covers the binary frequency-shift keying (BFSK) transceiver that would be required to support USB-PD for devices with USB Type-A and Type-B connectors. The viability of including the BFSK transceiver is to be investigated to fulfil objective 1.3.2. Its requirements are set out in section 3.4.

6.1 Key concerns

6.1.1 Speed of operation

The high frequencies the BFSK scheme uses are likely to present issues in both transmission and receiving. The nominal 23.2MHz carrier is far faster than the 600kHz data clock used in the BMC scheme and so simple integer clock division cannot produce it and the signalling frequencies without an impractically fast master clock. The cost digital synthesisers is likely to be prohibitive. In receiving, the wide tolerances involved will restrict the choice of design: received signals may cover a 0.9–1.2MHz band within the range 21.8–24.6MHz,[62] and so the precise carrier frequency cannot be known ahead of time.

6.1.2 Synchronisation

As with the BMC scheme, the BFSK scheme provides no clock and requires a receiver to derive its own from the received signal. However, as the BFSK scheme does not use a line coding to guarantee a minimum number of transitions, any synchronisation is made more complex—the sequence *000111*, for example, has eight transitions with BMC but only one under BFSK. This may exclude PLLs and other techniques that demand a minimum level of feedback to maintain a lock.

6.2 Transmitter design

The concept for the BFSK transmitter is straightforward—connect one of three frequencies to V_{BUS} in each data period, ensuring phase discontinuity between each is minimised. This section considers its design in two parts: the modulator which produces the signalling frequency and its control logic.

6.2.1 Modulator design

The modulator is responsible for translating digital data into the frequency shifts used in transmission. An obvious realisation would be a set of three oscillators multiplexed onto the line as required, but this is complicated by the requirement to maintain phase continuity—over most of their period, two waves of different

[62] See requirements 3.4.6 to 3.4.8.

frequencies will be out of phase. This problem is exacerbated with the addition of a third frequency. If not mitigated, distortion will occur at the switchover point as the line voltage instantaneously jumps up or down.[63] One mitigation would be to use so-called 'orthogonal' frequencies, which are determined as in equation 4 so that phase difference between them returns to zero over a data period T.[64]

$$f_{out} = f_c \pm {}^{\Delta f}/_{2} \qquad \Delta f = {}^{k}/_{4T} \tag{4}$$

Here, f_{out} is the output frequency and f_c the carrier.

This is exploited in 'minimum shift keying' (MSK), a form of BFSK (or n-FSK). By maintaining continuous phase and using $k = 1$ for the narrowest possible spread of frequencies, the bandwidth required is minimised. In the current application, values of $T = 3.33\mu s$ (for a 300kbps data rate) and $k = 12$ give $f_c \pm 450kHz$—that is, the minimum deviation USB-PD allows. As such, the technique is workable for USB-PD. As a visual aid, figure 13 illustrates orthogonality at this value of Δf.

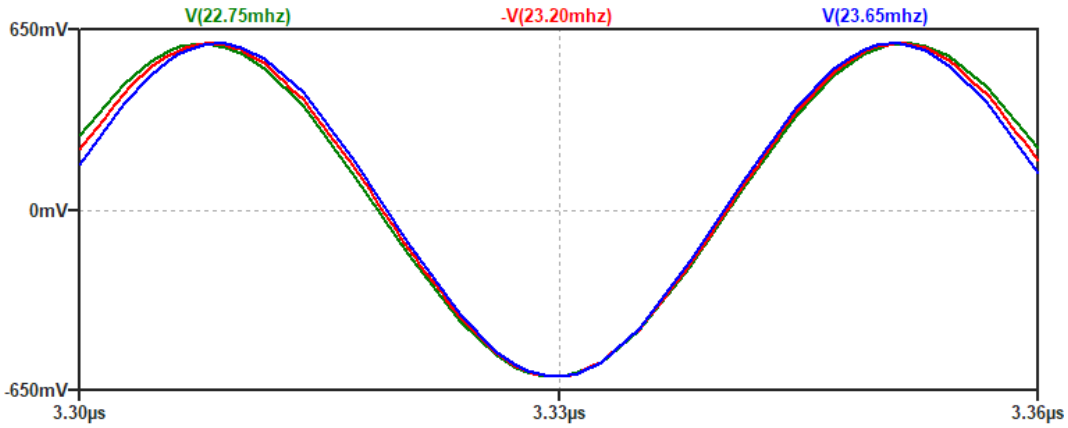


Figure 13. The phase difference of 23.2MHz±450kHz waves returning to zero at the end of a data period of 3.33µs.

Note that the carrier is 180° out of phase and so has been inverted.

Despite its simplicity, this method has substantial drawbacks: non-uniformity in the start-up times of free-running oscillators would prevent phase difference from returning to zero, and so at least three PLLs are required. As a data clock is also needed, a fourth PLL may be necessary if a crystal or other device in a target frequency cannot be obtained. This would significantly increase unit cost.

Further, if a 23.2MHz reference is used, precise ratios of 0.9806 and 1.0194 are

[63] Refer to footnote 26 on page 15.

[64] *Per* Goldsmith [58] at § 5.4.1.

needed to obtain the signalling frequencies. For a divide-then-multiply PLL,[65] obtaining these ratios accurately requires division by such a large value that the phase comparator is likely to be starved of input, as table 2 shows.[66]

Table 2. PLL divisors and multipliers for a four-oscillator BFSK modulator.

Accuracy	Divisor	Multiplier	Phase comparator input
f_{out} = 22.75MHz			
2 d.p.	103	101	225.2kHz
3 d.p.	567	556	40.9kHz
4 d.p.	5000	4903	4.6kHz
f_{out} = 23.65MHz			
2 d.p.	52	53	446.2kHz
3 d.p.	2732	2785	8.5kHz
4 d.p.	5000	5097	4.6kHz

This means that, while it may be worthwhile investigating the true performance of a four-oscillator system, it is necessary to explore other options. For example, if one oscillator can be made to smoothly transition between frequencies, there will be few or no discontinuities without the expense of many PLLs.

If a tuned circuit were used instead, a fixed tuning element could be replaced with a variable one. Circuits with more than a single tuning element are likely to prove unwieldy to control, and so can be ignored. Crystal oscillators have their frequencies fixed by their mechanical dimensions, limiting their range to far less than the nominal ±500kHz required. Moreover, any circuit which would rely on varying an inductance can be disregarded—typically, inductances are varied by moving the magnetic core within the coil, which is not practical electrically. It may be possible to use a switched bank of inductors, but the need for protection against inductive kickback means this technique makes little sense as a choice when capacitance or resistance can be switched without protection. Ideally, too,

[65] As the name suggests, 'divide-then-multiply' PLLs first divide the input frequency and multiply the result to produce the output frequency. Division is straightforward: a modulo-n counter on the PLL input. Multiplication places a modulo-r counter in the feedback loop from the PLL's variable oscillator, reducing the frequency apparent to the phase comparator. It commands an increase in frequency to match the reference, giving $f_{out} = f_{in} \cdot {}^{n}/_{r}$. A further modulo-$m$ counter may be present on the output, enabling lower values of n and r so that the phase comparator is fed more edges to use in continually retuning [18].

[66] These values were identified by exhaustive machine search through $M = 0.9806 \times D$, where D is the integer divisor, for values of M with a fractional part less than 10^{-2}, 10^{-3}, or 10^{-4} as appropriate. In cases where f_{out} = 23.65MHz, 0.9806 is replaced by 1.0194.

a natural sine wave oscillator would be selected. This would avoid the need to filter out the harmonics other waves introduce.

Based on this, a variety of designs can be excluded: relaxation oscillators, which produce triangle, sawtooth, or square waves; Wien bridge oscillators, which have capacitor pairs and resistor pairs, preventing tuning with a single element; and, more broadly, any op-amp-based oscillator owing to the gain–bandwidth product of inexpensive devices limiting oscillation to well under 1MHz. The discrete amplifiers this leaves can be further narrowed, with bipolar junction transistor-based (BJT) types disregarded due to the inherent 0.7-volt forward voltage that is likely only to complicate design. Common FET-based types include the Pierce, Colpitts, Hartley, and Clapp oscillators, each of which uses a resonant LC network to fix its frequency. From these, the Hartley type can be ignored—it uses coupled inductors which, being less common, increase its cost. Circuit diagrams for the other types are shown in figure 14.[67]

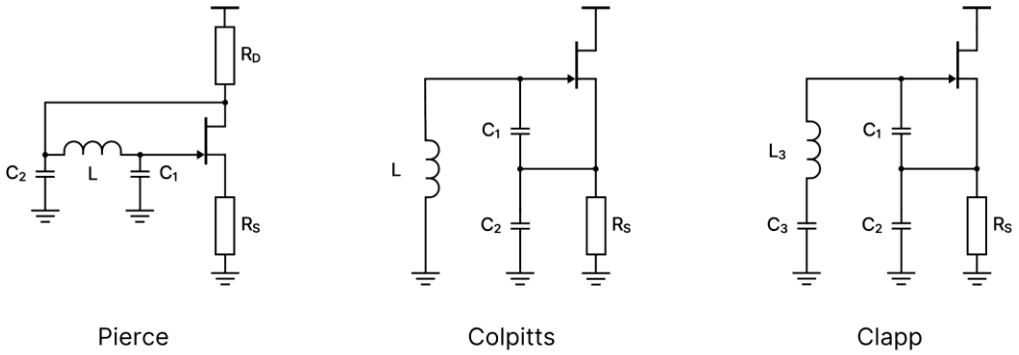


Figure 14. Comparison of topologies of LC sine wave oscillators.

Topologically, the Colpitts oscillator is the simplest, followed by the Clapp and then the Pierce oscillator. As would be expected for a resonant LC network, the oscillation frequency is that given in equation 5.

$$f_{osc} = {}^{1}\!/_{2\pi\sqrt{L \cdot C}} \tag{5}$$

In the general case, the loop gain required to bring each topology into oscillation is that shown in equation 6.[68]

$$g_m \cdot R_s > {}^{C_1}\!/_{C_2} \tag{6}$$

[67] These diagrams are based on Figures 3.3, 3.5, and 3.10 from Gonzalez [59].

[68] This condition is identified by Neamen [61] at p. 1083.

This condition must be considered in designing the oscillator to ensure that it both starts and operates correctly.[69] In particular, for ease of control, the tuning element should have minimal effect on loop gain to avoid the risk of the gain becoming too low or too high to sustain oscillation. The Pierce oscillator has no point where another capacitor could be inserted without affecting loop gain. For the Colpitts and Clapp oscillators, a capacitor could be inserted parallel to the inductor without affecting C_1 or C_2. The choice between Colpitts and Clapp can then be one based on simplicity: the Colpitts design has the simpler topology and, with no other clear advantages for either, that makes it the better choice.

To tune the oscillator, a capacitor bank would likely work. However, the fineness of the switched capacitances would be limited, and so the potential for abrupt changes in output remains. Variable capacitances are available as components in the form of 'varicap' or 'varactor' diodes; using the device's terminals as parallel plates, the widening of the p–n depletion region with reverse bias gives a varying capacitance controlled by a DC voltage [18]. If that voltage is varied smoothly, it follows that capacitance—and hence output frequency—will do the same. Added to this, the straightforward circuit and parts count make a tuneable Colpitts oscillator an attractive alternative to the four-oscillator design.

6.2.2 Control logic: four-oscillator design

In terms of control, a design where oscillators are switched onto the line as needed is conceptually very simple. A general diagram is given in figure 15.

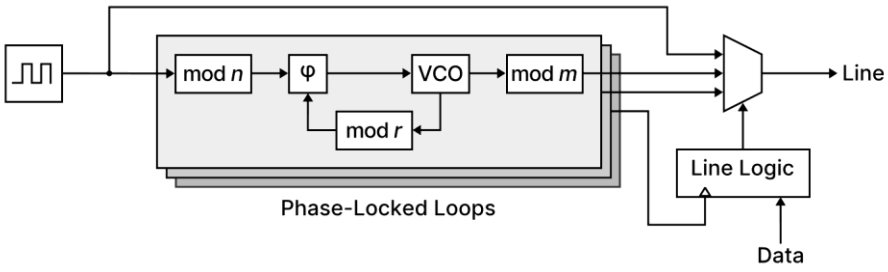


Figure 15. High-level design of the four-oscillator BFSK transmitter.

The reference oscillator and PLLs shown operate autonomously. Two of the PLLs produce the signalling frequencies, while the third produces a data clock. When

[69] Gonzalez [59] identifies another condition ($g_m > [2\pi \cdot f_{osc}]^2 \cdot R_s \cdot C_1 \cdot C_2$) with his equations 3.28 and 3.29 on pp. 116–117. It is not clear how this condition is intended to be used: typical JFETs have low transconductances, but applying values of 100.8MHz, 90pF, 106pF, and 68 ohms gives 260mS. Simulating with an MMBFJ309LT1 [60] ($g_m \leq 18$mS), the condition from Gonzalez is obviously not satisfied. Yet, noting that Neamen's condition is satisfied, the simulated design works. Neamen's condition is also described in other literature [62].

there is data to be transmitted, the line logic uses the data clock to evaluate its data input and reconfigure the multiplexer. In addition, at the start and end of a transmission, as requirements 3.4.12 and 3.4.13 set out, the line logic switches the reference oscillator onto the line. This need not occur on a data clock edge.

This conceptual simplicity is matched by simplicity in the state diagram for the line logic, as figure 16 shows.

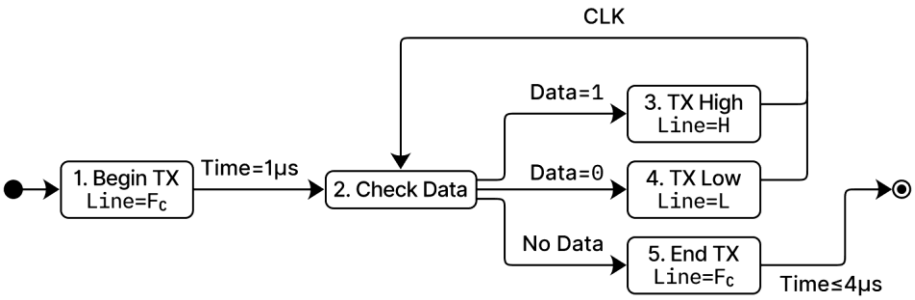


Figure 16. State diagram for the four-oscillator BFSK transmitter line logic.

In the states, *Line* indicates which oscillator is switched onto the line, and the value should be taken to persist until changed in another state. The value F_C is the carrier frequency, and H and L are the logic-high and logic-low signalling frequencies, respectively.

6.2.3 Control logic: Colpitts design

Control over the Colpitts design is achieved by biasing a varactor diode to vary its capacitance, changing the resonant point of the LC network. This requires that a DC voltage is produced which varies fast enough to signal at 300kbps.

At this data rate, a monolithic digital-to-analogue (DAC) converter is unlikely to be economical. Low-cost devices such as the DAC43401 [22] (30.6p at 6ku) have settling times far longer than is permissible: 12µs for the DAC43401, with 25ns for the MAX5190 [23] (£2.34 at 250u) at significantly increased cost. The useful bias region of a typical low-voltage varactor such as the BB145B [24] spans several volts even with a capacitance ratio of 2.2, however, and so other means of producing an analogue voltage—even if low resolution—are likely to be viable.

One such method is the ‘PWM DAC’, a technique which filters the high-frequency components out of a pulse-width-modulated (PWM) signal to give DC output at a voltage proportional to the duty cycle [25]. As a purely digital form of modulation, the PWM signal could be easily generated by the system controller. Inexpensive RC filters could be cascaded through relatively low-performance amplifiers with a final amplifier scaling the output to the useful bias range. For the BB145B, this is from 0.5 volts to 4 volts, and so even 5-bit resolution would be equivalent to a

resolution around 110mV across this range. Assuming a similar characteristic for the chosen varactor and that somewhat greater resolution of six or seven bits is attainable, this is likely to be a suitable technique for biasing the varactor.

As a final note, although the bias curve will vary between devices, this has little effect on the system. The varactor need only be biased to three voltages, which can be established by a power-up calibration routine ahead of time. Once these are established, the controller can switch between the three bias levels without the need for closed-loop control.

6.3 Receiver design

The BFSK receiver, like the transmitter, is simple in concept but more complex in implementation. It works by identifying which of three frequencies is present and observing for how many data periods it persists. This divides into three core functions: demodulation, timing and decoding, and squelch detection.

6.3.1 Demodulator design

The demodulator identifies the frequency of the received BFSK signal and, from that, produces output which can be interpreted by the decoding logic. As a subset of frequency modulation (FM), the methods available for BFSK at first appear to be numerous: in addition to BFSK-specific methods, any technique which works for FM will also work for BFSK, and the widespread use of FM in broadcasting is likely to make a wealth of information available. In truth, this is not the case.

USB Power Delivery allows a BFSK transmitter to operate with loose tolerances. A transmission can have its carrier at any point from 22.4MHz to 24MHz, deviation from which can be 450kHz to 600kHz.[70] The result is that, in general, it cannot be known ahead of time whether a frequency is a carrier or signalling frequency. In these conditions, demodulators based around tuned circuits—such as the slope detector used for FM and deviation-tuned bandpass filters for BFSK—cover too narrow a range to produce useful output. The quadrature detector is excluded for the same reason, although its principle of operation differs.[71] Of the common FM demodulators, this leaves the PLL demodulator—by forcing a PLL to lock to the incoming signal, changes in the control output from the phase comparator will mirror the baseband signal.

Of the designs intended for BFSK, a technique described by Winterer [26] is elegant but similarly defeated by the wide tolerances. If a pair of monostable multivibrators (*i.e.* edge-triggered one-shot pulse generators) are cascaded with

[70] See requirements 3.4.6 to 3.4.8.

[71] Briefly, a quadrature detector produces an output frequency proportional to its input. This is passed through a low-pass filter, which gives attenuation related to frequency. Hence, a frequency to amplitude conversion is achieved: Sobot [28] at pp. 312–315.

their retriggering times set longer than the periods of high and low frequencies in turn, they will extract the inverse of baseband signal as shown in figure 17.[72]

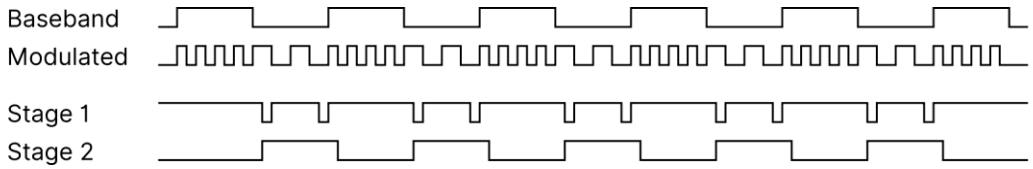


Figure 17. FSK demodulation with cascaded multivibrators.

An alternative technique might simply rectify the received pulses and count the time between. This defers almost all the work to digital logic but, in doing so, it allows easy adaptation to the wide range of input frequencies.

Data can be extracted by taking the total time that input persists at a given frequency and dividing by the data period. In an arbitrary system this would be subject to accumulating error for long sequences of the same bit value, but the 4b5b encoding applied by USB-PD guarantees sequences no longer than eight *1*s or five *0*s.[73] This limits the maximum error which can accumulate. If no effort is made to identify the true data period, error of up to 3µs (or one data period) may accumulate.[74] However, if the USB-PD preamble is used to determine the length of the data period, it is likely that this error can be made negligible.

As with the modulator, investigating multiple designs may be worthwhile. While the time-averaging design is almost fully digital and so offers reduced cost, it may not be practical in a commercialised product. From those considered, the PLL demodulator is the only design which appears suitable. A device such as the CD74HC4046A [27] (16.2p at 5ku) packages both a PLL and a voltage-controlled oscillator (VCO) and, without the high-value counters a modulator would need, it is available at relatively low cost. This specific device has the advantage of being extensively documented.[75] Monitoring the VCO control signal as discussed, the baseband could be extracted in two ways: first, by removing the DC offset and observing for positive and negative pulses [28]; or second, an analogue-to-digital converter could be used to obtain the precise changes in value. If the latter were used, there would be little need for a high-resolution ADC and so cost savings may be possible if comparators and a priority encoder were used in its place.

The BFSK frequencies cover a 2.4MHz band and, with deviation of 450kHz or more, eight bands of 350kHz will ensure permissible deviation always crosses a

[72] This diagram is based on that from Winterer [26] at fig. 2.

[73] See USB-PD revision 2.0 [10] at Table 5-1.

[74] Over eight data periods, maximum period minus nominal is 8×(3.7µs − 3.33µs) = 2.96µs.

[75] See, for example, its datasheet [27] at Figures 16–21 and 33–37.

band boundary and can be detected with three bits of resolution. A pair of LM339 quad comparators [29] (6.18p at 5ku) feeding into an SN74HC148 8-to-3 priority encoder [30] (9.11p at 5ku) with an AN431 voltage reference [31] (4.25p at 6ku) will give a unit cost, excluding the resistor ladder, around 25.72p. Comparatively, an ADC covering the full 5-volt tuning range with reasonable resolution such as the ADC081C027 [32] (50.1p at 5ku) is almost double the cost.

6.3.2 Automatic gain controller

The nominal amplitude of the received BFSK signal is $150\text{mV}_{\text{RMS}}$, with the full permissible range covering $55–300\text{mV}_{\text{RMS}}$.[76] This means that, irrespective of the choice of demodulator, amplification is required before processing. The high dynamic range of 14.7dB, however, precludes simple fixed gain.

Instead, a system known as an 'automatic gain controller' (AGC)—which adjusts a variable gain element to produce the desired output—will be required to produce constant-amplitude output that can be processed. It is illustrated in figure 18.

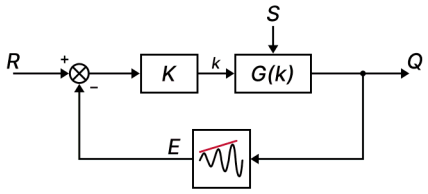


Figure 18. Block diagram of the automatic gain controller.

Here, R represents the reference voltage (for example, 3.3 volts), S is the signal received by the receiver, and Q is the output of the AGC. The $G(k)$ block has gain proportional to its input k from the K block. The input to K is the error in the output, or the difference between the reference R and the envelope E of output Q.

Although significantly more complex than a fixed amplifier, including an AGC allows any processing components to work within a much narrower range of values, and so the overall cost of the demodulator can likely be reduced.

6.3.3 Timing and decoding logic: time-averaging design

The time-averaging demodulator rectifies a BFSK signal and averages the time between pulses to obtain the frequency. Once a frequency is identified, data can be extracted by dividing the time a frequency persists by data period length.

At the BFSK scheme's frequencies, single pulses are too short to measure. The shortest period is 40.65ns and the longest 45.87ns. A timing clock of 100MHz would cycle only four or five times within either period. If its frequency were

[76] See requirement 3.4.9.

increased to 200MHz and the period of 20 pulses measured instead, between 162 and 183 cycles would be recorded and deviation of 450–600kHz would produce a deviation of four or five cycles. This could be exploited by the system in figure 19 to extract data from the received signal.

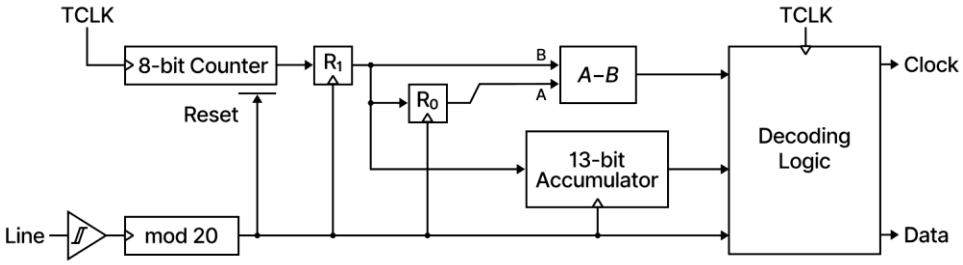


Figure 19. High-level design of the time-averaging BFSK receiver.

Here, a counter records the number of timing clock (*TCLK*) cycles for every 20 pulses in the received signal. The count is shifted through R_0 and R_1 before each counter reset. A subtractor computes the difference between the last two counts, while an accumulator records the total number of counts. Its 13-bit size holds the sum of up to eight counts.[77]

To extract data, the decoding logic monitors the subtractor for a difference of at least eight counts—equivalent to a 900kHz shift from one signalling frequency to the other. Once detected, dividing the accumulator value by the counts that correspond to a data period gives the number of data bits. The value of the bits is indicated by the sign of the difference: positive difference means a longer period and a lower frequency, indicating that the previous frequency was higher and so that the bits are logic high. Negative difference similarly indicates logic low.

The period of a nominally 300kHz data clock will be around 667 counts. The system as drawn could exploit the USB-PD preamble to refine this estimate; the count accumulated in the preamble between each frequency shift will be the data period, which can be averaged over the 64 UIs the preamble provides.

A return to the carrier frequency can be detected by a slighter difference of four or five counts, while line idle would be indicated by the squelch detector.

6.3.4 Timing and decoding logic: frequency–voltage design

The frequency–voltage design for the BFSK receiver uses a PLL and VCO, relying on the PLL to produce an analogue control signal corresponding to the frequency of the received signal. A basic ADC can convert this signal to a digital form which

[77] As 6.3.1 notes, the longest contiguous sequence is eight bits. The 20 pulses take at least 813ns, and so four counts may cover in a 3.7µs data period. Eight bits by four counts of up to 255 gives a sum of no more than 8160, almost 2^{13}.

can be decoded with logic like that for the time-averaging design. The illustration in figure 19 may aid understanding.

In this design, a change in frequency gives a change in ADC output. This enables some simplification—as the logic need no longer determine frequency directly, it can do away with the accumulator and increase the capacity of the counter. When a frequency shift is detected, the counter will hold the duration the frequency persisted. Dividing this count by the data period will give the number of bits and the sign of the change in ADC output gives the value as 6.3.3 describes. As a count covers an entire data period, a slower clock will retain adequate granularity; at 50MHz, counts will be between 151 and 185 depending on the data rate.

Additional logic may be required for when a signalling frequency lies at a band boundary. In this situation, slight control voltage changes might cause the ADC to rapidly alternate between two bands. This could be resolved by logic that ignores a frequency shift unless it persists for a minimum period. Alternatively, a filter on the input to the ADC could remove high-frequency ripple.

6.3.5 Squelch detection

The ‘squelch’ is a system that detects whether a signal above a noise threshold is present, enabling or disabling the BFSK receiver as appropriate [10]. It ‘opens’ if such a signal is present and ‘closes’ when one is not. In a USB-PD source with a Type-C receptacle, its only purpose is to detect whether the line has become idle.

Line idle occurs if no signal between 20.4 and 26MHz is detected on V_{BUS} with an amplitude of 35–55mV_{RMS}.[78] Given the presence of a bandpass filter before the receiver,[79] this could be implemented with an envelope detector and comparator.

[78] See requirement 3.4.19.

[79] See USB-PD revision 2.0 [10] at Figure 5-9.

7. USB Type-C Signalling

In addition to the requirements set by USB-PD, a source will need to comply with the requirements from the USB Type-C specification. Although these generally relate to Type-C cables, a small number are relevant to the current project. This chapter considers the relevant portions.

7.1 Rp–Rd signalling

The Rp and Rd resistors are used by USB Type-C devices to establish whether a valid attachment has occurred, how the USB cable is oriented, and the basic level of current the source can provide. This is discussed more thoroughly in 2.3, and figure 4 provides a diagram illustrating their connection.

When a Type-C connection between a USB-PD source and sink is made, these resistances are also used for flow control.[80] Under USB-PD revision 2.0, either the source or the sink could begin transmitting when the line became idle;[81] to avoid the potential for collisions on the line, revision 3.0 requires that a transmission by the source is preceded by a change in the Rp resistance. This indicates to the sink that it cannot begin a transmission, and so the risk of collision is reduced.

In implementation, switched resistances have a fixed voltage at one node and so do not face the same issues as a switched transmitter, described in 5.2.2. That said, the transmission gates used for the BMC transmitter are low-cost, have a logic-level gate, and would work in this application. Their use would also avoid adding to the Bill of Materials, potentially reducing cost. Detecting attachment is a matter of monitoring the 'CC' wire for voltages above 0.8 and 2.6 volts,[82] and so can be done with simple comparators.

7.2 Cable power

The USB Type-C specification introduces the concept of 'electronically-marked cable assemblies' (EMCAs), which integrate a controller into the cable to extend its functionality. This is used to enable Type-C cables to carry HDMI and other interfaces or, most relevantly to this application, to allow a Type-C cable to state whether its current-carrying capacity exceeds the standard 3 amps.

The electronic marker in an EMCA is connected to both 'CC' wires. On the wire connecting the source and sink, it can respond to messages. On the other, now

[80] See requirements 3.3.13 and 3.3.14.

[81] See USB-PD revision 2.0 [10] at § 5.7.

[82] See requirements 3.5.1 and 3.5.2 and BS EN IEC 62680-1-3 [12] at Table 4-30.

referred to as V_{CONN}, it receives the power it needs to operate. Although this requires a source to provide a power switch to each 'CC' wire, the demands on V_{CONN} are low: up to 1 watt of power, supplied at 3.0 to 5.5 volts.[83] This gives a current demand of between 200mA and 333mA, and so a small P-channel FET acting as a high-side switch is likely to be a suitable and inexpensive option.

To reduce the risk of capacitances holding a high voltage, USB-PD requires that a discharging resistance be connected to V_{CONN} on detachment.[84] As a switch to ground, a small N-channel FET is likely to be suitable. If a complementary pair is used for both power and discharge switches, some cost saving may be possible.

[83] See requirements 3.1.9 to 3.1.11.

[84] See requirement 3.1.12.

8. System Realisation

This chapter reviews considerations which are relevant to the translation of the designs in chapters 5 to 7 into implementations. It informs component selection in Appendix A and the detailed design of the system controller in Appendix B.

8.1 Design for excellence

According to Bralla [33], the design for excellence (DfX) approach is intended to broaden the scope of what designers consider—to move from only the traditional concerns of manufacturability and cost reduction to a process where all stages of a product's life are considered. Under DfX, Bralla states, a designer should think not only of performance, safety, and manufacturability, but also reliability and durability over the longer term, impact on the natural environment, ease of service or replacement, intuitiveness for a lay-user, aesthetics and saleability, additional features that may add value or utility over its core use-case, and how quickly it can be taken from concept to marketed product.

In the context of the current project, DfX is somewhat different. The PHY does not face the user directly and would not normally be user-serviceable, reducing the influence of factors such as aesthetics and serviceability. Further, any PCB designed will be a prototype, and so must cater to test and hardware design engineers rather lay-users. The constraints on a prototype will also be looser than for a final commercial design, allowing more to be sacrificed in favour of aiding product development than would otherwise be permissible.

This section will consider the application of DfX to the system's realisation.

8.1.1 Design for manufacturing

Many general design for manufacturing (DfM) guidelines listed by Bralla apply to PCBs in much the same way they apply to mechanical components. Simplifying a design and reducing the number of unique parts will save on material costs and may attract more favourable volume discounts, as well as reducing the cost of PCB assembly[85] by reducing the number of feeders needed for pick and place machines. Designing to the manufacturing process, such as by ensuring a design can be manufactured on the common PCB layer counts of one, two, or four layers and with common materials (such as FR4 laminate) will ensure that the required capabilities are widely available at low cost. To an extent, this also includes the guideline of limiting machining: automated PCB assembly is a necessary part of the manufacturing process and cannot be eliminated, but the machining a PCB

[85] Here, 'PCB assembly' means the process of placing and soldering components on a PCB, while the more general 'assembly' means the construction of the final plug socket unit.

requires in fabrication can be reduced by avoiding through-hole components and internal slots in the PCB. Handily, this ties in with Bralla's PCB-specific guideline of maximising the use of surface-mount technology (SMT) components—not only does this reduce the number of holes which require to be drilled and plated, but SMT parts are more easily placed and soldered by machine and so reduce the machine time required. These are perhaps the most relevant or most readily applicable of Bralla's guidelines. He provides a fuller list too long to reproduce.

Industry standard guidelines also exist: IPC-2221 [34] provides a set of general requirements for PCBs, while IPC-2222 [35] is a supplement for rigid PCBs. They include recommendations such as avoiding large copper planes (which might warp during fabrication) in favour of crosshatched copper, incorporating sets of symbols for pick-and-place orientation,[86] and ensuring that land pattern pads have adequate thermal reliefs to avoid impeding soldering. These are given with other recommendations related more to general design than manufacturing.

These design requirements can be made more concrete when the capabilities of the fabricator are considered. It is common, for example, for a PCB assembly house to stock common components and offer preferential pricing when stocked components are used. In the case of PCB design, every fabricator publishes a list of design rules setting out the finest tolerances they can manufacture, adherence to which helps ensure that manufacturing defects are avoided.

8.1.2 Design for test

In general, designing a PCB for test involves somewhat obvious guidelines. As Bralla writes, it should be possible to test the PCB during fabrication and after PCB assembly. This would typically involve exposing test points—areas of copper which can easily be probed and are connected to signals likely to be needed in test. More complex PCBs with programmable logic might expose programming or communications ports, enabling use of a hardware debugger and simplifying the relaying of data to an external monitor. In the current project, the PCB being a prototype rather than a production design allows additional expense to improve testability, and so indicator lights or physical user inputs might be added.

Another key guideline Bralla identifies is to ensure that, as far as possible, the parts which are used in the design are parts in which a test engineer can have confidence. Parts which fulfil common needs and are sold in large quantity are likely to have fewer problems and, where they do have problems, these are likely to be better documented than more obscure parts. If an issue is to be diagnosed, proven parts reduce the test surface and so allow faster testing.

[86] Commonly known as 'fiducials', they are typically exposed (*i.e.* unmasked) copper shapes which a machine can recognise and use to determine how the PCB is oriented. These might appear as a set of three for the board generally with others near high-pin-count parts to enable calibration.

8.1.3 Design for safety

Safety is a paramount concern for all products with which users will regularly interact. In the case of a mains electrical device such as a USB-PD plug socket, a poorly-designed or defective product could easily expose users to the risks of electric shock and fire. Less seriously, failure to consider safety could result in financial rather than physical harm—a USB-PD source with inadequate voltage regulation might provide too high a voltage and damage a downstream device, or a failure to properly account for static discharge could damage the source itself and so require that the user purchase a replacement.

Standards such as BS EN 62368-1 [36] provide a framework for ensuring safety, with evaluations based around hazard level, user skill, and safeguards and sets of requirements for typical hazards. Particularly relevant are sections 5 and 6 on electrically-caused injury and electrically-caused fire. Other standards provide more specific requirements: BS EN 61508-1 [37] is a more specific framework for functional safety, where the safety of a system depends on electronic control elements functioning correctly; the BS EN 61000 series of over 40 standards sets requirements related to electromagnetic compatibility, including in the context of functional safety and to ensure that products are not damaged by expected or routine phenomena; and BS EN 61558-1 [38] provides specific requirements for the safety of power supplies. Many other standards exist, but these form the core frameworks and compliance with them allows a manufacturer to presume that their product conforms with the European Union's Low Voltage Directive.[87]

In the context of a prototype, the scope is more limited. A prototype need not be connected directly to mains electricity and, in this case, would not include the control for a power supply. As such, only the aspects of safety which are useful in prototyping or which are likely to impact the operation of a final design need be considered. Protection against static discharge is desirable in all contexts, for example, while protection against reverse-polarity connection might only be needed for a prototype (live and neutral being functionally interchangeable for a mains-connected final design) and protection against insulation breakdown would only be needed when mains voltages are present.

8.2 Test-driven development

The general philosophy of test-driven development (TDD) of software is that the implementation follows the tests. Given a set of requirements, the behaviour of the system can be described as a set of evaluations or assertions, and hence the implementation is produced to ensure that each assertion is true [39].

[87] Article 12 of the Low Voltage Directive [68] allows that presumption to be made for standards published in the *Official Journal of the European Union*. A list of standards [67] was published in 2018 and included EN 62368-1 and EN 61558-1.

The advantages of employing TDD are numerous. In the first instance, TDD is a practical description of requirements—no longer expressed in natural language open to interpretation, each requirement must be set out in code which must be valid and which must assess definite values and actions. This not only allows feedback into the requirements definition process by identifying possible clarity issues, it requires that the test-writer understand each requirement and the structure of the system, giving opportunities to identify design issues at early stages of development. The easy extensibility of TDD—that a new requirement can be verified with new tests isolated from all others—supports the gradual expansion of a system over its life, facilitating iterative prototyping.

This also enables a further advantage of promoting confidence in changes: once made, a change can be evaluated quickly and automatically against an existing test suite to determine whether it maintains the desired functionality. Although this relies on an extensive test suite, portions of the codebase with poor test coverage, once identified, can be readily covered with new tests. In addition, TDD encourages compartmentalisation—designing each component as a 'black box' that exposes only the necessary information to consumers. A compartmentalised black box provides abstraction, and so each test can be simplified and cover a smaller surface. A further benefit of this is that compartmentalisation tends to lead into modularity, where the implementation of a system can be readily swapped. As consumers are only aware of the inputs and outputs which are relevant to the desired behaviour, they are agnostic to implementation. A test suite written to the same interface will allow the automatic evaluation of a new module for correct functionality and operation with minimal or no adaptation.

Freeman and Price [39] identify three classes of test:

- Acceptance, testing the system at a high level with the least insight into its architecture and broadly describing system functionality;
- Integration, testing the interaction of a black box with the wider system or with external systems, *i.e.* the *integration* of the black box into the system;
- Unit, testing a single logical component of the system (the black box) in isolation from other components and verifying its core functionality.

In the context of an embedded system, the latter two classes are likely to be the most relevant. The interaction with the real world and reliance on physical phenomena common in these systems limits the extent to which acceptance testing can be done without complicated test fixtures—a practical test fixture is difficult to automate and observe, while a simulated fixture requires substantial supporting components which may require their own test suites.

This section considers the use of test-driven development in the context of a system based around a field-programmable gate array (FPGA).

8.2.1 Integration testing

In the ideal case, integration testing would be carried out by connecting the two or more functional blocks to be tested and providing sufficient stimulus to begin the interaction to be monitored. However, as Maxfield [40] describes, this is rarely a viable technique—system complexity in a non-trivial system might make simulation so slow as to impede development, while the reliance on proprietary components may mean that access to system internals is too restricted to observe whether a test completed successfully. Further, real-world components are intended to operate correctly, and so must be circumvented if failure-mode testing is to be performed. Instead, so-called 'bus functional models' (BFMs) are often used to emulate the presence of a complex component. In the broader software development terminology, these are a kind of 'mock object'.

BFMs need not be synthesisable[88] and need only provide as much functionality as is required to interface with a testbench. This allows them to be far simpler than the true components—they can be written much more like software, forgo the complex processing required in a practical system, and defer to their consuming testbench for anything more complex than acting as an adaptor—and so they can be simulated with greatly reduced computational demands.

In the current application, BFMs are likely to be most useful at the boundaries of subsystems. The components within those subsystems are likely to be simple, with much of the complexity in the glue logic holding them together.

8.2.2 Unit testing

The goal of unit testing, as the term would imply, is to ensure that a single logical unit within the system exhibits all the behaviour it should and, as far as practical, none of the behaviour it should not. If the functionality of a unit in isolation is assured, integration testing can be simplified to test only the behaviour specific to the interaction of two components.

As Freeman and Price describe, the process of devising unit tests is relatively straightforward in the general case: each unit is considered, based on applicable requirements, as a set of operations; for each operation, tests are written first to verify the most basic cases, seeding the tests for more complex cases where the reaction to particular conditions must be tested; and for each case, the most obvious implementation is produced to 'prove' the test. With confidence that the test performs as intended, an obvious implementation can be replaced with a preferred one (however evaluated) at any later point in development.

8.2.3 Formal verification

Although likely the most common form of TDD, unit, integration, and acceptance tests are not the sole form. Where these tests verify that the anticipated response

[88] That is, it need not be possible to implement them in practical FPGA hardware.

is given in selected situations, formal verification of a component ensures that it always exhibits a set of properties. More narrowly, formal verification is often considered as either equivalency checking or model checking [40, 41].

Of the two, equivalency checking is the simpler. The 'property' it verifies is that two components, or two representations of a single component, behave in the same manner for the same inputs. Seligman *et al.* [41] give the example of a truth table and a hand-assembled arrangement of logic gates: the manual translation process introduces a risk of human error, but equivalency checking verifies that the truth table and logic circuit are equivalent. A similar application may be in normal development flow, where a high-level description produced by a human and a lower-level translation into another form by a tool are verified equivalent, or where a component of a system is replaced and the ability to verify that the replacement has introduced no regressions is desired.

Model checking, on the other hand, is far more complex. The desired behaviour of the component is described as a set of assertions—for example, from Maxfield, that two signals should never be active at the same time—framed by constraints which establish the 'world' the assertion exists in. Commonly, a constraint may be used to describe the behaviour of a clock signal, both ensuring that formal verification can proceed with the component driven correctly and to avoid the need to filter errors which would never occur in a practical situation if the component is used correctly. Once specified, these assertions and constraints are provided to a 'SAT' (satisfiability) solver—a utility which determines, from the input space, whether an input exists which causes an assertion to fail [41]. As the result of applying formal verification is full coverage,[89] it is plainly desirable.

Formal verification is unlikely to be useable in the current project. Tools to verify designs are often restricted by expensive commercial licences, with very few free alternatives. The free tools which are available are sparsely documented, highly unstable, and may depend on commercial components. The 'SymbiYosys' [42] tool, for example, aggregates synthesis, SAT-solver, and related tools but relies on a non-free commercially-licensed parser to support SystemVerilog Assertions, a language used to write assertions. It does not support VHDL assertions, but the free GHDL simulator with the beta[90] synthesis tool 'ghdlsynth' [43] does and can produce output which SymbiYosys can use. However, this software is in the early stages of development and so is not currently useful in a final design.

[89] That is, that all portions of the design (or, in the case of *code* coverage, all lines of source code) are evaluated by a test.

[90] The pre-release lifecycle of software is often separated into pre-alpha, alpha, and beta stages, each representing a progressively more-developed product. Software may then progress to release-candidate stage before its first stable release. Before stable release, software is generally expected to contain bugs and incomplete portions.

9. Review

This chapter provides an evaluation of the project and the extent to which its aim, objectives, and requirements have been met.

9.1 Project schedule

The current project was organised around a 28-week period of work with a set of high-level tasks arranged into three general phases:

- **Planning**, covering the development of the project proposal and requirements as well as the essential research (weeks 1–10);
- **Implementation**, where research was to be transformed into practical designs and those designs tested (weeks 9–23);
- **Evaluation**, where the project was to be reviewed and collated into its final form before being edited and sent for print (weeks 21–26).

The remaining two weeks were allocated for slippage.

In the first half of the period of work, progress was as planned. As the result of unfruitful research, a slight delay of half a week accrued whilst investigating the analogue portions of the design (week 7). This did not compound and so had only a minor impact on the project's schedule.

At the transition between the planning and implementation phases, progress on the project suffered a substantial setback. In the project's proposal, it had been planned that designs for printed circuit boards would be sent to a fabricator in weeks 12 to 17. This period encompassed the Christmas holiday, and so the impact of lead time would have been reduced as the lead period would align with a time of reduced work. Despite the approval of that proposal, which included a broad estimate of expenditure, a request for funding was denied. This prevented the use of a fabricator. It is unlikely that the precise impact on progress can be determined, but it is thought that this caused the loss of at least six weeks.

The request for funding was made in week 12, circuit components were received in week 20, and circuit board assembly was completed in week 24. This would initially suggest a seven-week delay. However, slowed progress meant that the earliest a circuit board design could have been started was week 13. Based on the time required for the two PCBs produced, a full design could have been created in two weeks, moving the time of order to the end of week 14. This would have shifted the lead period so that it ended in week 19, and so completion in week 24 would be a delay of five weeks. In actuality, a funding decision was not obtained

until week 14, and uncertainty prevented the creation of a design—the fabricator provided capabilities so substantially different from those available without funding that a design produced for one eventuality could not be usefully adapted for the other. Delay in obtaining a decision pushed circuit board design work into the university examinations period and the Christmas period, both increasing any delay already present and consuming time which would have been spent on implementing the system controller. This prevented the implementation of the system controller from beginning before week 18, accounting for the increase in the delay to six weeks or more.

The result of this is that the planned schedule ceased to be meaningful in the second half of the period of work. The loss of more than a fifth of the total time available removed any prospect of successful implementation and testing.

The implementation of the system controller continued until week 23. At that point, approximately half of the intended functionality was implemented; the biphase mark code (BMC) transceiver was simulated and tested, leaving the less complex binary frequency-shift keying (BFSK) transceiver and supporting functionality required to operate the transceivers. Considering the five weeks taken to gain familiarity with the tools and implement the BMC portion, it is not unreasonable to assume that the simpler BFSK portion and related functions could have been implemented had the six weeks been available. In that scenario, the prototype hardware would have been available from week 17 and so testing of both portions could have been carried out concurrently.

Those factors in mind, although the project substantially deviated from its original schedule, it is reasonable to assume that—had funding not been unpredictably denied—the project would have continued to progress as planned, with the delay in week 13 made insubstantial by the allocated slippage and any opportunities to regain progress which might have arisen.

9.2 Designs

While a full test of the project designs was not possible, reflective commentary on some portions of the designs remains possible. This section provides that commentary and considers the outcome of the tests that could be carried out.

9.2.1 BMC transceiver circuit

It was initially intended that the BMC transceiver circuit in figure A1 would include a USB Type-C receptacle, allowing connection of a cable and testing with practical USB-PD devices. A failure in component selection prevented this. The receptacle selected, the Amphenol 10137062-0021LF [44], is mounted on pins that are 0.9mm long. A typical circuit board thickness, and the thickness of the transceiver board, is 1.6mm. Thus, soldering in a conventional manner was not possible and the receptacle was not mounted.

9.2.2 BFSK transceiver circuit

Despite the BFSK transceiver shown in figure A2 not being tested, it is possible to identify a likely issue from a review of its schematics in chapter A1.

The PWM DAC filters a high-frequency input square wave to obtain a DC output proportional to the duty cycle. As the schematic shows, this is done with three cascaded RC filters. Each is buffered by an op-amp with a 1MHz gain–bandwidth product (GBW). Commonly, the input to a filter is also buffered to avoid external impedances impacting filter operation, and the PWM DAC is no exception. The issue here lies in the frequency of the input wave—A3.5.2 suggests a comparison of 2MHz and 20MHz, both of which lie above the buffer's GBW.

Had a direct connection to the filter been made, this would have presented no issue: the RC filters have bandwidths well below the input frequencies, and so the buffers would be presented with low-bandwidth near-DC input. However, as the input buffer has no RC filter, this does not occur. The precise result depends on the internals of the op-amp, but a level of distortion and the potential for reduced gain are likely. These may prevent full output swing, limiting varactor control.

In the test circuit, this is readily corrected by shorting out the input buffer—the ability to control the test environment limits the impact of external impedance. A final design might use a discrete transistor buffer, avoiding the cost of a higher-performance op-amp.

9.2.3 System controller: BMC transmitter

The software for the system controller was a project area where substantial progress was made within the time available, and the BMC transmitter—which operates the hardware line driver—was one of the components completed and tested in simulation. This enables fuller review than for other project areas.

The high-level design of the BMC transmitter is shown in figure B1, and the functional descriptions of the individual components in chapter B2. Overall, the transmitter's implementation was successful, but use in the test environment revealed areas where design could be improved. As designed, the transmitter is somewhat spartan; its interface provides raw access to a transmit buffer where the user is responsible for assembling packets and generating the check code, if present. This could not be directly attached to the protocol layer as figure B1 shows—a separate translation layer would be required, or the protocol layer would require abstraction-breaking knowledge about lower-level components.

A better interface—that is, one better suited to the protocol layer–would likely present USB-PD's message types as part of its interface and adopt a transactional approach. This could be designed primarily around the most common use-case and provide special interfaces for 'Hard_Reset' and similar messages. As an example, this interface might require a user to first write to a 'destination select'

register and then, if appropriate, a ‘data sink’ register. Acting statefully, bus error signalling can indicate when an action is invalid. The transmitter, once its supply of data was exhausted, could then automatically append a check code and ‘EOP’ packet-terminating K-code. This would both simplify the user’s task and remove the potential for a user to send a malformed packet.

In relation to bus errors, that system could also be improved. The transmitter exposes an *ERRNO* register which holds a code corresponding to the last error that occurred. This necessitates a further bus transaction whenever an error is encountered. The Wishbone specification, however, permits a Wishbone slave to set its data output when an error is asserted,[91] and so an error code could instead be reported on the Wishbone data bus. This would simplify design.

9.2.4 System controller: BMC receiver

As with the BMC transmitter, the BMC receiver portion of the system controller software was implemented and tested in simulation within the time available. Its high-level design is also shown in figure B1.

Similarly to the transmitter, the receiver would be improved by a change in its interface. As designed, it provides no means of identifying when a transmission has been received other than by checking that the receive buffer is not empty. It also directly exposes the K-codes included in the packet. Instead, a receiver which interpreted the contents of packets and filtered out those not addressed to the attached protocol layer would be preferable. In addition, some means of identifying when multiple transmissions have been received—and separating the two when reading from the receive buffer—would improve usability.

The receiver’s means of error reporting also require improvement. As specified, an error related to a transmission persists until that transmission ends. This is fundamentally unfit for purpose. If an error occurs at the end of a transmission, it may be cleared before a user has time to notice it. Alternatively, if the user is processing data in chunks, an error that occurs during processing may clear the receive queue before the user has time to notice, potentially leaving the user in an invalid state with only partial data. As with the transmitter, some of these design issues could likely be resolved with a fuller-featured abstraction—if the receive queue was instead managed rather than being a ‘dumb’ queue, received messages could be requested individually and only the relevant data read out. A set of separate error signals would be required to indicate errors which occur asynchronously with Wishbone bus transactions (such as the receive queue filling), allowing a partially interrupt-based system.

More technically, the receiver would be better implemented if—like the transmitter—it operated from two separate clocks. As implemented, the clock

[91] See the Wishbone specification [69] at rule 3.65.

used for the Wishbone interface is the same clock used to sample the line, but a high sampling frequency of 100MHz could present timing issues when used for the bus. Internally, the sampling portion and the bus-attached portion are separated by a FIFO, and so synchronisation requires only that the synchronous FIFO be replaced with an asynchronous one.

9.2.5 System controller: FPGA resource utilisation

It was intended, as A3.1 discusses, that the system controller would be realised on a field-programmable gate array (FPGA). The flexibility of FPGAs combined with their ability to perform high-speed highly parallel processing makes them well-suited to operating multiple transceivers while carrying out unrelated control tasks. Unlike a microcontroller, however, it is generally not possible to enable an FPGA to carry out more varied tasks by connecting more memory containing additional software—once the hardware resources of an FPGA are exhausted, the two solutions available are replacement with a larger FPGA or delegation (over an external bus) to an accelerator or coprocessor.[92]

It is clear, then, that evaluation of the system controller must consider whether a complete system could be deployed on the target FPGA. Table 3 provides close estimates of the resources used by the components of the system controller.

Although a Lattice ICE40UL1K is the intended final target, a Xilinx Artix-7 XC7A35T was used to prototype. This necessitates translation from resources available on the XC7A35T to those available on the ICE40UL1K. The XC7A35T's lookup tables (LUTs), for example, include dual outputs that allow some pairs of LUTS to be combined [45]. Where this occurred, it was assumed that a LUT4 was eliminated. The ICE40UL1K includes only LUT4s [46], two of which can represent a logic function with seven inputs and one output. Each LUT6 can therefore be replaced by two LUT4s, with a further added for each elimination to provide the second output. This will inflate the estimate in certain edge cases, such as where a pair of two- or three-input functions are combined into one LUT6. These could be represented by two individual LUT4s, but this approximation will estimate three. On the other hand, where two LUT6s are combined, this method will give three LUT4s where four would be needed, reducing the estimate. The number of eliminations is shown in table 3.

[92] This is not strictly true. Many vendors support so-called 'partial reconfiguration', where some of the logic elements within the FPGA are reprogrammed while others operate normally. This can provide substantial performance gains, but requires sizeable external memories and, owing to the long programming times, advance knowledge of when reconfiguration is required [70]. No sources indicating that the target FPGA supports partial reconfiguration could be identified.

Table 3. System controller FPGA resource utilisation.

Component	LUT	LUT6	LUT4	FF	Elims.
BMC receiver	**315**	**267**	**48**	**162**	**57**
Control unit	178	142	30	108	27
CRC engine	25	10	15	32	7
End-of-packet detector	78	78	0	6	15
Binary search unit	27	27	0	11	4
4b5b decoder	13	10	3	5	4
BMC transmitter	**53**	**37**	**16**	**50**	**11**
Control unit	31	21	10	32	6
Line-driving block	8	6	2	7	3
Preamble generator	9	6	3	6	2
4b5b encoder	5	4	1	5	0

Estimating as described, a total utilisation of 740 LUT4s is likely when targeting the ICE40UL1K. That device provides 1248 logic cells, each with a LUT4 and a flip-flop, and so around 508 LUT4s would remain for implementing the BFSK transceiver, the protocol layer, and other functional units.

The true resource cost of the BFSK transceiver is difficult to predict, but a broad estimate is possible. It can reasonably be assumed that the BFSK receiver could share the CRC engine, end-of-packet detector, and 4b5b decoder. The transmitter would be identical to the BMC transmitter, except with the line-driving block replaced by a PWM generator. If the PWM generator uses four LUT4s[93] and the BFSK receiver control unit is equally as complex as BMC receiver control unit, the additional cost of the BFSK transceiver is around 345 LUT4s and so the total resource cost rises to 1085 LUT4s.

This leaves 163 LUT4s for the protocol layer and miscellaneous physical layer signalling. Although the transmitting portion of the protocol layer is likely to be simple—its work is largely copying from memory to a transmit buffer—the receiving portion is likely to be expensive. As table 3 shows for the end-of-packet detector, state machines to interpret packets can come at considerable cost. It is unlikely that this could be implemented in 163 LUT4s, and even a solution such as a soft processor core—which enables reuse of basic functional units—is likely to demand more resources than are available. That said, the system controller was developed with the intention of producing a working prototype and with only limited regard to resource efficiency. The timescales permitted by USB-PD are

[93] This is double the resource cost of the frequency divider in the BMC receiver, which is similar in implementation to a PWM generator. In table 3, the divider is included with the control unit.

long,[94] and so refinement to a resource-minimising design which exploits the ample time available could likely make significant savings. Further, the BFSK receiver control unit will be simpler than that for the BMC receiver, and so it is likely to consume significantly fewer than the 345 LUT4s predicted.

Assuming a 15% saving, the resource budget would grow to 325 LUT4s—more than enough to host a general-purpose 8-bit processor.[95] Using an application-specific instruction set instead, further savings could be achieved by eliminating unneeded functional units or capabilities. As such, while resource availability will present a substantial challenge to viability, it is not unreasonable to assume that full implementation in 1248 LUT4s would be possible with optimisation.

As a final note, the supply of input–output pins (I/Os) on the FPGA may impact viability. The schematics in chapter A1 show that the prototype designs require 22 I/Os. The ICE40UL1K includes 26 I/Os on its QFN36 package and so, once any other I/Os required to operate or interact with the power converter are included, a final design may use all that are available. That said, I/O usage in a final design could be optimised. For example, pairs of transmission gates are used (each gate having its own control signal), but their replacement with analogue multiplexers could eliminate one of the two control signals. Alternatively, a port expander like the PCA9570 [47] (15.6p at 8ku) could be used for the Rp resistances and other low-speed switching tasks.

9.3 Objectives

9.3.1 Implementing the USB Power Delivery physical layer

Objective 1.3.1 was to produce a design implementing USB-PD's physical layer with a BMC transceiver. The BMC scheme is set to be used by USB-PD for the foreseeable future, and so it is inseparable from the wider need to implement the physical layer. The result is a broad objective which covers both hardware and software portions of the project design.

As is to be expected, the changes to the schedule discussed in 9.1 severely impacted compliance with this objective. Prototype hardware was received too late to test to any meaningful extent and reduced development time prevented full software implementation of the physical layer, meaning compliance with many of the requirements in chapter 3 must be determined or assumed from a theoretical design rather than practical verification. In mitigation, an automated test suite was developed for the software and portions of the hardware design were refined in a circuit simulator, allowing a small number of requirements to

[94] Most standard-specified times are tens or hundreds of milliseconds: [9] at § 6.6.21.

[95] Lattice Semiconductor, the manufacturer of the ICE40UL1K, quote as little as 193 LUT4s for their Mico8 design at up to 48MHz on a comparable complex programmable logic device (CPLD) [71].

be tested against in simulation. This does not guarantee correct functioning in a practical device but can demonstrate that the structure is fundamentally correct.

In summary, of the 48 requirements in 3.1, 3.2, 3.3, and 3.5, compliance with 13 has been demonstrated by simulation, three are guaranteed by design, and the remaining 32 are untested.

Those tested against in simulation primarily relate to the software portion of the design. Requirements 3.3.4, 3.3.10, and 3.3.15 to 3.3.20 are representative examples, although hardware requirements such as 3.3.23 are also included. In the case of those guaranteed by design, each is a simple requirement: 3.1.1, 3.3.3, and 3.3.24 all relate to well-specified component values. Although similar to these, requirements such as 3.1.8, 3.3.25, and 3.5.2 cannot be evaluated without practical testing because of the presence of controlled or active devices.

As assessing compliance with most requirements was not possible, this objective has not been met.

9.3.2 Supporting legacy communication modes

Although BMC is the primary transmission format for USB-PD, support for the legacy BFSK scheme from USB-PD revisions 1.0 and 2.0 is likely to be valuable in ensuring wide compatibility. Objective 1.3.2 was to investigate the viability of including support for this format in the project design.

Much like for the BMC transceiver and USB-PD more widely, changes to the schedule prevented full satisfaction of this objective. Requirements specific to the BFSK transceiver are set out in 3.4 and, of the 19 specified, none can be considered met. In part, this is the result of choices made to simplify the test circuit—no connection to a power converter or practical device is present and so, for example, requirements such as 3.4.1 and 3.4.10 have no application. Others largely relate to the processing logic which, without a software implementation and with only part of the hardware systems simulated, has been tested to too little an extent to make assumptions about it reasonable.

At a high level, some assessment is possible. The Colpitts oscillator was tested in simulation successfully with various fixed capacitances. The software, although not produced due to changes in the schedule, would share a substantial part of its functionality with the BMC portion of the software, as figure B1 shows. If the varactor tuning scheme works as intended, software-side consolidation in the transmitter could mean only a slight increase in the burden on the system controller. This is somewhat negated by the expected complexity of the receiver portion but, as 9.2.5 concludes, it is likely to remain viable. For the hardware required, its cost is evaluated in 9.3.3 and its size in 9.3.4. They conclude that the hardware is likely to be cost-effective and that area pressure is likely to be minimal. As there is no question that hardware for a system of this kind can be

implemented, there are no challenges to viability from the hardware portion.

Considering that assessment and noting that implementation is not expressly a target outcome, it is reasonable to consider this objective partially satisfied.

9.3.3 Offering a cost-competitive solution

The cost of the project designs is a key factor in the success of the project. The use of a semi-discrete construction has clear downsides, most significantly its added complexity, which must be balanced by it having substantial advantages over a monolithic construction. As 2.1 discusses, a practical advantage is that the semi-discrete construction can provide wider compatibility with a reduced level of redundancy. However, if a solution with wider compatibility is not competitive on cost, the value of the compatibility advantage provided is greatly reduced. If objective 1.3.3 is to be satisfied, the produced design must be cost-competitive.

Table 4 estimates the cost of each solution using 5000-unit prices or the nearest price breaks available. An item marked with • includes general estimates.

Table 4. Cost comparison of semi-discrete and monolithic solutions.

Semi-discrete solution		**Monolithic solution**	
System controller	£1.067	TI TPS65988	£2.810
V_{CONN} power switches	£0.170	SPI Boot Flash, ≥64kB	£0.189
Type-C ‘CC’ multiplexing	£0.335		
BMC signal extraction	£0.412		
Attachment detection •	£0.100		
Tuneable oscillator •	£0.400		
Rectifier receiver •	£0.300		
SPI EEPROM, ≥16kB[96]	£0.202		
V_{BUS} power switches	£0.530		
Total	£3.516	**Total**	£2.999

In estimating cost, the prototype designs in Appendix A provide much of the information on components. Although not implemented, sub-circuits such as the automatic gain controller (6.3.2 and A3.6) must still be considered, and so the remaining portions of the cost can be general estimates made using knowledge of the kinds of component likely to be required.

The cost estimates also make several assumptions about the design. First, it is assumed that the design as prototyped functions correctly. Second, it is assumed

[96] Assuming, as discussed in 9.2.5, that a soft processor core is implemented.

that the costs for power supply systems,[97] circuit boards, and small passives are either very similar or unlikely to differ significantly. Third and finally, the choice of controller for the monolithic design is assumed to be the TPS65988 [8]. It is the most expensive of those discussed in 2.1, but is also extensively documented and provides 5-amp-rated power switches, current limiting, and USB-BC support.

As estimated, the cost of the semi-discrete solution is 51.7p (17%) above that of the monolithic solution. If resource exhaustion (see 9.2.5) necessitates a larger FPGA for the system controller, the Lattice ICE5LP2K (£2.72 at 2ku) would make 2048 LUT4s and 39 I/Os available but would increase cost to £2.17 (72%) above the monolithic solution.[98] Assuming that the ICE40UL1K is suitable, the 17% increase in cost to support an entirely new transmission format would appear to make the semi-discrete solution more favourable. The cost of the ICE40UL1K is that quoted by a distributor at orders above 490 units, and so in high-volume production the difference may be lower still. Market analysis would be necessary to understand the number of USB-PD revision 1.0 and 2.0 devices in circulation and hence assign a more precise value to the inclusion of the BFSK portion of the PHY, but that is beyond the scope of the current project.

If it was shown that market conditions did not justify inclusion of the BFSK portion, it would likely be possible to adapt the BFSK portion to implement a system like that envisaged by Reydams and Lauwereys *et al.* [1]. In their paper, they describe using amplitude shift keying (ASK) to transmit Ethernet packets over house wiring, enabling telemetry monitoring and communication between plug sockets. If USB-PD packets were transmitted instead, system controller components could be reused to minimise additional resource cost. To avoid the potential for data loss or corruption, it would be necessary to ensure that BMC and BFSK receivers could operate simultaneously. This could likely be done by implementing pre-decode buffering.

Given the relatively minor cost increase, it is reasonable for the semi-discrete solution to be assessed as cost-competitive and so this objective is met.

9.3.4 Supporting installation in place of a plug socket

Objective 1.3.4 is for the project design to be amenable to integration into a plug socket-sized enclosure. As noted in 2.1, BS 4662 [3] specifies sizes of enclosures for plug sockets and so gives target sizes for the project design. A flush-to-wall one-gang box has a cavity with dimensions 68.3×68.3×47mm and is likely to be

[97] Note that, while the prototype design uses five voltages (+12, +5, +3.3, −5, and −3.3), adaptation for operation from two voltages (+5 and −5) is possible without major design change.

[98] Note that the Lattice MachXO3 CPLDs are available at lower cost (in the £2.25 range), but that these devices are intended for use as glue logic rather than as FPGAs. It is unclear what impact this difference in target application would have on the device's suitability. Devices from other manufacturers are also available but at significantly increased cost—the Intel 10M02M153 provides 2000 LUT4s and 112 I/Os, but costs £3.56 at 500u.

the most suitable target—it occupies the least space and so has minimal aesthetic impact and, along with two-gang boxes, should be widely available.

Figure 20 provides an annotated view of the PCB for the BFSK transceiver, with the areas required by significant sub-circuits indicated.

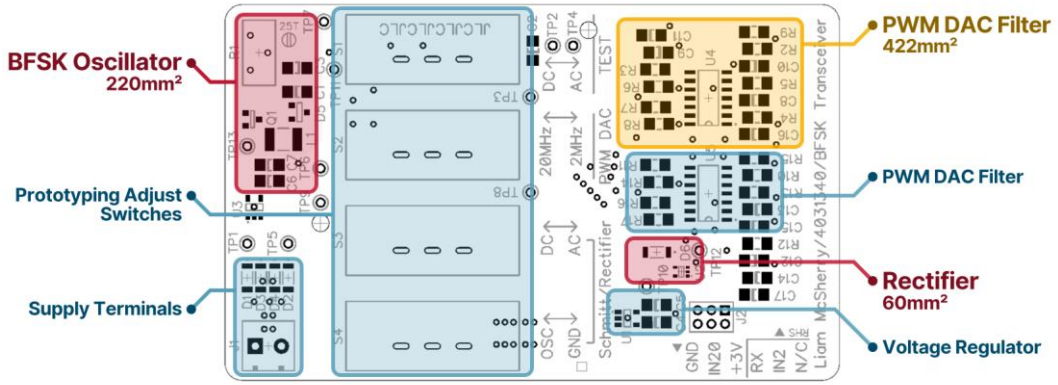


Figure 20. Annotated diagram of the BFSK printed circuit board.

A similar diagram is provided in figure 21 for the PCB for the BMC transceiver.

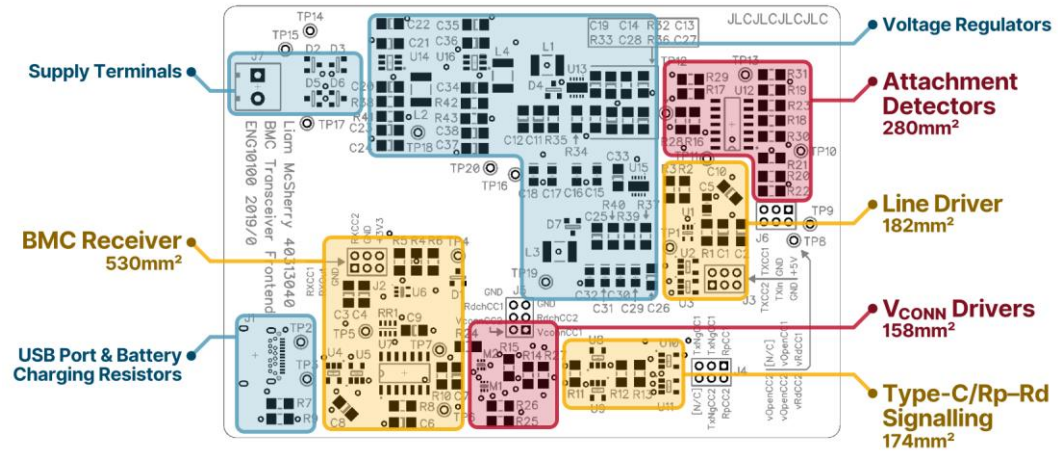


Figure 21. Annotated diagram of the BMC printed circuit board.

In evaluating this objective, it is immediately possible to discount the dimension of height—the PHY, as prototyped, is almost entirely surface-mount components, and so its height is unlikely to exceed single-digit millimetres. Although a power converter is likely to occupy significant space, its design is not considered as part of the current project. A standard one-gang box has mounting lugs recessed at least 4mm from its outer face, and so the minimal height of the PCB allows the

entire 68.3×68.3mm cavity to be exploited.[99] Added to this, additional space will be available behind a plug socket faceplate if required. This in mind, a broad estimate can be made based on the area required by the prototype circuits.

The sum of the areas highlighted in figure 20 and figure 21 is 20.26cm². This compares well with the 46.65cm² available, being 43% of that area. Although this does not include the power circuitry,[100] the system controller itself, and other sub-circuits, the remaining space being 57% of the total is a positive indicator. It is especially positive when the opportunities for size reduction in a final design are considered—the prototypes use relatively large '1206'-size[101] components to ease hand assembly but, if automated assembly were used, components of a much smaller size could be used instead. Further, the prototype circuit boards are two-layer, but four-layer capabilities are widely available and could likely improve the density of routing significantly. Added to this, double-sided assembly[102] is not uncommon, and could reduce area pressure if used. These considerations show that space is unlikely to be a challenge to project viability and so that this objective has been met.

Briefly considering the power converter, the size of typical power transformers and the insulation requirements for mains voltages[103] could make multiple stacked PCBs a desirable solution, which would further alleviate area pressure.

9.3.5 Complying with standards and law

The purpose of objective 1.3.5 is to ensure that any project design meets the requirements that are likely to apply to it, easing commercial adaptation. It covers both legal requirements, which might include safety and electromagnetic compatibility, and standard requirements. A clear business need exists for each: a product cannot be sold if it does not meet legal requirements, and a product will have no market if it cannot meet the standards.

This objective's extent is limited by its context. Legal requirements as to safety have limited impact—the prototypes do not deal with power conversion, and their working voltages are well below the level likely to present an electric shock risk.

[99] If greater depth were required, it would be necessary to ensure the PCBs would fit between the mounting lugs. This limits a square PCB to 52×52mm.

[100] While voltage regulators are highlighted in those figures, the difference in the prototypes and any final design is likely to be great enough that the prototype circuit will not accurately predict area.

[101] Size codes are length and breadth dimensions, typically in hundredths of an inch unless metric sizes are expressly mentioned. For example, '1206' is 0.12×0.06″ [72].

[102] That is, assembly with components on both sides of a circuit board.

[103] See BS EN 62477-1 [73] at § 4.4.7, which specifies 2.5kV impulse withstand and hence conductor separation of at least 0.56mm and case clearance of at least 1.5mm for a device which is part of the fixed installation in a house (overvoltage category III) in minimally-polluted area (pollution degree 1), where the peak DC voltage is 357.8V (that being the peak of rectified 230V +10% AC) and the circuit is assembled on an FR4-based PCB (group IIIa insulating material).

Although electromagnetic compatibility remains relevant, it is difficult to assess compliance outside of specialised laboratories. No access to such facilities was available. Interference mitigation can be applied broadly, such as through the limiting of slew rate in USB-PD, or by the inclusion of decoupling capacitors to reduce noise infiltration. Section 8.1.3 provides further consideration. As it notes, BS EN 61000 includes over 40 standards, and so assessing full compliance is far beyond the scope of the project.

As regards standards requirements, the project was designed to an extensive requirements specification (see chapter 3) which transposes those most relevant to design, and designs for PCBs were produced taking the guidelines mentioned in 8.1.1 into account. This is visible in the routing, where trace length is minimised and traces through copper planes are arranged so that a clear and direct path for plane-to-plane connections is maintained, and through the ample inclusion of test points and use of recommended land patterns. Although not all recommendations are implemented, derogation has been with justification—for example, although large solid copper planes are not recommended for wave or reflow soldering, it was known that the prototypes would be hand-soldered and so that planes could be made solid to offer easier routing to components within the plane boundaries.

It is reasonable, then, to consider that the objective has been met as far as it relates to identifying requirements. The compliance portion of the objective, however, cannot be considered met—as 9.3.1 and 9.3.2 discuss, the changes to the schedule described in 9.1 prevented testing and so prevented assessment of whether most requirements had been met. Overall, as compliance cannot be assessed, this objective cannot be considered met.

9.4 Professional competence

The UK Standard for Professional Engineering Competence (UK-SPEC) [48] is the standard against which prospective Engineering Technicians, Incorporated Engineers, and Chartered Engineers are evaluated. As the style of Chartered Engineer (CEng) is the terminal qualification for engineers, evaluation against the CEng criteria is one useful method of determining the value of a piece of work in demonstrating engineering skill and professional competence.

Table 5 briefly considers each part of the CEng standard and identifies relevant work in the current project demonstrating satisfaction of the standard.

Table 5. Review of project against the CEng standard.

CEng standard criterion		Relevant work
Engineering knowledge and understanding		
A1	Maintain and extend a sound theoretical approach to technology	Research into USB-PD and other standards.
A2	Engage in creative and innovative development of technology	In part, cross-disciplinary use of analogue and digital design.
Application of engineering methods		
B1	Identify projects and opportunities	The context in 1.1 taken with the background description.
B2	Conduct research, design, and development of solutions	Chapters 5 and 6, Appendix B.
B3	Manage and evaluate implementations	Chapter 9, limited by schedule.
Technical and commercial leadership		
C1	Plan for effective implementation	Scheduling and continuous review with supervisor.
C2	Plan, budget, organise, direct, and control tasks, people, and resources	General project control, use of technician services.
C3	Lead teams and develop staff	*Not applicable.*
C4	Bring about continuous improvement through quality management	*Not applicable.*
Interpersonal skills		
D1	Communicate with others at all levels	Continuous review meetings, the interim report, logbook.
D2	Present and discuss proposals	Detailed project proposal.
D3	Demonstrate personal and social skills	Continuous review meetings, interaction with technicians.
Professional standards and obligations		
E1	Comply with relevant codes of conduct	Interaction with technicians.
E2	Manage, apply safe systems of work	*Not applicable due to schedule.*
E3	Further sustainable development	*Not demonstrated.*
E4	Undertake continuous professional development	*Not applicable.*
E5	Exercise responsibilities ethically	*Not demonstrated.*

As shown, the project work aligns well with the CEng standard. The standard's focus is engineering in a professional context, either in commercial projects or in research, and so certain points have limited applicability. Similarly, although the current project aligns well at the surface level, it cannot demonstrate the same professional depth as a commercial or novel research project. When taken with future work, however, it is likely to provide a sturdy foundation.

The current project failed to meet two criteria: E3, which is to 'undertake engineering activities in a way that contributes to sustainable development', and E5, which is to 'exercise responsibilities in an ethical manner'.

For criterion E3, UK-SPEC gives the examples of carrying out impact and risk assessments, implementing best practice, and adopting appropriate practice. In principle, it can be seen that widespread adoption of USB-PD plug sockets could have a positive environmental impact—with a reduced need for device-specific power supplies, waste from power supplies for obsoleted devices can be reduced and the working life of any USB-PD supplies manufactured can be extended. The project scope does not include consideration of sustainable development, and so the project does not formally evaluate the potential impact. It would therefore be difficult to justify an evaluation that criterion E3 had been met.

To satisfy criterion E5, application of the Statement of Ethical Principles adopted by the Engineering Council and Royal Academy of Engineering [49] must have been demonstrated. Its four principles are:

- Honesty and integrity;
- Respect for life, law, the environment, and public good;
- Accuracy and rigour;
- Leadership and communication.

In large part, the failure to satisfy this criterion can be attributed to a lack of opportunity. Although conducted ethically, the current project offers few specific and demonstrative scenarios. It could be argued that respect for the environment is demonstrated by the waste-reducing nature of the project design discussed above, or that accuracy and rigour are demonstrated in the project's research and the critical evaluation in this chapter, but these are somewhat insubstantial justifications and fail to cover all four of the principles. As such, the criterion cannot be considered satisfied.

9.5 Conclusion

The aim of the current project was to design and build a physical layer for the USB Power Delivery standard. The physical layer was to use semi-discrete

components and was to be suitable for integration into a power supply in a conventional plug socket form factor. Five guiding objectives were set for the project, of which two were met and one was partially satisfied. Derived from these objectives, 67 requirements were identified for the project designs and, of those, 16 were met. As such, the aim of the project has not been fulfilled.

The project work broadly divides into the BMC transceiver (chapter 5), the BFSK transceiver (chapter 6), and the system controller logic (Appendix B). As discussed in 9.1, 9.3.1, and 9.3.2, changes in the project schedule prevented the full realisation of the designed components and so, while a physical layer has been designed and built, it is not known whether the design is suitable. However, partial tests in simulation have shown that the foundations of the circuits are sound and that the system controller, as far as implemented, functions as intended. This, taken with the informed estimations in 9.3.3 and 9.3.4, suggest that the project designs are likely to be viable from cost and size perspectives: at 17% higher parts cost than a BMC-only PHY, the semi-discrete solution should be able to provide both BMC and BFSK support; and, of the area available in a one-gang box for a plug socket, the non-ancillary prototype circuitry required less than half before optimisation for space efficiency. These results, although heavily based on estimation, allow the conclusion that a semi-discrete solution is both viable and economical, provided that market justification for the inclusion of support for the BFSK scheme exists.

Further, although the project failed to fulfil its aim, 9.4 shows that work carried out for the project is aligned well with what is expected of professional engineers and so that the project provided valuable experience.

Appendix A

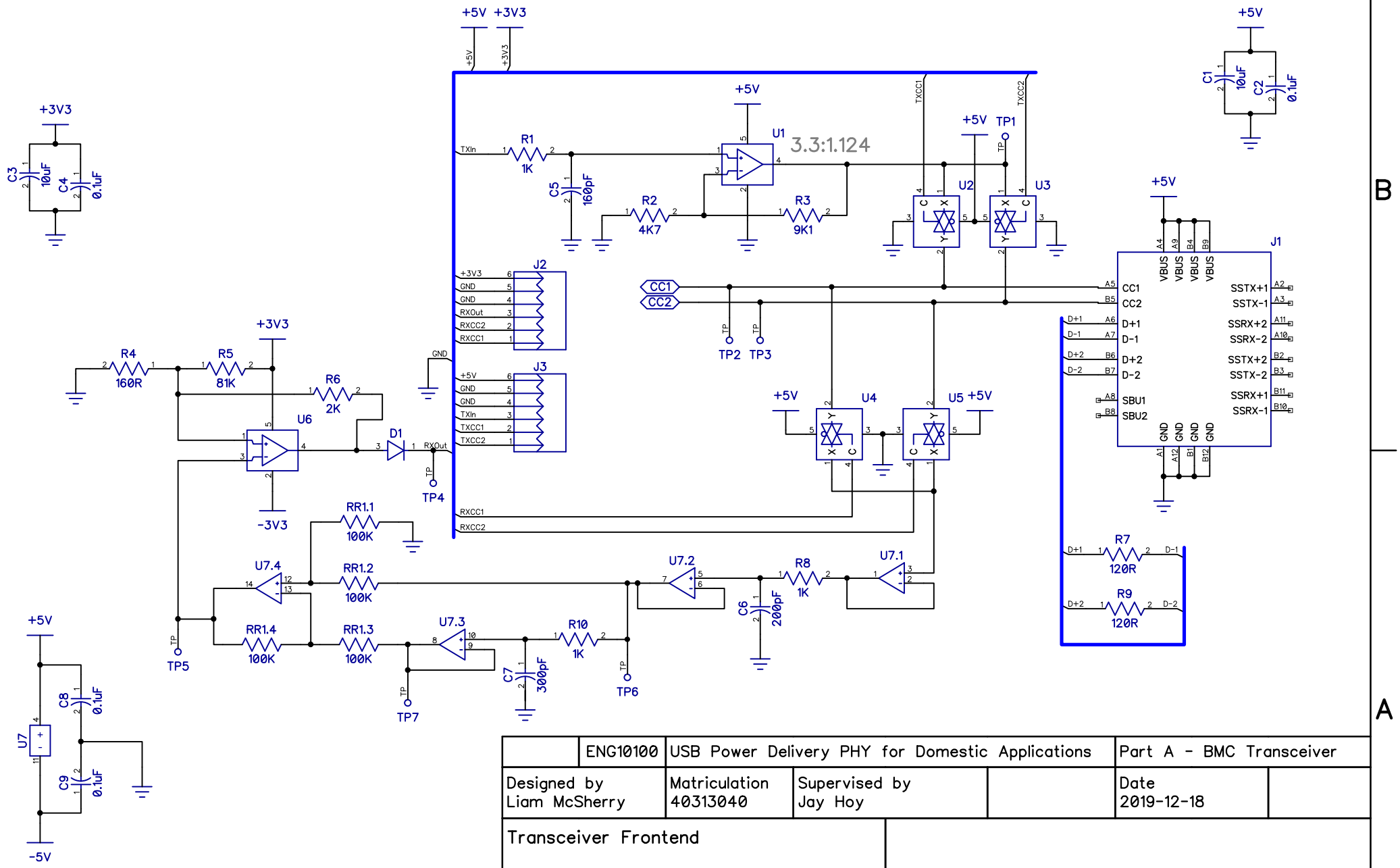
Circuit Realisation

A1. Schematic Diagrams

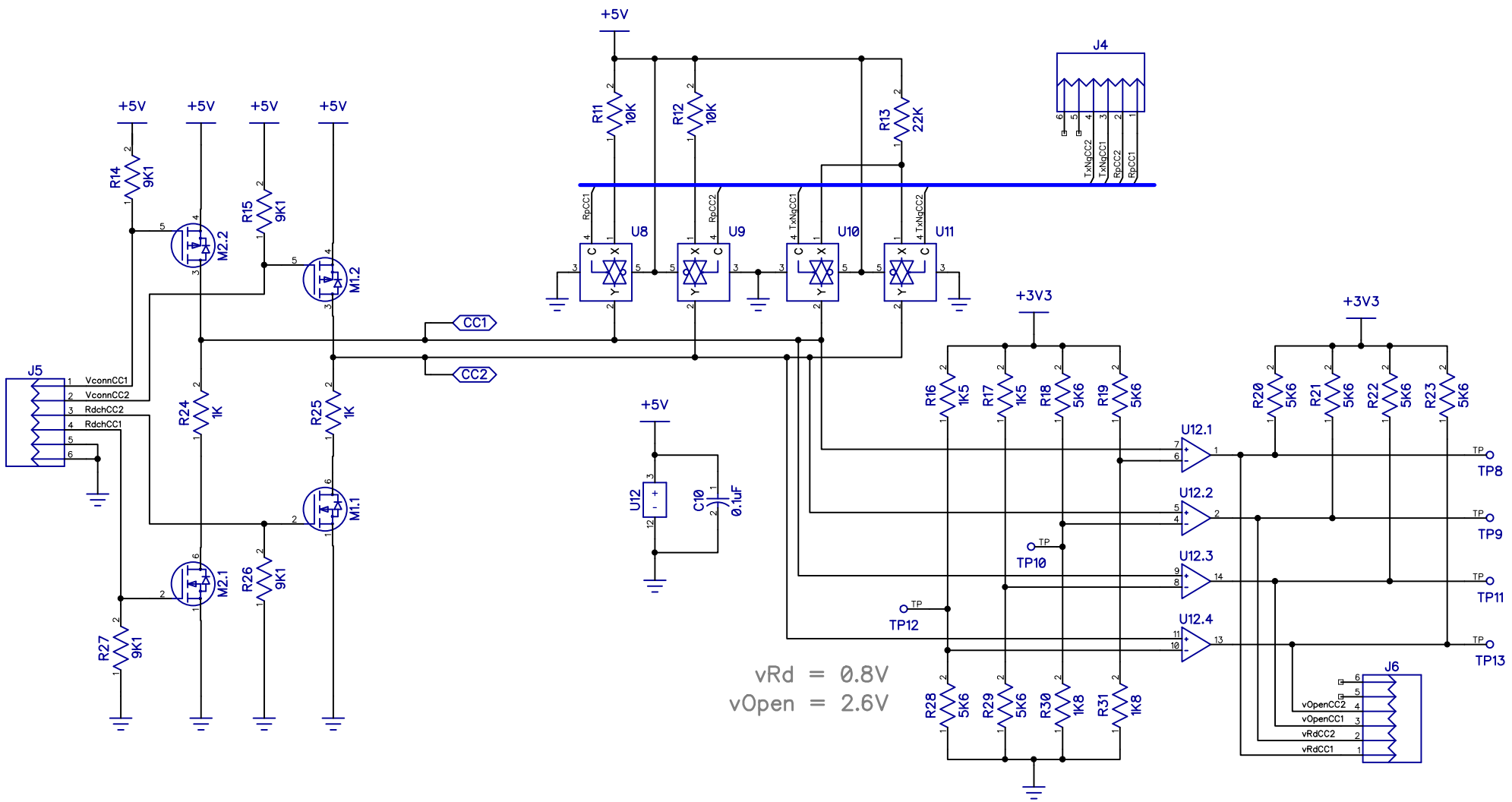
This section provides two sets of schematic diagrams. Although originally proposed as a single circuit, and treated as such in section A3, restrictions on the project's funding necessitated simplification of each circuit so that it could be laid out on a two-layer circuit board rather than the four layers initially planned.

The first set of diagrams illustrates the circuit for a BMC transceiver, including the subtractor receiver, slew rate-limited buffer transmitter, Rp–Rd signalling system, V_{CONN} drivers, and attachment-detection comparators.

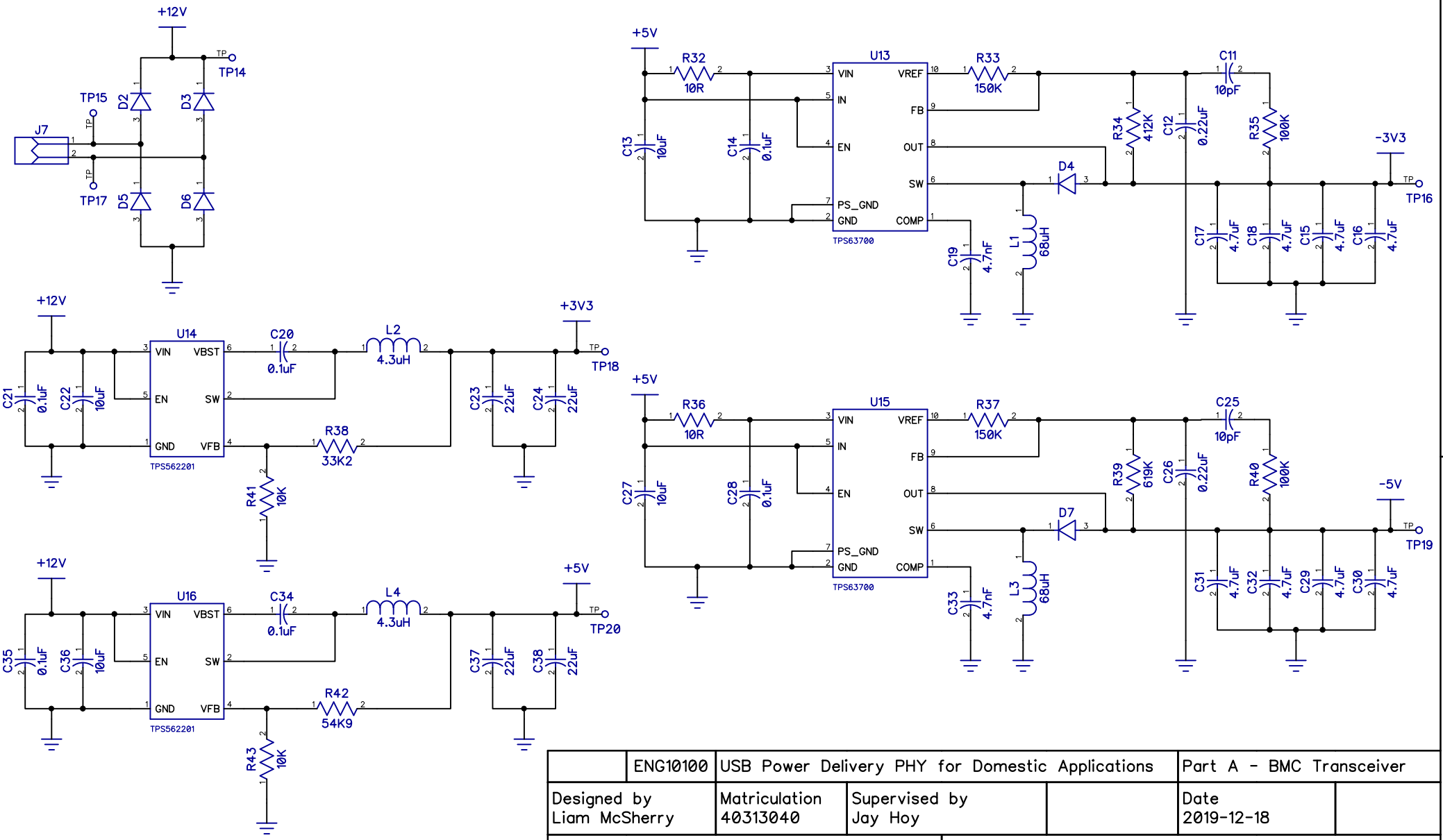
The second set provide the circuit for a BFSK transceiver, with a tuneable Colpitts oscillator, rectifier for the receiver, and filters for a PWM DAC.



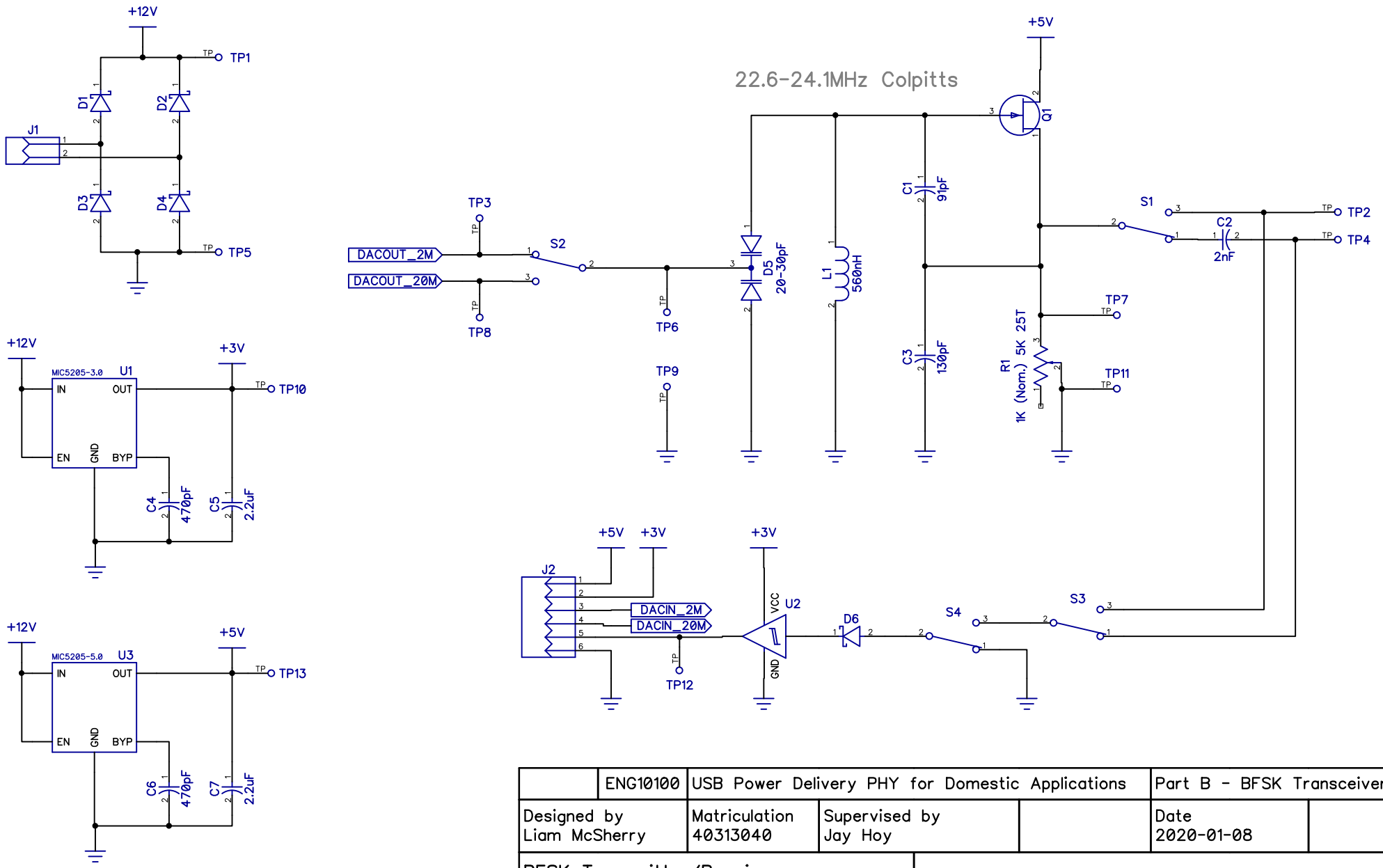
	ENG10100	USB Power Delivery PHY for Domestic Applications			Part A - BMC Transceiver
Designed by Liam McSherry	Matriculation 40313040	Supervised by Jay Hoy		Date 2019-12-18	
Transceiver Frontend (Slew rate-limited buffer output, subtractor-Schmitt trigger receiver)					Sheet 1/3



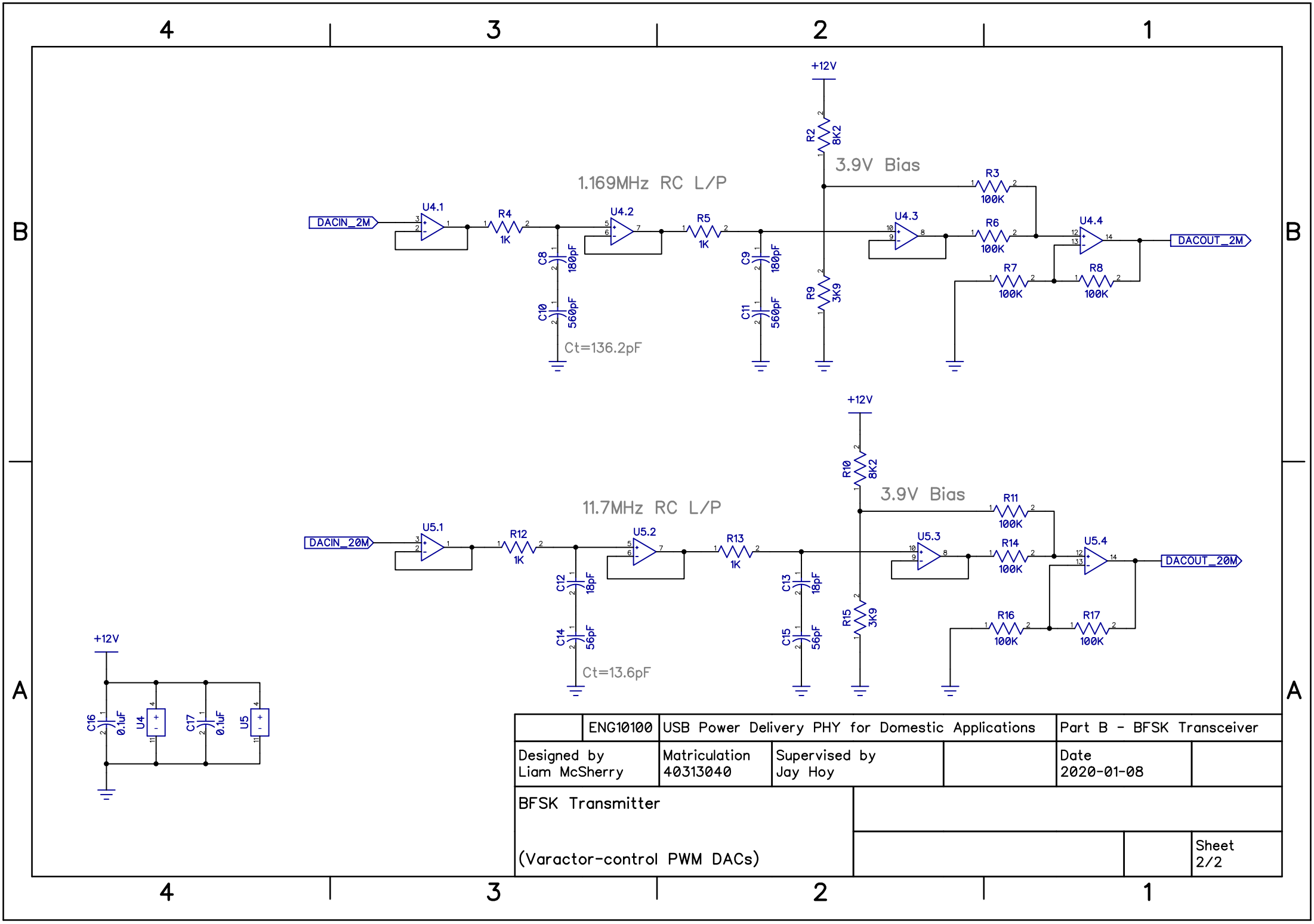
	ENG10100	USB Power Delivery PHY for Domestic Applications			Part A - BMC Transceiver
Designed by Liam McSherry	Matriculation 40313040	Supervised by Jay Hoy		Date 2019-12-18	
Rp-Rd Signalling and Vconn (Switched resistors, comparators, and Vconn push-pull drivers)					Sheet 2/3



	ENG10100	USB Power Delivery PHY for Domestic Applications			Part A - BMC Transceiver
Designed by Liam McSherry	Matriculation 40313040	Supervised by Jay Hoy		Date 2019-12-18	
Power Supplies (Incoming terminals, regulators)					Sheet 3/3



	ENG10100	USB Power Delivery PHY for Domestic Applications			Part B - BFSK Transceiver
Designed by Liam McSherry	Matriculation 40313040	Supervised by Jay Hoy		Date 2020-01-08	
BFSK Transmitter/Receiver (Colpitts oscillator, switches, 3-volt and 5-volt regulators. receiver buffer)					Sheet 1/2


+12V
R2
8K2
3.9V Bias
1.169MHz RC L/P
R3
100K
U4.1
DACIN_2M
R4
1K
U4.2
R5
1K
U4.3
R6
100K
U4.4
DACOUT_2M
C8
180pF
C9
180pF
R7
100K
R8
100K
C10
560pF
C11
560pF
R9
3K9
Ct=136.2pF
+12V
R10
8K2
3.9V Bias
11.7MHz RC L/P
R11
100K
U5.1
DACIN_20M
R12
1K
U5.2
R13
1K
U5.3
R14
100K
U5.4
DACOUT_20M
C12
18pF
C13
18pF
R15
3K9
R16
100K
R17
100K
C14
56pF
C15
56pF
Ct=13.6pF
+12V
C16
0.1uF
U4
C17
0.1uF
U5
ENG10100
USB Power Delivery PHY for Domestic Applications
Part B - BFSK Transceiver
Designed by
Liam McSherry
Matriculation
40313040
Supervised by
Jay Hoy
Date
2020-01-08
BFSK Transmitter
(Varactor-control PWM DACs)
Sheet
2/2

A2. Circuit Layout

Figure A1 shows the first circuit board, which hosts the BMC transceiver.

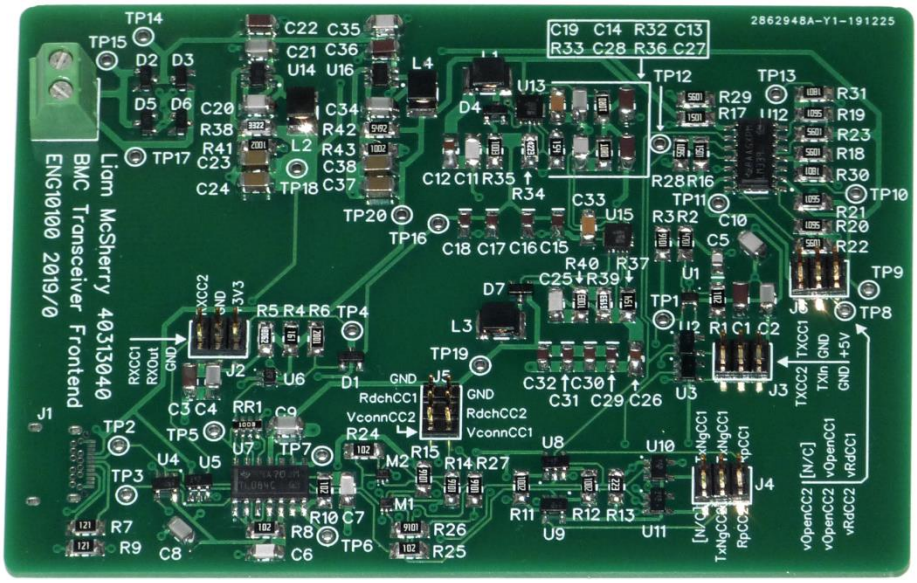


Figure A1. BMC transceiver circuit layout.

Figure A2 shows the second circuit board, which includes the BFSK Colpitts transmitter and time-averaging receiver.

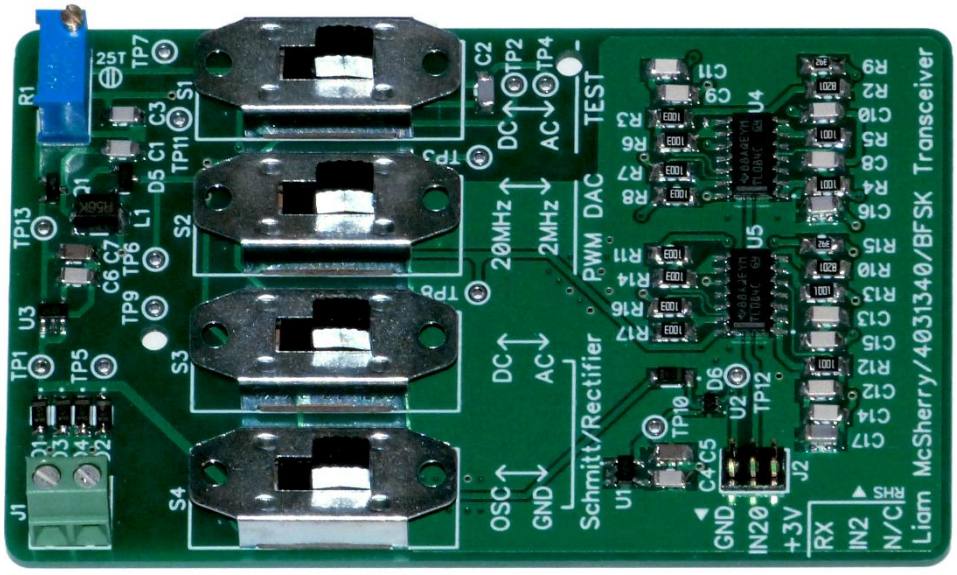


Figure A2. BFSK transceiver circuit layout.

A3. Bill of Materials

This section lists the components selected for use in the realised circuits and provides selection rationale for each major subsystem. Table A1 provides a bill of materials for the BMC transceiver circuit.

Table A1. Bill of materials for the BMC transceiver circuit.

Reference	Part Number	Unit	#	Sum
C1, C3, C13, C22, C27, C36	KEMET C1206C106J3RACAUTO	0.650	6	3.900
C2, C4, C8, C9, C10, C14, C20, C21, C28, C34, C35	Murata GRM31C5C2A104JA01L	0.348	11	3.828
C5	KEMET C0805C161J1GAC7800	0.250	1	0.250
C6	KEMET C1206C201J5GAC7800	0.310	1	0.310
C7	KEMET C1206C301J5GACTU	0.310	1	0.310
C11, C25	Yageo CC1206JRNPO9BN100	0.160	2	0.320
C15, C16, C17, C18, C29, C30, C31, C32	KEMET C0805C475J4RACAUTO	0.460	8	3.680
C19, C33	KEMET C1206C472J5RACTU	0.130	2	0.260
C23, C24, C37, C38	Samsung CL31A226MOCLNNC	0.290	4	1.160
D1, D2, D3, D4, D5, D6, D7	Diodes ZHCS1000TA	0.430	7	3.010
J1	Amphenol 10137061-00021LF	1.690	1	1.690
J2, J3, J4, J5, J6	Molex 87758-0616	0.440	5	2.200
J7	Phoenix Contact 1727010	0.970	1	0.970
L1, L3	Murata LQH43NN680J03L	0.490	2	0.980
L2, L4	Murata LQH43NH4R7J03L	0.580	2	1.160
M1, M2	ON Semi NTZD3155CT1G	0.270	2	0.540
R1, R8, R10, R24, R25	Yageo RC1206JR-071KL	0.070	5	0.350

Reference	Part Number	Unit	#	Sum
R2	Yageo RC1206FR-074K7L	0.070	1	0.070
R3, R14, R15, R26, R27	Yageo RC1206FR-079K1L	0.070	5	0.350
R4	Yageo RC1206JR-07160RL	0.070	1	0.070
R5	Panasonic ERJ-8ENF8062V	0.070	1	0.070
R6	Bourns CR1206-FX-2001ELF	0.070	1	0.070
R7, R9	Yageo RC1206JR-07120RL	0.070	2	0.140
R11, R12, R41, R43	Yageo RC1206FR-0710KL	0.070	4	0.280
R13	Yageo RC1206JR-0722KL	0.070	1	0.070
R16, R17	Yageo RC1206FR-071K5L	0.070	2	0.140
R18, R19, R20, R21, R22, R23, R28, R29	Yageo RC1206FR-075K6L	0.070	8	0.560
R30, R31	Yageo RC1206FR-071K8L	0.070	2	0.140
R32, R36	Bourns CR1206-FX-10R0ELF	0.070	2	0.140
R33, R37	Yageo RC1206JR-07150KL	0.070	2	0.140
R34	Panasonic ERJ-8ENF4223V	0.070	1	0.070
R35, R40	Yageo RC1206FR-07100KL	0.070	2	0.140
R38	Panasonic ERJ-8ENF3322V	0.070	1	0.070
R39	Panasonic ERJ-8ENF6193V	0.070	1	0.070
R42	Yageo RC1206FR-0754K9L	0.070	1	0.070
RR1	Bourns CAY16-1003F4LF	0.070	1	0.070
U1, U6	Microchip MIC920YC5-TR	0.250	2	0.500
U2, U3, U4, U5, U8, U9, U10, U11	ON Semi MC74VHC1GT66DTT1G	0.250	8	2.000
U7	Texas Inst. TL084CDR	0.320	1	0.320
U12	Texas Inst. LM339DR	0.290	1	0.290
U13, U15	Texas Inst. TPS63700DRCR	1.580	2	3.160
U14, U16	Texas Inst. TPS562201DDCR	0.510	2	1.020
				34.938

Table A2 provides a bill of materials for the BFSK transceiver circuit.

Table A2. Bill of materials for the BFSK transceiver circuit.

Reference	Part Number	Unit	#	Sum
C1	AVX 12065A910FAT2A	0.697	1	0.697
C2	AVX 12065A202JAT2A	0.340	1	0.340
C3	Murata GRM31A5C3A131JW01D	0.356	1	0.356
C4, C6	Yageo CC1206FRNPO9BN471	0.350	2	0.700
C5, C7	Taiyo Yuden EMK316BJ225MD-T	0.155	2	0.310
C8, C9	AVX 12061A181FAT2A	0.520	2	1.040
C10, C11	KEMET C1206C561FCGACTU	0.760	2	1.520
C12, C13	KEMET C1206C180GBGAC7800	0.760	2	1.520
C14, C15	KEMET C1206C560F5GACTU	1.100	2	2.200
C16, C17	Murata GRM31C5C2A104JA01L	0.400	2	0.800
D1, D2, D3, D4, D6	Nexperia BAT54GWJ	0.110	5	0.550
D5	Infineon BB844	0.279	1	0.279
J1	Phoenix Contact 1727010	0.970	1	0.970
J2	Molex 87758-0616	0.440	1	0.440
L1	Bourns CM453232-R56ML	0.300	1	0.300
Q1	ON Semi. MMBFJ309LT1G	0.360	1	0.360
R1	Bourns PV36Y502C01B00	1.140	1	1.140
R2, R10	Yageo RC1206FR-078K2L	0.080	2	0.160
R3, R6, R7, R8, R11, R14, R16, R17	Yageo RC1206FR-07100KL	0.080	8	0.640
R4, R5, R12, R13	Yageo RC1206FR-071KL	0.080	4	0.320
S1, S2, S3, S4	CW GF-124-0013	0.610	4	2.440
U1	Microchip MIC5205-3.0YM5	0.300	1	0.300
U2	Nexperia 74LVC1G17GW.125	0.210	1	0.210
U3	Microchip MIC5205-5.0YM5	0.300	1	0.300
U4, U5	Texas Instruments TL084CDR	0.320	2	0.640
				18.532

A3.1 System controller

As considered for this project, the system controller is responsible for USB-PD negotiation, controlling the transceivers, operating the power converter, and any other decision-making required.

The key requirement on the system controller is that it must be able to operate each part of the USB-PD physical layer. For the BMC transmitter, this means generating and operating to a data clock to produce the data, tristate, and Type-C control signals. As the BMC receiver is largely digital, the system controller must accept the output from the line instrumentation and internally implement the digitally-controlled oscillator, shift register, and supporting logic. Its capabilities must also extend to providing a clock multiplexer and an analogue varactor control signal for the BFSK transmitter and, for the BFSK receiver, it must be able to count high-frequency pulses up to 24.6MHz.

These requirements exclude the default choice of a microcontroller. Although versatile, well-equipped, and inexpensive, the serial nature of a microcontroller is unlikely to lend itself to the highly parallel task of operating the transceivers. A high-performance device such as the MIMXRT1011DAE5A [74] (£1.57 at 5ku), a 500MHz ARM Cortex-M7 system-on-chip, might only be able to execute 1,666 instructions in each unit interval.[104] Once line interpretation, 4b5b decoding, check-code generation, and transfer to a data structure for processing are done, a significant portion of this instruction budget is likely to be exhausted. Parts of this work can be delegated to peripherals, but enough flexibility must remain for overheads from interrupt latency and other system tasks, increasing pressure.

A requirement for high-speed parallel processing makes a programmable logic device (PLD) or a field-programmable gate array (FPGA) the next clear choice. In each case, the device contains a number of programmable logic elements which can be used to implement logic functions.[105] These are connected with a flexible communication ‘fabric’, and so portions can be configured to act independently and concurrently for faster processing. As each element is relatively simple, it can operate at high speeds into the hundreds of megahertz.

Evaluating the products available, the lowest-cost FPGAs are from Lattice Semiconductor. The ICE40UL640 (£0.863 at 1ku) and ICE40UL1K (£1.067 at 490u)[106] [46] were identified as offering the lowest cost. These provide 640 and

[104] Assuming one instruction per cycle (IPC). A precise IPC figure could not be identified.

[105] The common building block is the ‘lookup table’ (LUT), a small memory where inputs to the logic function are the address and the stored data is the output. This is incredibly powerful—a LUT4, or four-input LUT, can implement any four-variable one-output logic function regardless of its complexity [40]. LUTs are commonly arranged in ‘configurable logic blocks’ (CLBs) containing useful resources such as carry logic, multiplexers, and flip-flops.

[106] This cost is for the 36-pin package. The 16-pin package is available at lower cost (£0.914 at 1ku).

1248 logic cells, respectively, each with a LUT4, flip-flop, and carry logic. For these devices, size—which determines whether the FPGA has enough logic cells to implement the needed functionality—is the main constraint. It is difficult to predict size requirements ahead of implementation, but at least one project has used the same family of devices to implement a RISC-V processor with UART in 3000 logic cells [75]. The physical layer is thought to be far simpler than a processor, and so the ICE40UL1K should be selected as the target device.

To minimise the validation surface, the prototype is to use a Xilinx XC7A35T with 33,280 logic cells. This simplifies the prototype circuits and will enable system controller implementation even if the design exceeds 1248 logic cells.

A3.2 BMC transmitter

The design of the BMC transmitter is discussed in 5.2. It was determined there that the tristate buffer required by the USB-PD specification would be most easily implemented as a slew rate-limited buffer followed by a transmission gate. The latter is available as a singly-packaged device, while the former might use an op-amp which is surplus from an op-amp array used in the BMC receiver.

This section considers the transmitter's realisation as those two components.

A3.2.1 Slew rate-limited buffer

The slew rate-limited buffer is straightforward: an RC filter slows the rise and fall times, a unity-gain amplifier provides impedance isolation, and a transmission gate allows the buffer to enter a high-impedance output state.

The discussion in 5.2.2 considers an RC filter with a 750-ohm resistor and a 220pF capacitor. The fabricator did not stock this resistance, and so revising to use a stocked 1kΩ resistance requires a 165pF capacitor to achieve the same time constant, or 160pF if rounding to the preferred series [54]. With tolerance of ±5%, this gives a rise time of 317–388ns and slew rate of 2.32–2.84 volts/µs.

Certain capacitors exhibit a reduction in capacitance with increasing DC bias [76], and so the part chosen should use a 'C0G' dielectric or equivalent. This dielectric offers a flat response to DC bias.

To meet USB-PD's requirements, the combined output impedance of the amplifier and transmission gate must not exceed 75 ohms. Negative feedback in a unity-gain amplifier makes its output impedance negligible,[107] and so selection can be based on the required gain–bandwidth product (GBW) and slew rate. The MIC920 [77] was selected in A3.3.2 and, while far more capable than required, its use here simplifies the Bill of Materials for the prototype.

[107] See Neamen [61] at p. 625.

The MIC920 is specified at ±5 volts and so is best configured as a fractional-gain amplifier. This retains negligible output impedance. If output from the system controller is 3.3 volts, a divisor of 2.93 will produce 1.126 volts—near the nominal swing of 1.125 volts. Arbitrarily selecting a preferred value of 4.7kΩ, the ideal counterpart would be 9.07kΩ. Using the nearby preferred value of 9.1kΩ, the divisor is 2.936 and output is 1.124 volts. In isolation, ±5% tolerance is enough to maintain the required output voltages. However, if the supply variation is ±5%, a tolerance no wider than ±1% would be necessary and would give a worst-case output range of 1.054 to 1.196 volts (for a permitted 1.05 to 1.20 volts).

A3.2.2 Transmission gate

Although the discussion in 5.2.2 suggests use of the 74AHC1G66 transmission gate, its high-level input voltage $V_{IH} \geq 3.85$ volts when supplied from 5.5 volts is higher than the 3.3-volt system controller output voltage and so it is not suitable for use. An alternative can be found in the MC74VHC1GT66 [78] (4.2p at 6ku), which has $V_{IH} \geq 2.0$ volts. Although its on-resistance of 45 ohms is greater than for the 74AHC1G66, it remains low enough to satisfy USB-PD's requirements.

As a brief note, while the 74AHC1G66's input threshold can be reduced with a lower supply voltage, this would necessitate another regulator and would raise its on-resistance to up to 70 ohms.

A3.3 BMC receiver

The BMC receiver is considered in 5.3. As a largely digital system, much of it is realised as part of the system controller; only its line instrumentation subsystem is separate from the system controller, and it is that subsystem that is discussed here. As 5.3.3 details, the line instrumentation has two key parts: the subtractor and the hysteresis comparator (or Schmitt trigger).

A3.3.1 Subtractor

The subtractor is a pair of low-pass filters feeding into a difference amplifier. The USB-PD specification [9] at D.2.1 suggests time constants of 200ns and 300ns for the filters and, as one of the filters is on the output of the other, it is necessary to insert buffers between them to ensure that they do not interact with each other. A further buffer is required on the output of the cascaded filter to avoid interaction with the difference amplifier's bias and feedback resistors.

If an RC low-pass filter is used with 1kΩ resistance, capacitances of 200pF and 300pF are required. Both are preferred values. As in A3.2.1, a flat response to DC bias is desirable and so 'C0G'-dielectric capacitors should be selected.

If the difference amplifier is configured for unity gain, it requires four matched resistances. A failure to match may alter gain or introduce signal asymmetry. A fine-tolerance resistor array will provide matching in a small package, and so is a

suitable option. A high resistance is preferable to avoid significant voltage drop across the bias resistors, and so the Bourns CAY16-1003F4LF [79] (0.71p at 10ku)—a four by 100kΩ ±1% array—is a reasonable choice.

As 5.3.3 establishes, the input signal is limited to 3 volts/µs slew and bandwidth around 882kHz. An array of op-amps with this slew rate and GBW of 1MHz or more will be suitable and provide cost reduction.[108] The TL084 [80] (11.4p at 1ku), a quad op-amp with 3MHz GBW and slew rate greater than 5 volts/µs, is a low-cost choice with a high input resistance of 1TΩ.[109]

A3.3.2 Schmitt trigger

The Schmitt trigger produces the square wave processed by the receiver logic and, as 5.3.3 discusses, an important requirement is that it can perform a rail-to-rail swing within 378ns. The system controller's 3.3-volt input voltage means this is a swing of 6.6 volts, and so a slew rate of at least 17.5 volts/µs is needed. This was verified in simulation. Square wave output requires bandwidth higher than the fundamental frequency; simulation suggests a reasonable wave is obtained with frequencies up to the ninth harmonic, and so a GBW above 6MHz will be suitable with the maximum data clock of 660kHz. These criteria identified the MIC920 [77] (19.3p at 3ku), which has a GBW of 80MHz and a slew rate of 1350 volts/µs typical when supplied at ±5 volts.

The feedback and bias resistors need no change from 5.3.3.

As the output of the Schmitt trigger will swing negative, a diode should be added before the input to the system controller to provide protection.

Cost savings may be possible if the MIC920 can be replaced by a comparator, as these are simpler devices designed to enter saturation quickly [81]. However, many comparators require overdrive—where input exceeds the threshold by a minimum amount—to achieve their rated propagation delay, which not be easily attained with the low signal level from the subtractor.

A3.4 BFSK four-oscillator transmitter

> **Note:** The changes to the schedule discussed in 9.1 prevented realisation of the four-oscillator transmitter. This section is omitted for brevity.

[108] This was confirmed in simulation by comparing against a 100MHz GBW, 50 volts/µs op-amp.

[109] Note that requirement 3.3.3 sets a minimum of 1MΩ.

A3.5 BFSK Colpitts transmitter

The design of the Colpitts-oscillator BFSK modulator is discussed in 6.2.1. Its circuit can be considered in two parts: the Colpitts oscillator and the varactor control subsystem.

A3.5.1 Colpitts oscillator

The Colpitts oscillator is based around a resonant LC network, meaning its frequency is given by equation 5. Rearranging this allows either capacitance or inductance to be calculated if the other is known. An inductor is typically a more expensive component, and so it is sensible to select it first. A parametric filtering tool was used to identify the most common preferred values of inductance below 1µH, excluding inductors which were self-resonant below 29MHz,[110] toleranced looser than ±5%, and which were not specified near the carrier frequency. This identified 560nH as a particularly common value, and so the capacitance needed for resonance at 23.2MHz is 84.04pF.

The choice of the other tuning elements depends on the transistor. The gate of a JFET has inherent capacitances which will sum with those setting the resonant point, changing output frequency. The MMBFJ309L [60] (9.9p at 3ku) is a radio-frequency JFET with an input capacitance C_{iss} of 5pF—a not insubstantial portion of the total 84.04pF required. Of the devices rated for the frequencies BFSK uses, it was identified as having the lowest unit cost.

As fixed tuning elements, the capacitors must be selected last to make up the difference from the varactor capacitance. The range from 23.7MHz to 22.7MHz requires capacitances between 80.53pF and 87.78pF, and so a varactor with a characteristic spanning 10pF is desirable. Its control voltage should also be low enough that no other voltage regulators are required, and certainly less than 20 volts (the maximum USB-PD V_{BUS} voltage). A common-cathode pair of varactors would also be desirable to reduce distortion and parasitics [82], as well as to lower cost and provide matching. These criteria identified the BB844 [83] (7.8p at 3ku), which has a tuning range of 11.5–43.75pF at bias between 8 and 2 volts.

Fixed capacitors totalling around 60pF would place the required capacitances near the centre of the varactor's tuning range. Equation 6 in mind, an initial estimate for the two series capacitances may be 130pF and 180pF. This gives a total of 75.48pF, higher than the target. Lowering to 91pF and 130pF, a more suitable 53.5pF is obtained.

The JFET must then be biased to fulfil equation 6. The capacitance ratio is 0.70 nominally and 0.78 in the worst case with +5% tolerance. A common-drain JFET

[110] A margin of 125% over the nominal 23.2MHz carrier. Beyond the self-resonant point, capacitance in the device dominates and it ceases to act as an inductor.

amplifier inherently provides gain below 1,[111] and so its transconductance can be any value that will provide gain above 0.78. The midpoint of the MMBFJ309L's range is 13mS, and this is achieved at drain current around 8mA.[112] Using a 5-volt supply, this is a source resistance of 625 ohms and hence gain around 0.89. If the fabricator's stocked 1kΩ resistances are used, a marginally higher gain of 0.91 can be achieved at drain current 5mA and transconductance 10mS.

In simulation with a model of the MMBFJ309L and the varactor replaced by fixed capacitances, this produced output with a peak-to-peak amplitude of 4.3 volts.

In a commercialised circuit, a post-amplifier would be needed to scale this output to the 150mV$_{RMS}$ nominal from USB-PD. The automatic gain controller will not be included in the prototype (see A3.6) and so, with DC offset removed and a diode added to clip negative swing, the oscillator's output could be fed into the receiver without attenuation.

A3.5.2 Varactor control

An analogue voltage is required to control the varactor, and a low-cost method of producing this voltage is the PWM DAC. The key parameters when applying the PWM DAC are resolution, settling time, and ripple.

Resolution determines the fineness of the adjustment in oscillator frequency. It was estimated in 6.2.3 that a resolution of up to seven bits might be required for a typical varactor, although the BB844's datasheet shows that a much coarser resolution would be viable: the 22pF point[113] occurs around 5 volts, the 25.5pF point around 4.3 volts, and the 29.3pF point around 3.9 volts.[114] Assuming the system controller's output were given a 3.9-volt DC offset, 5-bit resolution would give 103mV granularity—enough to locate each point. A higher resolution can be achieved with a higher generator clock frequency: a 1MHz PWM signal driven by a 100MHz clock can have 100 different duty cycles, giving 6.6 bits; if a 50MHz clock is used, there are 50 duty cycles and resolution is 5.6 bits. Regardless, while obtaining the ideal bias may not be possible, the 450–600kHz deviation that USB-PD requires should be easily achievable.

An RC filter is suggested in 6.2.3 as the method for obtaining a DC voltage. As five time constants are required to reach 99% of the input level, the filter—and hence the PWM frequency—determines the settling time of the DAC. A higher frequency enables a wider passband with a shorter time constant but, with a practical limit

[111] See Neamen [61] at equation 4.33(a) on p. 228. As shown in figure 14, the drain is common here.

[112] See its datasheet [60] at Figure 2.

[113] Equivalent to 23.7MHz operation: 80.53pF is required, the capacitors and JFET contribute 58.5pF.

[114] The latter two are equivalent to 23.2MHz and 22.7MHz, respectively.

of 250MHz on the generator clock,[115] reduces resolution. Assuming a signalling frequency must be present for 3.03µs—the minimum data period—a settling time no longer than 270ns stretches the overall period to the 3.3µs nominal and would mean a time constant of 54ns or less. This would require a PWM frequency of 18.5MHz, which plainly limits resolution far below that required.

However, it may be possible to exploit the filter's exponential characteristic to reduce settling time: five time constants are needed to reach 99% output, but 0.7 are required for output to reach 50%. By overdriving the DAC at twice the duty cycle required, the settling time can become 0.7 time constants. The maximum output required is 5 volts which, for 3.3-volt input with a 3.9-volt offset, means a non-overdriven duty cycle of 33%. It is clearly within capability to double this. If it is assumed that 600ns of the 3.3µs period can be taken up by a frequency shift, an overdriven filter with a 1.167MHz passband will settle within 600ns and offer sufficient headroom to have 2MHz PWM generated by a 200MHz clock, giving a resolution of 6.6 bits.

The final parameter for the PWM DAC is ripple. The period of zero-volt output inherent to PWM partially discharges the filter capacitors, decreasing output until the next 'mark' period recharges them. In a BFSK modulator, this would result in frequency ripple in the output. However, as both the XC7A35T and the ICE40UL1K can place their outputs in a high-impedance state, it may be worth investigating whether this can reduce discharge and maintain a constant filter output. In simulation with four cascaded 1.5MHz filters with 2MHz PWM, a drop of around 10% was observed after tristating the output but output was otherwise constant. If this can be replicated in a practical circuit, the technique is viable.

Simulation also highlighted that, with 2MHz PWM, the rate of change in output may not be sufficient. Although resolution-limited, 20MHz PWM gave substantial performance improvements. It may be desirable to investigate both. Attempting control with modulation schemes other than PWM may also be worthwhile.

A3.6 BFSK automatic gain controller

The basic operation of the automatic gain controller (AGC) is shown in 6.3.2. Its purpose is to produce constant-amplitude output from variable-amplitude input, simplifying later processing steps.

In theory, figure 18 can be translated almost directly into a circuit: the summing junction becomes a difference amplifier, K is an op-amp taking a reference and the error output from the difference amplifier, and $G(k)$ is replaced with a fixed-gain amplifier and a variable attenuator. The variable attenuator might use a FET as one part of a voltage divider, for example. The envelope detector might be an

[115] See the ICE40UL1K datasheet [46] at Table 4.13.

RC filter fed through a diode. In this circuit, the fixed-gain amplifier makes the output amplitude greater than desired. This produces a large error which causes the op-amp to drive the transistor such that attenuation is increased, reducing the error until the difference between the reference and output envelope is zero.

A circuit following this description was implemented, but with limited success. It appeared to demonstrate variable attenuation, but over a very limited range. In one case, constant output was only maintained over input of 3.8 to 4.2 volts. The transistor appeared to become 'wound up'—either fully open or fully closed—if operated outside this range, providing fixed attenuation. The reasons for this behaviour are not clear, and insufficient time was available to troubleshoot. As the output of the Colpitts oscillator will be suitable for use by the receiver with its DC offset and negative portions removed, this has little impact on prototyping.

A3.7 BFSK squelch detector

The purpose of the squelch detector is to determine whether a valid signal is present, enabling the receiver as appropriate. In a prototype, transmissions will be produced as well as received by the system controller, and so this subsystem is unnecessary—if needed, an internal signal could be used instead. As 6.3.5 sets out, it is a straightforward system and so there is limited loss in omitting it.

A3.8 BFSK time-averaging receiver

The time-averaging BFSK receiver is a largely digital system, and so much of it is implemented internally to the system controller. The only circuit components it requires are a diode to clip negative portions of the input wave and a Schmitt trigger to recover a high-quality square wave from the received signal.

The prototype does not include an automatic gain controller (see A3.6) and so a diode with low forward voltage is desirable to ensure reasonable input to the Schmitt trigger. In a final design, as current will be negligible, the AGC could simply be adjusted to provide output which negates forward voltage. Schottky diodes are widely available and have low forward voltages, and so a device such as the BAT54GW [84] (1.77p at 6ku) is a simple choice.

Given the signal shape and amplitude, the precise thresholds for the Schmitt trigger are not as important here as they are in the BMC receiver in A3.3.2. This means a monolithic device such as the 74LVC1G17 [85] (2.9p at 6ku), with its positive- and negative-going thresholds of 1.29–1.71 volts and 0.88–1.24 volts, is suitable. The half-cycle duration of a 24.6MHz sine wave is 20.3ns, and so the worst-case propagation delay of 7ns at 3.3 volts presents no issues.

A3.9 BFSK frequency–voltage receiver

> **Note:** The changes to the schedule discussed in 9.1 prevented realisation of the frequency–voltage receiver. This section is omitted for brevity.

A3.10 BFSK bandpass filter

> **Note:** The changes to the schedule discussed in 9.1 prevented realisation of the BFSK bandpass filter. This section is omitted for brevity.

A3.11 Type-C signalling

The USB Type-C signalling system is straightforward, with three subparts: the Rp switches, the attachment detectors, and the V_{CONN} power and discharge switches.

A3.11.1 Rp switches

The discussion in 7.1 considers that the transmission gate used for the BMC transmitter would work equally well for switching in the Rp resistors. A single device, the MC74VHC1GT66, was selected in A3.2.2. It was considered an array might offer cost savings, but the common ‘4066 and ‘4316 devices had input thresholds higher than 3.3 volts or on-state resistances greater than 600 ohms, the latter of which would violate output impedance limits.

Type-C permits variation of ±5% for the Rp resistances,[116] but the added 40-ohm on-state resistance of the MC74VHC1GT66 necessitates resistors with a finer tolerance. For values 22kΩ and 10kΩ, tolerances finer than ±4.6% will comply.

A3.11.2 Attachment detectors

The attachment detectors are simple comparators, as 7.1 sets out. Using the LM339 [29] quad comparator from 6.3.1 would simplify the Bill of Materials.

The attachment detectors determine whether a sink or an electronic marker is detected, or if nothing is detected. The relevant thresholds are set out in the Type-C specification: less than 0.8 volts is an electronic marker, less than 2.6 volts is a sink, and more than 2.6 volts means nothing is connected.[117] These states can be detected with two comparators for each ‘CC’ wire: no output means a sink is connected, one output means an electronic marker, and both means nothing is connected. Voltage dividers can be used to produce the references.

For a 3.3-volt supply, thresholds of 0.8 and 2.6 volt require ratios of 3.125:1 and 0.269:1 in the dividers. For the former, preferred values of 5.6kΩ and 1.8kΩ give a

[116] See BS EN IEC 62680-1-3 [12] at Table 4-20.

[117] See BS EN IEC 62680-1-3 [12] at Table 4-30.

ratio of 3.111:1 and a threshold of 803mV; for the latter, 1.5kΩ and 5.6kΩ give 0.268:1 and 2.603 volts. A tolerance of ±5% would put the minimum thresholds near the maximums that attached sinks and markers could produce. If further variation applies, such as in the supplies or the comparators, a threshold below those maximums could result in incorrect detection of cable orientation. A finer tolerance such as ±2% would avoid this issue.

The LM339 has open-collector outputs, and so pull-up resistances are needed. It would simplify the Bill of Materials to use the 5.6kΩ resistors.

A3.11.3 Cable power and discharge drivers

The V_{CONN} power and discharge drivers can be implemented with a pair of complementary FETs, as 7.2 discusses. The NTZD3155C [16] (5.21p at 4ku) has a logic-level gate, rated currents of 540mA and −430mA, and protection on its gates against electrostatic discharge (ESD). The 1.2-ohm on-state resistance at a ±2.5-volt gate drive will result in dissipation of 147mW at 350mA—far lower than is likely to present issues. Protection against ESD is also highly desirable where connections to its gate will be made by hand, as in a prototype.

As the discharge resistance can be between 30 and 6120 ohms, using the 1kΩ resistor available from the fabricator's stock is permissible.

A3.12 Power supply

The prototype design relies on several power rails: +5 volts for V_{BUS} and various components, −5 volts for the BMC receiver, +3.3 volts for interfaces to the system controller, and −3.3 volts for the BMC receiver's Schmitt trigger. Although having many supplies will complicate the circuit design, their inclusion simplifies the choice of components and avoids the need for workarounds. The configuration of the supplies on the prototype is likely to differ substantially from that when a power converter is present, and so minor added complexity is not a concern.

A3.12.1 Protection

In a test environment, a supply free from mains transients can be used. As such, no protection against transient overvoltage is necessary in the prototype. The inclusion of reverse-polarity protection, however, is likely to be worthwhile. If a bridge rectifier were included on the incoming supply terminals, this risk would be mitigated. Power consumption in a prototype will not be high enough to make power dissipation in the diodes a significant issue.

A3.12.2 Positive-supply regulators

If a 12-volt supply is used, producing +5 and +3.3-volt supplies will require that 7 to 8.7 volts is removed by the regulator. At an output of 200mA, a linear regulator would dissipate over a watt. A switching regulator is the better choice here.

A device such as the Texas Instruments TPS562201 [86] provides a high current

capability, integrated power switches, and simple configuration. Its cost is not a concern as its use will only be in the prototype. To obtain the desired output, it requires configuration resistors in ratios of 3.297:1 (for 3.3 volts) and 5.51:1 (for 5 volts).[118] The datasheet provides recommendations for achieving those values.

A3.12.3 Negative-supply regulators

As no negative supply will be present and given the current demand will be greater than tens of milliamps, a switching regulator is also the most suitable choice for the negative supplies. The TPS63700 can produce both −3.3 and −5-volt supplies from 5-volt input. Its datasheet provides parts selection guidance.[119]

For −3.3-volt operation, the resistances are R_2 = 150kΩ and R_3 = 412kΩ, the inductor is 68µH, and the output capacitors are $C_{min} \geq 4.3\mu F$ for 10mV ripple.

For −5-volt operation, a resistance change to R_3 = 619kΩ is necessary. All other components can remain the same, although output ripple will increase by 3mV.

No other components need change.

[118] See its datasheet [86] at § 8.2.2.

[119] See its datasheet [87] at § 8.2.2.

Appendix B

System Controller

B1. Overview

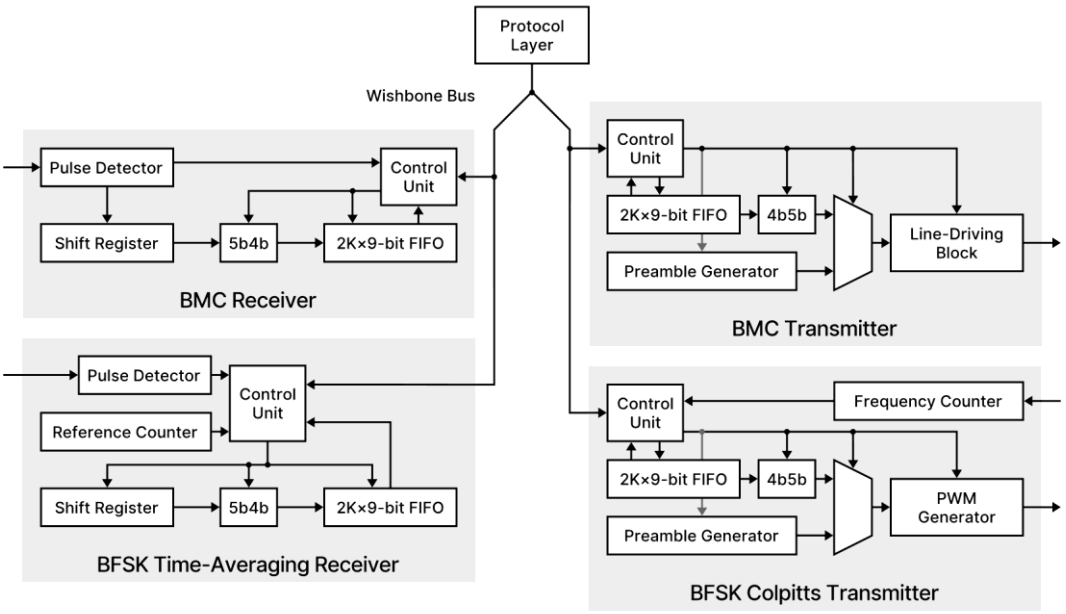


Figure B1. A high-level block diagram of the system controller.

The system controller is responsible for operating the major subsystems within the USB-PD plug socket. In a prototype system, its role is limited to operating the prototype circuits described in Appendix A to the extent required to assess their suitability to application in the plug socket.

The system controller selected in that appendix is a Lattice ICE40UL1K field-programmable gate array (FPGA), although a Xilinx XC7A35T is to be used in prototyping. The substantially increased resources available in the XC7A35T mean that 'first-generation' prototyping can proceed without being limited by the capabilities of the device, deferring optimisation to a future stage.

An FPGA requires a shift in thinking. The firmware for a microcontroller is a series of sequential steps, while the hardware description language (HDL) for an FPGA design describes logic circuits which operate concurrently and in parallel.

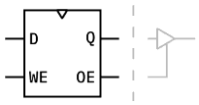
This appendix provides and documents the functional specifications for the components in the system controller. Figure B1 shows a simplified overview.

B2. Biphase Mark Code Transmitter

B2.1 Transmitter line-driving block

```
entity BiphaseMarkDriver is
port(
      CLK    : in   std_logic;
      D      : in   std_logic;
      WE     : in   std_logic;

      Q      : out  std_ulogic;
      OE     : out  std_ulogic
      );
end BiphaseMarkDriver;
```



B2.1.1 Signals

CLK
: The data clock. Its period is the BMC unit interval.

D
: The data input to the block.

WE
: Write enable; the signal indicating data is present on the data input.

Q
: Output; the logic state the line driver is to drive when enabled.

OE
: Output enabled; the signal indicating when the line driver is to be enabled.

B2.1.2 Functional description

The line-driving block (LDB) generates control signals for a tristate line driver such that the driver produces biphase mark-coded (BMC) output which complies with BS EN IEC 62680-1-2 [9].

The LDB can be taken to operate in two states. In the initial state, *WE* is evaluated on the rising edges of *CLK* and output is disabled (*OE* is unasserted). Asserting *WE* causes the LDB to transition to the transmitting state. Data to transmit is read from *D* on each rising edge where *WE* is asserted. In the transmitting state, when the LDB is ready to control the line driver, it asserts *OE* and writes to *Q*. The values on *OE* and *Q* may change on both rising and falling edges. Deasserting *WE* indicates the end of the transmission to the LDB. When ending a transmission, the LDB will continue to assert *OE* and write to *Q* until all data is processed. The LDB deasserts *OE* when it has returned to the initial state.

Figure B2 illustrates an example transaction where *10* is transmitted.

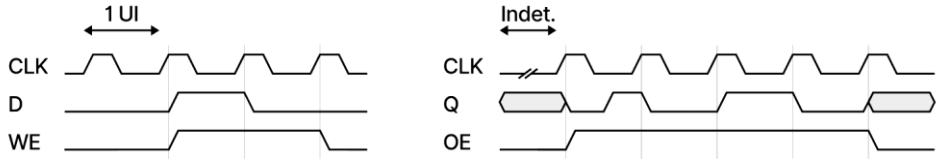


Figure B2. Biphase mark-coded transmission of sequence 10.

Under BMC, *Q* inverts at the end of each UI and the value transmitted within a UI is determined by the presence or absence of a transition at time 0.5 UI: absence is used to encode logic low, and presence logic high. USB-PD specifies additional constraints: *Q* must begin at logic low; *Q* must invert after the final value is sent; if that inversion results in *Q* = *1* (as above), *Q* is held at *1* for one UI and then at *0* for one UI before transmission ceases; and if that inversion results in *Q* = *0*, *Q* is held at *0* for one UI before transmission ceases. Figure B3 illustrates this latter case.

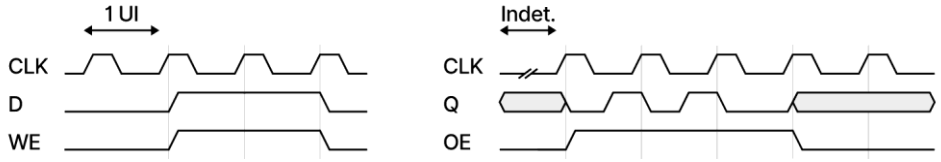


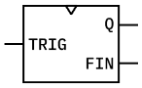
Figure B3. Biphase mark-coded transmission of sequence 11.

In each case, the time between the LDB accepting input on *D* and *WE* and generating output on *Q* and *OE* is indeterminate. A consumer must rely on the *OE* signal to determine the state of the LDB. In areas where *Q* is shown in solid grey, its value is unspecified. If *WE* is deasserted and reasserted before *OE* returns to zero, behaviour is unspecified.

B2.2 Preamble generator

```
entity PDPreambleGen is
port(
      CLK    : in   std_logic;
      TRIG   : in   std_logic;
      RST    : in   std_logic;

      Q      : out  std_ulogic;
      FIN    : out  std_ulogic
      );
end PDPreambleGen;
```



B2.2.1 Signals

CLK
: The data clock. Its period is the BMC unit interval.

TRIG
: The trigger which causes the generator to begin producing output.

RST
: Reset; voids any previous trigger and returns the generator to idle.

Q
: Output; the current bit value of the preamble to be transmitted.

FIN
: Final bit; driven high when the value on *Q* is the final bit of the preamble.

B2.2.2 Functional description

The preamble generator produces a USB-PD preamble—a 64-bit pattern which alternates between *0* and *1*—when triggered by a logic high on *TRIG*.

Bit 0 of the preamble is available on *Q* at the same instant *TRIG* is asserted. The value on *Q* advances through the preamble on each *CLK* rising edge. When bit 63 is presented on *Q*, the final bit signal *FIN* is asserted. *FIN* is deasserted on the next *CLK* edge, and the preamble generator will idle until retriggered.

Figure B4 demonstrates this behaviour.

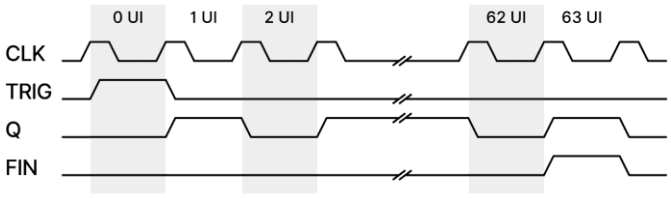


Figure B4. The output waveform for the USB-PD preamble generator.

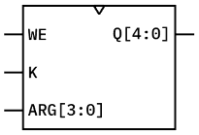
The output of the preamble generator is not biphase mark-coded.

The reason for output on *Q* to begin at the same instant *TRIG* is asserted is related to the intended application of the preamble generator. It is intended that it will be triggered by an external actor writing to a buffer, and so minimising the time between a buffer write and output reduces the likelihood of overflow.

B2.3 4b5b encoder

```
entity Encoder4b5b is
port(
      CLK    : in   std_logic;
      WE     : in   std_logic;
      K      : in   std_logic;
      ARG    : in   std_logic_vector(3 downto 0);

      Q      : out  std_ulogic_vector(4 downto 0)
      );
end Encoder4b5b;
```



B2.3.1 Signals

CLK
: The encoder data clock.

WE
: Write enable; indicates that input is present on *K* and *ARG*.

K
: When writing is enabled, indicates whether that *ARG* is a K-code or data.

ARG[3:0]
: The four-bit bus which provides the data to be encoded.

Q[4:0]
: Output; a five-bit bus providing the 4b5b line-coding of the data.

B2.3.2 Functional description

The 4b5b encoder translates four bits of raw data to the five-bit line symbols specified in BS EN IEC 62680-1-2 [9] at § 5.3.

The encoder evaluates its inputs on the rising edges of *CLK*. When *WE* is asserted, the values on *K* and *ARG* are processed. If *K* = *0*, *ARG* is taken to represent four bits of raw data and the appropriate symbol is presented on *Q* on the next *CLK* cycle. If instead *K* = *1*, *ARG* is taken to represent a K-code[120] and, on the next clock cycle, the K-code indicated is presented on *Q*.

Table B1 maps *ARG* states with *K* = *1* to K-codes.

As USB-PD specifies only six of a possible 16 K-codes, it is possible to provide invalid input to the encoder. Output on *Q* for invalid input is not specified.

[120] A five-bit code which does not map to data but is sent in-band to represent a control signal.

Table B1. Map of 4b5b encoder input to K-codes.

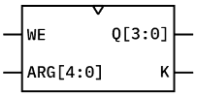
ARG[3:0]	K-code
0000	Sync-1
0001	Sync-2
0010	Sync-3
0011	RST-1
0100	RST-2
0101	EOP

The intention in remapping the values of K-codes is to enable easy adaptation for a future purpose. This may be related to USB-PD or may be a means of conveying control signals specific to the BMC transmitter. In addition, no input validation is included because it is intended that the encoder will be a component of a larger system which will provide validation and exception-raising.

B2.4 4b5b decoder

```
entity Decoder4b5b is
port(
     CLK    : in  std_logic;
     WE     : in  std_logic;
     ARG    : in  std_logic_vector(4 downto 0);

     Q      : out std_ulogic_vector(3 downto 0);
     K      : out std_ulogic
     );
end Decoder4b5b;
```



B2.4.1 Signals

CLK
: The encoder data clock.

WE
: Write enable; indicates that input is present on *ARG*.

ARG[4:0]
: The five-bit bus which provides the data to be decoded.

Q[3:0]
: Output; a four-bit bus providing the decode of the 4b5b line-coded input.

K
: K-code; when high, indicates that the decoded input was a K-code.

B2.4.2 Functional description

The 4b5b encoder translates five-bit symbols specified in BS EN IEC 62680-1-2 [9] at § 5.3 into their four-bit equivalent symbols.

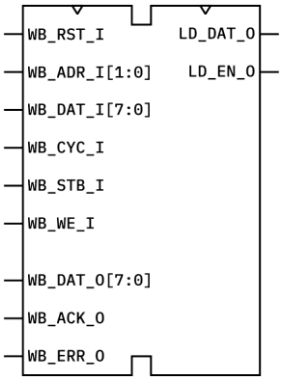
The decoder evaluates its inputs on the rising edges of *CLK*. When *WE* is asserted, the value of *ARG* is processed and the decoded output is presented on *Q* on the next *CLK* cycle. *K* indicates whether the value on *Q* is a K-code: if high, the output is a K-code number from B2.3.2; and if low, the output is raw data.

As USB-PD does not specify all 32 possible symbols, it is possible to provide the decoder with invalid input. Output on *Q* for invalid input is not specified.

B2.5 Control unit

```
entity BiphaseMarkTransmitter is
port(
   WB_CLK   : in  std_logic;
   DCLK     : in  std_logic;
   WB_RST_I : in  std_logic;
   WB_ADR_I : in  std_logic_vector(1 downto 0);
   WB_DAT_I : in  std_logic_vector(7 downto 0);
   WB_CYC_I : in  std_logic;
   WB_STB_I : in  std_logic;
   WB_WE_I  : in  std_logic;

   WB_DAT_O : out std_ulogic_vector(7 downto 0);
   WB_ACK_O : out std_ulogic;
   WB_ERR_O : out std_ulogic;
   LD_DAT_O : out std_ulogic;
   LD_EN_O  : out std_ulogic
   );
end BiphaseMarkTransmitter;
```



B2.5.1 Signals

WB_CLK
: The clock used by the Wishbone bus interface (*i.e.* *CLK_I*).

DCLK
: Data clock; sets the rate at which the transmitter operates. Its period is a BMC unit interval.

WB_RST_I
: The Wishbone reset signal.

WB_ADR_I[1:0]
: The Wishbone address bus.

WB_DAT_I[7:0]
: The Wishbone data-in bus.

WB_CYC_I
: The Wishbone cycle-active signal.

WB_STB_I
: The Wishbone strobe (select) signal.

WB_WE_I
: The Wishbone write-enable signal.

WB_DAT_O[7:0]
: The Wishbone data-out bus.

WB_ACK_O
: The Wishbone slave acknowledge signal.

WB_ERR_O
: The Wishbone slave error signal.

LD_DAT_O
: Line driver data output; when enabled, indicates the value the line driver should place on the line.

LD_EN_O
: Line driver enable; when enabled, indicates the line driver should place *LD_DAT_O* on the line.

B2.5.2 Functional description

The control unit (CU) co-ordinates the internal components of the transmitter and exposes a Wishbone Classic [69] interface to allow connection within a larger system. The Wishbone interface supports the 'SINGLE' and 'BLOCK' read and write bus cycles with an 8-bit data port and 2-bit addresses.

The CU is modelled as a set of registers. Table B2 describes the register set.

Table B2. Register map for the BMC transmitter control unit.

Address	Action	Name	Description
`00`	W	`KWRITE`	Values written to this register are added to the TX buffer as a K-code. Table B1 lists the values which are valid.
`01`	W	`DWRITE`	Values written to this register are added to the TX buffer as raw data.
`10`	R	`STATUS`	Provides status flags useful for flow control. `[0]` `FULL` Set if the TX buffer is full. `[1]` `FILLING` Set if the TX buffer is half full or more than half full. `[2]` `EMPTY` Set if the TX buffer is empty. `[7:3]` — *Reserved.*
`10`	W	`BIST`	Controls whether the transmitter is in built-in self-test (BIST) mode: writing *`01h`* enables the mode, while writing *`00h`* disables it. All but the least-significant bit should be reset to zero.[121]
`11`	R	`ERRNO`	Provides an error code describing the last error indicated on the Wishbone interface.

The CU maintains an internal transmit (TX) buffer. The buffer is used both to hold data until it can be transmitted and to cross a clock domain between the Wishbone interface and the transmitter, allowing BMC transmission at a rate different from that at which the buffer is populated. This is necessary owing to the low 300kbps data rate USB-PD specifies. As this allows the buffer to fill faster than it can be emptied, the *`STATUS`* register is provided for flow control. It is suggested that the buffer be written to until the *`STATUS.FULL`* flag is asserted and then that writing continues once the *`STATUS.FILLING`* flag is deasserted.

A transmission is initiated by a write to either *`KWRITE`* or *`DWRITE`* and continues

[121] See BS EN IEC 62680-1-2 [1] at § 5.9 and 6.4.3 for information on BIST.

until the TX buffer is empty (indicated by *STATUS.EMPTY*). A write after *EMPTY* is asserted will begin a new transmission.

The indication of an error does not halt a transmission. A Wishbone interaction the result of which is an error is treated as if it had never occurred. Table B3 lists the error codes which are defined for the CU. If no error has been raised, the value of the *ERRNO* register is unspecified.

Table B3. Error codes for the BMC transmitter control unit.

Code	Description
00h	**Bus error**; the control unit received a request which did not comply with the rules for the Wishbone bus.
01h	**Unrecognised register**; the address specified in the Wishbone request does not correspond to a known register.
02h	**Operation not supported**; an attempt was made to read from or write to a register which does not support that operation.
80h	**Invalid K-code**; the value written to *KWRITE* is not a known K-code.
81h	**TX buffer overflow**; an attempt was made to write to the TX buffer while it was full (*STATUS.FULL* asserted).

It is considered that a common system of error codes may be desirable, and so generic codes are assigned from *00h* here and CU-specific codes from *80h*.

In BIST mode, the transmitter will produce a continuous stream of biphase mark-coded *0* and *1* until disabled. Writes to *KWRITE* or *DWRITE* in this time will fail and error code *02h* (operation not supported) will be indicated.

Output is generated on the *LD_DAT_O* and *LD_EN_O* ports, which are intended for direct connection to a tristate buffer. The latter, when high, indicates that the buffer should be enabled and driving the line. The former provides the value that the buffer should drive. When not enabled, the value on *LD_DAT_O* is unspecified.

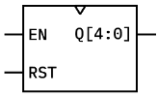
Asserting *WB_RST_I*, which is synchronous to *WB_CLK*, will return the control unit to its initial state. To ensure correct reset, the signal must be asserted for at least three *DCLK* cycles. This ensures that enough time has elapsed for the signal to cross from the Wishbone clock domain to the line driver clock domain. Output on *LD_DAT_O* and *LD_EN_O* is not specified for a reset during a transmission.

Note that the control unit does not generate and append check codes or ‘EOP’ K-codes. It is the responsibility of the consumer to ensure that these are present when USB-PD requires.

B2.6 5-bit Gray code generator

```
entity GrayGenerator5b is
port(
      CLK    : in   std_logic;
      EN     : in   std_logic;
      RST    : in   std_logic;

      Q      : out  std_ulogic_vector(4 downto 0)
      );
end GrayGenerator5b;
```



B2.6.1 Signals

CLK
: The generator clock.

EN
: Enable; when high, indicates that the generator should produce the next code in the sequence.

RST
: Reset; returns the generator to the start of its sequence.

Q[4:0]
: Output; the five-bit Gray code.

B2.6.2 Functional description

The 5-bit Gray code generator produces a Gray code—that is, a sequence where the difference between one code is a change in one bit from the previous code.

If *EN* is high on a rising edge of *CLK*, the value on *Q* on the next rising edge will change to the subsequent code in the sequence. The precise sequence is not specified, but no two adjacent items differ by more than one bit and the sequence does not repeat before at least 32 codes have been produced.

The produced codes, in addition to being Gray codes, exhibit the following properties: first, the most-significant bit of code is *0* for the first 16 codes and *1* for the last 16; and second, the exclusive-or of the two most-significant bits alternates every eight codes, beginning at *0*. These properties ease the implementation of the FIFO9 *FFREG* architecture (see B2.7). The latter ensures that the 5-bit Gray code can easily be transformed to a repeating sequence of 4-bit addresses, while the former requirement means that whether the addresses have ‘wrapped’ back to the start is encoded in a cross-clock-domain-safe form.

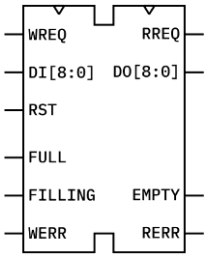
The *RST* signal, if asserted on a rising edge of *CLK*, returns the generator to the first code in the sequence.

B2.7 FIFO9 block

```
entity FIFO9 is
port(
    WRCLK    : in   std_logic;
    WREQ     : in   std_logic;
    DI       : in   std_logic_vector(8 downto 0);
    FULL     : out  std_ulogic;
    FILLING  : out  std_ulogic;
    WERR     : out  std_ulogic;

    RDCLK    : in   std_logic;
    RREQ     : in   std_logic;
    DO       : out  std_ulogic_vector(8 downto 0);
    EMPTY    : out  std_ulogic;
    RERR     : out  std_ulogic;

    RST      : in   std_logic
    );
end FIFO9;
```



B2.7.1 Signals

WRCLK
: Write clock; used to synchronise writes to the FIFO.

WREQ
: Write request; indicates that data is present to be written on *DI*.

DI[8:0]
: Data input; a 9-bit data port used for writing data to the FIFO.

RDCLK
: Read clock; used to synchronise reads from the FIFO.

RREQ
: Read request; signals for data to be loaded from the FIFO onto *DO*.

RST
: Reset; empties the FIFO and returns it to its initial state.

FULL
: Indicates that the FIFO has no more space free to store data.

FILLING
: Indicates that half or more of the FIFO's storage capacity is used.

WERR
: Asserted when an attempt to write to the FIFO is made while it is full.

DO[8:0]
: Data output; a 9-bit data port used to convey data from the FIFO.

EMPTY
: Indicates that the FIFO does not contain any data.

RERR
: Asserted when a read from the FIFO is attempted while it is empty.

B2.7.2 Functional description

The FIFO9 (first-in first-out, 9 bits) block is a dual-clocked memory element where data written to the element is read out sequentially in write order. Write and read operations can be synchronised to clocks which are unrelated in both frequency and phase, and so separate *WRCLK* and *RDCLK* inputs are provided.

Two FIFO9 architectures are provided: *XBRAM*, which is a simple wrapper over a hardened FIFO block RAM primitive on Xilinx 7-series FPGAs; and *FFREG*, which implements a small memory using logic elements from the FPGA fabric and with a design derived from that described by Cummings [88]. It is intended that the latter will be used in testing and simulation and that the former can be used in synthesised designs where a larger buffer is required.

Operation is straightforward. When *WREQ* is asserted on a rising edge of *WRCLK*, the data on *DI* is written into the FIFO. If the FIFO is full and *WREQ* is asserted, the *WERR* signal will be asserted until either *WREQ* is released or the space becomes available in the FIFO. Read operation is similar: when *RREQ* is asserted on a rising edge of *RDCLK*, data is read from the FIFO onto *DO*. If the FIFO is empty and *RREQ* is asserted, *RERR* will be asserted either until *RREQ* is released or until the FIFO is no longer empty. *WERR* and *RERR* change on *WRCLK* and *RDCLK*, respectively.

Three status signals—*FULL*, *FILLING*, and *EMPTY*—are provided to allow users to effectively moderate data flow. The former two are synchronised to *WRCLK* and the latter to *RDCLK*. The FIFO asserts *FULL* when no free space remains for data and *FILLING* when half or more of the space for data is in use. It asserts *EMPTY* when no data remains in the FIFO. Assertion of the status signals is guaranteed to prevent over- or underflow. Status signal deassertion is guaranteed to occur after the assertion condition is removed, but the time between removal and deassertion is not specified.

The *RST* signal is used to clear the FIFO of its contents. The time for a reset to take effect is unspecified, but a reset will cause assertion of *EMPTY* once it has completed. A user must rely on this signal to determine completion. *RST* is synchronised to *WRCLK*.

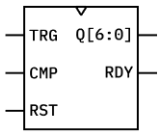
FIFO capacity is not specified and is architecture-dependent.

B3. Biphase Mark Code Receiver

B3.1 Binary search block

```
entity BinarySearcher is
port(
     CLK    : in  std_logic;
     TRG    : in  std_logic;
     CMP    : in  std_logic;
     RST    : in  std_logic;

     Q      : out std_ulogic_vector(6 downto 0);
     RDY    : out std_ulogic
     );
end BinarySearcher;
```



B3.1.1 Signals

CLK
: The master clock.

TRG
: Trigger; indicates that the block should process its inputs.

CMP
: Comparator; indicates whether the output on *Q* is too high or too low.

RST
: Reset; returns the search block to its idle state until released.

Q[6:0]
: Output; a 7-bit output over which a binary search is performed.

B3.1.2 Functional description

The binary search block searches through a 7-bit space for a value equal to that at its input. Its intended application is in the frequency refinement portion of the BMC receiver, as described in 5.3.1.

The binary search block determines the value of each bit of *Q* in sequence, starting at the most significant bit. It begins with the most-significant bit set and, on rising edges of *CLK* where *TRG* is asserted, evaluates the state of *CMP*. If *CMP* is low, indicating that *Q* is less than the target value, the block moves onto the next most significant bit. Alternatively, if *CMP* is high (indicating *Q* is greater than the target value), the block resets the current bit to zero and moves to the next most significant. This process repeats for all bits except the least-significant bit where,

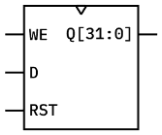
instead of proceeding to the next bit, the block asserts *RDY* and waits in an idle state until *RST* is asserted.

The assertion of *RST* returns the block to its initial state. Once *RST* is deasserted, subsequent assertions of *TRG* will cause the block to proceed from the most-significant bit of *Q*.

B3.2 Cyclic redundancy check engine

```
entity PDCRCEngine is
port(
     CLK    : in  std_logic;
     WE     : in  std_logic;
     D      : in  std_logic;
     RST    : in  std_logic;

     Q      : out std_ulogic_vector(31 downto 0)
     );
end PDCRCEngine;
```



B3.2.1 Signals

CLK
: The data clock.

WE
: Write enable; indicates that data is available and should be processed.

D
: Data input; the source of data to feed through check code generation.

RST
: Reset; returns internal state to its specified initial value.

Q[31:0]
: Output; a 32-bit check code for all the data passed through the engine.

B3.2.2 Functional description

The cyclic redundancy check (CRC) engine provides means to generate, in a streaming manner, a check code of the kind BS EN IEC 62680-1-2 [9] requires be included with USB-PD data messages (see *ibid.* at § 5.6.2).

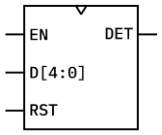
The CRC engine can be imagined as a 32-bit shift register. On rising edges of *CLK* where *WE* is asserted, the CRC engine shifts in the value on *D*, which causes it and previous data to propagate through the register. Logic between each bit position in the shift register causes the CRC polynomial *04C11DB7h* to be applied. The value of *Q* represents the check code computed for all data provided to the engine and updates one clock cycle after data has been clocked in. Compared to the internal shift register, *Q* is both inverted and reversed in bit order. This is needed to meet the requirements of USB-PD.

Asserting *RST* asynchronously resets the shift register to *FFFFFFFFh*, the initial state required by USB-PD.

B3.3 End-of-packet detector

```
entity PDEOPDetector is
port(
     CLK    : in  std_logic;
     EN     : in  std_logic;
     D      : in  std_logic_vector(4 downto 0);
     RST    : in  std_logic;

     DET    : out std_ulogic
     );
end PDEOPDetector;
```



B3.3.1 Signals

CLK
: The master clock.

EN
: Enable; indicates that a line symbol is present on the data input.

D[4:0]
: Data input; a 5-bit line symbol to be processed by the detector.

RST
: Reset; returns the detector to its initial state.

DET
: Asserted when an end-of-packet condition is detected.

B3.3.2 Functional description

The end-of-packet detector processes 5-bit line symbols used in USB-PD to determine whether an end-of-packet condition has occurred. It is intended to enable the receiver to determine when to stop listening for data.

On each rising edge of *CLK* where *EN* is asserted, the detector evaluates the 5-bit line symbol presented on *D*. If it detects an end-of-packet condition, *DET* is asserted on the next rising edge of *CLK*.

Three end-of-packet conditions are recognised:

- After a reset, the first four symbols are K-codes for the ‘Hard_Reset’ ordered set (*RST-1 RST-1 RST-1 RST-2*);
- After a reset, the first four symbols are K-codes for the ‘Cable_Reset’ ordered set (*RST-1 Sync-1 RST-1 Sync-3*);
- At any time, the symbol presented is the ‘EOP’ K-code.

Figure B5 describes the operation of the detector.

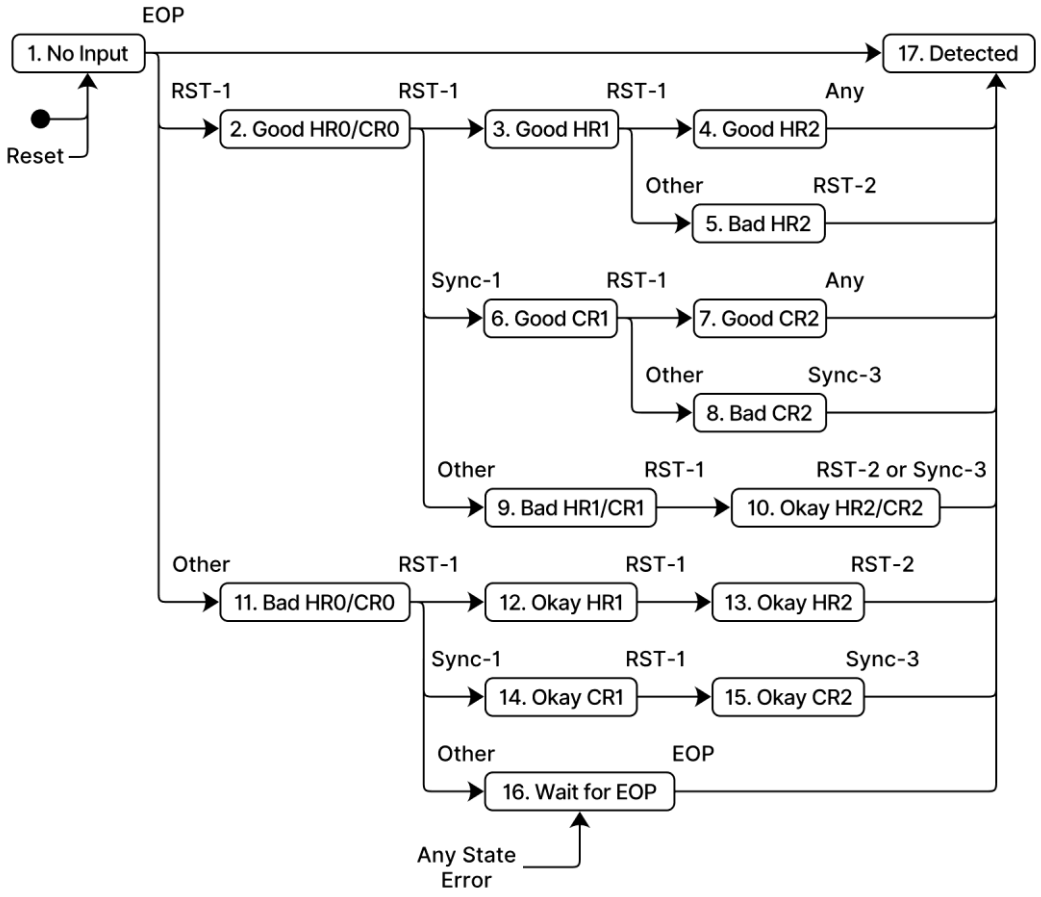


Figure B5. State diagram for the end-of-packet detector.

In that figure, ‘HR’ represents ‘Hard_Reset’ and ‘CR’ represents ‘Cable_Reset’, with transition conditions being evaluated when *EN* is asserted. The conditions given are the line symbols presented on *D*. If any symbol other than one listed in a transition condition is presented, a state error occurs.

The first two conditions will be met if any three of the four K-codes which make up the given ordered sets are identified. This is consistent with the permission in BS EN IEC 62680-1-2 [9] at § 5.4.

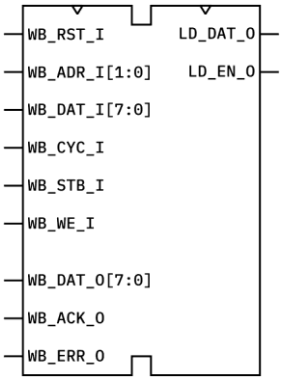
Once *DET* is asserted, it will remain asserted until the detector is reset. Further data presented to the detector will not influence its state until it is reset.

Asserting *RST* asynchronously resets the detector and returns its state to the initial configuration.

B3.4 Control unit

```
entity BiphaseMarkReceiver is
port(
   WB_CLK    : in   std_logic;
   WB_RST_I  : in   std_logic;
   WB_ADR_I  : in   std_logic_vector(1 downto 0);
   WB_DAT_I  : in   std_logic_vector(7 downto 0);
   WB_CYC_I  : in   std_logic;
   WB_STB_I  : in   std_logic;
   WB_WE_I   : in   std_logic;
   RXIN      : in   std_logic;

   WB_DAT_O  : out  std_ulogic_vector(7 downto 0);
   WB_ACK_O  : out  std_ulogic;
   WB_ERR_O  : out  std_ulogic;
   );
end BiphaseMarkReceiver;
```



B3.4.1 Signals

WB_CLK
: The clock used by the Wishbone bus interface (*i.e.* *CLK_I*) and the master clock for internal logic.

WB_RST_I
: The Wishbone reset signal.

WB_ADR_I[1:0]
: The Wishbone address bus.

WB_DAT_I[7:0]
: The Wishbone data-in bus.

WB_CYC_I
: The Wishbone cycle-active signal.

WB_STB_I
: The Wishbone strobe (select) signal.

WB_WE_I
: The Wishbone write-enable signal.

RXIN
: The incoming signal from an external transmitter.

WB_DAT_O[7:0]
: The Wishbone data-out bus.

WB_ACK_O
: The Wishbone slave acknowledge signal.

WB_ERR_O
: The Wishbone slave error signal.

B3.4.2 Functional description

The control unit (CU) for the BMC receiver provides the logic required to decode an incoming USB-PD transmission from an asynchronous remote transmitter. It also exposes a Wishbone Classic interface to enable connection to a processing element. The interface supports ‘SINGLE’ and ‘BLOCK’ read operations with an 8-bit data port and 2-bit addresses.

Table B4 describes the registers the CU exposes to the Wishbone bus.

Table B4. Register map for the BMC receiver control unit.

Address	Action	Name	Description
00	R	RXQ	The value provided by this register is the value at the front of the receive queue. Values are qualified by the state of *TYPE*.
01	R	TYPE	Indicates the type of data present in *RX*. 00h — No data present (value unspecified). 01h — Raw binary data. 02h — *Reserved.* 03h — USB-PD K-code. — — *Reserved.*
10	R	ERRNO	Provides an error code describing the last error indicated on the Wishbone interface.
11	—	—	*Reserved.*

Internally, the CU maintains a receive (RX) buffer which holds data decoded from the line. Considered as a queue or FIFO, the *RXQ* register exposes the value at the front of the queue. Reading from *RXQ* prompts the CU to load the next value. As a USB-PD transmission may include both data and in-band signalling, the *TYPE* register is provided to identify the function of the current value in *RXQ*. Reading from *TYPE* does not prompt the loading of new data, and so *TYPE* should be read first before *RXQ*. The *TYPE* register also provides status information by indicating when no valid data is present in *RXQ* (*i.e.* when the RX buffer is empty).

If *TYPE* indicates that raw binary data is present, *RXQ* contains an 8-bit little-endian value. If *TYPE* indicates a K-code is present, *RXQ* is a K-code number given in B2.3.2. When *TYPE* indicates that no data is present, *RXQ* is not specified.

The *ERRNO* register provides a code indicating the last reason that the Wishbone error signal was asserted. Its value is not specified if the error signal has not yet been asserted. Table B5 provides error codes specific to the CU, but the common error codes given in B2.5.2 (that is, those with a value less than *80h*) will also be

reported as appropriate.

Table B5. Error codes for the BMC receiver control unit.

Code	Description
80h	**Invalid line symbol**; a sequence was received which corresponded to a line symbol that BS EN IEC 62680-1-2 [9] requires not be used.
81h	**RX buffer overflow**; the receiver attempted to store a decoded symbol while the receive buffer was full.
82h	**Receive timeout**; a pulse lasted longer than anticipated, or the line returned to BMC idle[122] before a complete packet could be received.
83h	**CRC failure**; the receiver detected an invalid check code.

Error codes specific to the receiver CU are only generated in response to reads from *RXQ* when an error condition exists. The CU will continue to assert a given error until it detects the BMC idle condition. While an error of this kind is asserted, *TYPE* will report that no data is present in *RXQ*. Once the error is no longer being asserted, the RX queue will be empty.

The receiver determines it has validly reached the end of a packet if it detects either the ‘Hard_Reset’ or ‘Cable_Reset’ ordered sets (see 2.2.2) immediately after the preamble, or if it detects an ‘EOP’ K-code. The receiver does not verify the correctness of packets and so, for example, a packet consisting of a preamble followed immediately by data and the ‘EOP’ K-code may not result in an error.

All K-codes, including ‘EOP’, are retained in the RX buffer. The only element of a packet not stored by the receiver is the preamble.

The receiver does not filter received packets by addressee. A consumer must identify whether a start-of-packet ordered set is present and, if it is, whether it is the kind heeded by the consumer.

[122] See BS EN IEC 62680-1-2 [9] at § 5.8.6.1.

References

[1] H. Reydams, V. Lauwereys, D. Haeseldonckx, P. van Willigenburg, J. Woudstra, and S. De Jonge, "The development of a proof of concept for a smart DC/DC power plug based on USB Power Delivery," presented at the 22nd Domestic Use of Energy, Cape Town, South Africa, April 1–2, 2014, doi:10.1109/DUE.2014.6827761.

[2] V. Vossos, D. Gerber, Y. Bennani, R. Brown, and C. Marnay, "Techno-economic analysis of DC power distribution in commercial buildings," *Applied Energy*, vol. 230, pp. 663–678, Nov. 2015, doi:10.1016/j.apenergy.2018.08.069.

[3] *Boxes for flush mounting of electrical accessories — Requirements, test methods and dimensions*, BS 4662:2006+A1:2009, 2009.

[4] *FUSB302 Programmable USB Type-C Controller w/PD*, ON Semiconductor, July 2017. [Online]. Available: *onsemi.com/pub/Collateral/FUSB302-D.PDF*

[5] *PTN5110 — USB PD TCPC PHY IC*, NXP Semiconductors, Jan. 2018. [Online]. Available: *nxp.com/docs/en/data-sheet/PTN5110_DS.pdf*

[6] *UPD350 — Highly Integrated Small Form Factor USB Type-C™ Power Delivery 3.0 Port Controller*, Microchip Technologies, June 2018. [Online]. Available: *ww1.microchip.com/downloads/en/DeviceDoc/UPD350-Data-Sheet-DS00002643B.pdf*

[7] *AP43770 — High-Performance USB PD Controller*, Diodes, Inc., April 2019. [Online]. Available: *diodes.com/assets/Datasheets/AP43770.pdf*

[8] *TPS65988 Dual Port USB Type-C and USB PD Controller with Integrated Power Switches*, Texas Instruments, Aug. 2018. [Online]. Available: *ti.com/lit/ds/symlink/tps65988.pdf*

[9] *Universal Serial Bus interfaces for data and power — Part 1-2: Common components — USB Power Delivery Specification*, BS EN IEC 62680-1-2:2018, 2018.

[10] *Universal Serial Bus Power Delivery Specification — Revision 2.0, Version 1.3*, USB Implementers Forum, 2017.

[11] *Digital audio interface — Part 1: General*, BS EN 60958-1:2008+A1:2014, 2014.

[12] *Universal Serial Bus interfaces for data and power — Part 1-3: Common components — USB Type-C Cable and Connector Specification*, BS EN IEC 62680-1-3:2018, 2018.

[13] *Battery Charging Specification — Revision 1.2*, USB Implementers Forum, 2010.

[14] *74AUP1G125 — Low-power buffer/line driver; 3-state*, Nexperia, May 2018. [Online]. Available: assets.nexperia.com/documents/data-sheet/74AUP1G126.pdf

[15] *VT6M1 — 20V Nch+Pch Small Signal MOSFET*, ROHM, July 2019. [Online]. Available: rohm.com/datasheet/VT6M1/vt6m1t2cr-e

[16] *NTZD3155C — MOSFET – Small Signal, Complementary with ESD Protection, SOT-563*, ON Semiconductor, June 2019. [Online]. Available: onsemi.com/pub/Collateral/NTZD3155C-D.PDF

[17] R. McArthur, "Making Use of Gate Charge Information in MOSFET and IGBT Data Sheets," Advanced Power Technology, App. Note APT0103, 2001.

[18] P. Horowitz and W. Hill, *The Art of Electronics*, 3rd ed. New York, NY, USA: Cambridge University Press, 2015.

[19] *74AHC1G66; 74AHCT1G66 — Single-pole single-throw analog switch*, Nexperia, Dec. 2008. [Online]. Available: assets.nexperia.com/documents/data-sheet/74AHC_AHCT1G66.pdf

[20] R. W. Adams, "SPDIF-AES/EBU digital audio data recovery," US Patent 5 889 820, March 1999.

[21] "Hysteresis Setting for Comparator," ROHM, App. Note, 2017. [Online]. Available: rohmfs.rohm.com/en/products/databook/applinote/ic/amp_linear/comparator/gpl_cmp_hysteresis-e.pdf

[22] *DACx3401 10-Bit and 8-Bit, Voltage-Output Digital-to-Analog Converters with Nonvolatile Memory and PMBus™ Compatible I²C Interface in Tiny 2×2 WSON*, Texas Instruments, Dec. 2019. [Online]. Available: ti.com/lit/ds/symlink/dac43401.pdf

[23] *MAX5187/MAX5190 — 8-Bit, 40MHz, Current/Voltage-Output DACs*, Maxim Integrated, Dec. 2001. [Online]. Available: datasheets.maximintegrated.com/en/ds/MAX5187-MAX5190.pdf

[24] *BB145B — Low-voltage variable capacitance diode*, NXP Semiconductors, March 2004. [Online]. Available: nxp.com/docs/en/data-sheet/BB145B.pdf

[25] B. Nisarga, “PWM DAC Using MSP430 High-Resolution Timer,” Texas Instruments, App. Rep. SLAA497, 2011. [Online]. Available: *ti.com/lit/an/slaa497/slaa497.pdf*

[26] A. Winterer, “Low cost system for FSK demodulation,” US Patent 5 703 525, Dec. 1997.

[27] *CD54HC4046A, CD74HC4046A, CD54HCT4046A, CD74HCT4046A High-Speed CMOS Logic Phase-Locked Loop with VCO*, Texas Instruments, Dec. 2003. [Online]. Available: *ti.com/lit/ds/symlink/cd74hc4046a.pdf*

[28] R. Sobot, *Wireless Communication Electronics*, New York, NY, USA: Springer, 2012.

[29] *LM339, LM239, LM139, LM2901 Quad Differential Comparators*, Texas Instruments, Nov. 2018. [Online]. Available: *ti.com/lit/ds/symlink/lm139a.pdf*

[30] *SN54HC148, SN74HC148 — 8-line to 3-line Priority Encoders*, Texas Instruments, April 2004. [Online]. Available: *ti.com/lit/ds/symlink/sn74hc148.pdf*

[31] *AN431 — Adjustable precision shunt regulators*, Diodes, Inc., April 2015. [Online]. Available: *diodes.com/assets/Datasheets/AN431.pdf*

[32] *ADC081C021/ADC081C027 I²C-Compatible, 8-Bit Analog-to-Digital Converter with Alert Function*, Texas Instruments, March 2013. [Online]. Available: *ti.com/lit/ds/symlink/adc081c021.pdf*

[33] J. G. Bralla, *Design for Excellence*, New York, NY, USA: McGraw-Hill, 1996.

[34] *Generic Standard on Printed Board Design*, IPC-2221A, 2003.

[35] *Sectional Design Standard for Rigid Organic Printed Boards*, ANSI/IPC-2222, 1998.

[36] *Audio/video, information and communication technology equipment — Part 1: Safety requirements*, BS EN 62368-1:2014+A11:2017, 2017.

[37] *Functional safety of electrical/electronic/programmable electronic safety-related systems — Part 1: General requirements*, BS EN 61508-1:2010, 2010.

[38] *Safety of power transformers, power supplies, reactors and similar products — Part 1: General requirements and tests*, BS EN 61558-1:2005+A1:2009, 2009.

[39] S. Freeman and N. Price, *Growing Object-Oriented Software, Guided by Tests*, Boston, MA, USA: Addison-Wesley Professional, 2009.

[40] C. Maxfield, *The Design Warrior's Guide to FPGAs*, Burlington, MA, USA: Newnes, 2004.

[41] E. Seligman, T. Schubert, and M. V. Achutha Kiran Kumar, *Formal Verification: An Essential Toolkit for Modern VLSI Design*, Waltham, MA, USA: Morgan Kaufmann, 2015.

[42] *SymbiYosys*. (2020), C. Wolf. [Online]. Available: `github.com/YosysHQ/SymbiYosys`

[43] *ghdlsynth-beta*. (2020), T. Gingold. [Online]. Available: `github.com/tgingold/ghdlsynth-beta`

[44] *Drawing No. 10137061 — USB 3.1 Type C Receptacle, SMT and DIP Tails*, Amphenol ICC, April 2017. [Online]. Available: `cdn.amphenol-icc.com/media/wysiwyg/files/drawing/10137061.pdf`

[45] N. Mehta, "Xilinx 7 Series FPGAs: The Logical Advantage," Xilinx, White Paper WP 405, 2012. [Online]. Available: `xilinx.com/support/documentation/white_papers/wp405-7Series-Logical-Advantage.pdf`

[46] *iCE40 UltraLite Family Data Sheet*, Lattice Semiconductor, Aug. 2018.

[47] *PCA9570 — Remote 4-bit general purpose outputs for 1 MHz I²C bus*, NXP Semiconductors, Nov. 2017. [Online]. Available: `nxp.com/docs/en/data-sheet/PCA9570.pdf`

[48] *UK-SPEC — UK Standard for Professional Engineering Competence*, 3rd ed., London, United Kingdom: Engineering Council, 2014.

[49] *Statement of Ethical Principles*, 2nd ed., London, United Kingdom: Engineering Council and the Royal Academy of Engineering, 2017.

[50] J. Ziller, "Thunderbolt 3 — The USB-C That Does It All," Intel Corporation, 2015. [Online]. Available: `thunderbolttechnology.net/blog/thunderbolt-3-usb-c-does-it-all`

[51] "HDMI Releases Alternate Mode for USB Type-C™ Connector," HDMI Licensing, LLC, 2016. [Online]. Available: `hdmi.org/download/pressFileId/58`

[52] *Universal Serial Bus Specification — Revision 1.1*, Intel Corporation, Microsoft, NEC, 1998.

[53] *Universal Serial Bus Specification — Revision 2.0*, Compaq, Hewlett-Packard, Intel Corporation, Lucent Technologies, Microsoft, NEC, Koninklijke Philips Electronics, 2000.

[54] *Preferred number series for resistors and capacitors*, BS EN 60063:2015, 2015.

[55] *MAX220–MAX249 — +5V-Powered, Multichannel RS-232 Drivers/Receivers*, Maxim Integrated, May 2019. [Online]. Available: `datasheets.maximintegrated.com/en/ds/MAX220-MAX249.pdf`

[56] "Section 19. UART," in *dsPIC30F Family Reference Manual*, Microchip Technology, 2006.

[57] "16. USART — Universal Synchronous Asynchronous Receiver/Transmitter," in *EFM32G Reference Manual*, Silicon Labs, 2017.

[58] A. Goldsmith, *Wireless Communications*, New York, NY, USA: Cambridge University Press, 2005.

[59] G. Gonzalez, *Foundations of Oscillator Circuit Design*, Norwood, MA, USA: Artech House, 2007.

[60] *MMBFJ309L, MMBFJ310L, SMMBFJ309L, SMMBFJ310L — JFET – VHF/UHF Amplifier Transistor, N-channel*, ON Semiconductor, Oct. 2016. [Online]. Available: `onsemi.com/pub/Collateral/MMBFJ309LT1-D.PDF`

[61] D. A. Neamen, *Microelectronics: Circuit Analysis and Design*, 4th ed., New York, NY, USA: McGraw-Hill, 2010.

[62] M. K. Kazimierczuk and D. Murthy-Bellur, "Loop Gain of the Common-Drain Colpitts Oscillator," *Int. J. of Electron. and Telecommun.*, vol. 56, no. 4, pp. 423–426, 2010, doi: 10.2478/v10177-010-0057-5.

[63] *TPS251xx USB Dedicated Charging Port Controller*, Texas Instruments, Dec. 2016. [Online]. Available: `ti.com/lit/ds/symlink/tps2513a.pdf`

[64] "Chapter 22. Enhanced Universal Serial Communication Interface (eUSCI) – UART Mode," in *MSP430FR4xx and MSP430FR2xx Family User's Guide*, Texas Instruments, 2019.

[65] D. E. Knuth, *The Art of Computer Programming*, 2nd ed., vol. 3. Boston, MA, USA: Addison–Wesley, 1998.

[66] D. E. Knuth, *The Art of Computer Programming*, 3rd ed., vol. 1. Boston, MA, USA: Addison–Wesley, 1997.

[67] “Commission communication in the framework of the implementation of Directive 2014/35/EU of the European Parliament and of the Council on the harmonisation of the laws of the Member States relating to the making available on the market of electrical equipment designed for use within certain voltage limits (Publication of titles and references of harmonised standards under Union harmonisation legislation),” *Official Journal of the European Union*, vol. 61, C 326, pp. 4–93, Sept. 14, 2018.

[68] “Directive 2014/35/EU of the European Parliament and of the Council of 26 February 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of electrical equipment designed for use within certain voltage limits,” *Official Journal of the European Union*, vol. 57, L 96, pp. 357–374, Feb. 26, 2014.

[69] R. Herveille, *WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores*, rev. B4, Amsterdam, NL: OpenCores, 2010.

[70] L. Pezzarossa, A. T. Kristensen, M. Schoeberl, and J. Sparsø, “Using dynamic partial reconfiguration of FPGAs in real-time systems,” *Microprocessors and Microsystems*, vol. 61, pp. 198–206, 2018.

[71] “Lattice Mico8 Open, Free Soft Microcontroller.” Lattice Semiconductor. Accessed March 2, 2020. [Online]. Available: `latticesemi.com/Products/DesignSoftwareAndIP/IntellectualProperty/IPCore/IPCores02/Mico8.aspx`

[72] *Fixed resistors for use in electronic equipment — Part 8: Sectional specification — Fixed surface mount resistors*, BS EN 60115-8:2012, 2012.

[73] *Safety requirements for power electronic converter systems and equipment — Part 1: General*, BS EN 62477-1:2012+A1:2017, 2017.

[74] *MIMXRT1011DAE5A — i.MX RT1010 Crossover Processors Data Sheet for Consumer Products*, NXP Semiconductors, Sept. 2019. [Online]. Available: `nxp.com/docs/en/data-sheet/IMXRT1010CEC.pdf`

[75] *Icicle*. (2019), G. Edgecombe. [Online]. Available: `github.com/grahamedgecombe/icicle`

[76] M. Fortunato, “Temperature and Voltage Variation Ceramic Capacitor,” Maxim Integrated, App. Note 5527, 2012. [Online]. Available: `maximintegrated.com/en/design/technical-documents/tutorials/5/5527.html`

[77] *MIC920 — 80 MHz Low-Power SC70 Op Amp*, Microchip Technology, Oct. 2019. [Online]. Available: *ww1.microchip.com/downloads/en/DeviceDoc/MIC920-80MHz-Low-Power-SC70-Op-Amp-DS20006268A.pdf*

[78] *MC74VHC1GT66 — SPST (NO) Normally Open Analog Switch*, ON Semiconductor, Aug. 2011. [Online]. Available: *onsemi.com/pub/Collateral/MC74VHC1GT66-D.PDF*

[79] *CAT/CAY 16 Series — Chip Resistor Arrays*, Bourns, Inc., Sept. 2019. [Online]. Available: *bourns.com/docs/Product-Datasheets/CATCAY.pdf*

[80] *TL08xx JFET-Input Operational Amplifiers*, Texas Instruments, May 2015. [Online]. Available: *ti.com/lit/ds/symlink/tl081a.pdf*

[81] B. Carter, "Op Amp and Comparators — Don't Confuse Them!," Texas Instruments, App. Rep. SLOA067, 2001.

[82] "A Colpitts VCO for Wideband (0.95–2.15 GHz) Set-Top TV Tuner Applications," Skyworks Solutions, App. Note 1006, 2003. [Online]. Available: *skyworksinc.com/-/media/SkyWorks/Documents/Products/1-100/200316A.pdf*

[83] *BB844 — Silicon Variable Capacitance Diode*, Infineon Technologies, June 2011. [Online]. Available: *infineon.com/dgdl/Infineon-BB844SERIES-DS-v01_01-en.pdf?fileId=db3a30431be39b97011be4a058c80036*

[84] *BAT54GW — 30 V, 200 mA Schottky barrier diodes*, Nexperia, Nov. 2016. [Online]. Available: *assets.nexperia.com/documents/data-sheet/BAT54GW.pdf*

[85] *74LVC1G17-Q100 — Single Schmitt trigger buffer*, Nexperia, Jan. 2019. [Online]. Available: *assets.nexperia.com/documents/data-sheet/74LVC1G17_Q100.pdf*

[86] *TPS56220x 4.5-V to 17-V Input, 2-A Synchronous Step-Down Voltage Regulator in 6-Pin SOT-23*, Texas Instruments, April 2017. [Online]. Available: *ti.com/lit/ds/symlink/tps562201.pdf*

[87] *TPS63700 DC-DC Inverter*, Texas Instruments, Oct. 2015. [Online]. Available: *ti.com/lit/ds/symlink/tps63700.pdf*

[88] C. E. Cummings, "Simulation and synthesis techniques for asynchronous FIFO design," presented at SNUG San Jose 2002, San Jose, CA, USA, 2002. [Online]. Available: *sunburst-design.com/papers/CummingsSNUG2002SJ_FIFO1.pdf*