# 5 Gb/s Burst-Mode Clock Phase Aligner for Gigabit Passive Optical Networks

*Ming Zeng*

Department of Electrical & Computer Engineering
McGill University
Montreal, Canada

May 2009

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master's of Engineering.

## Abstract

Fiber to the Home (FTTH) has been proven to be an efficient medium for voice, audio, and video transfer. Passive Optical Networks (PONs) are being studied as an upgradeable and low-cost solution to the problem of limited bandwidth in local access networks in the medium of FTTH. In PONs, multiple users share the fiber infrastructure in a point-to-multipoint (P2MP) network topology. The P2MP nature of networks cause the data packets from each user to undergo different amplitude, phase, and frequency variations – resulting in burst-mode traffic at the receiving end of the network. This consequently creates new challenges for the design of optical receivers.

We design and experimentally demonstrate a 5 Gb/s burst-mode clock phase aligner (BM-CPA) featuring automatic phase acquisition with forward-error correction using (64, 57) Hamming codes. This BM-CPA is implemented with commercially available evaluation boards and provides instantaneous (0-bit) phase acquisition with packet loss ratio $< 10^{-6}$ and bit error rate $< 10^{-10}$ for any phase step ($\pm 2\pi$ rads) between consecutive packets. Implementation of a Reed-Solomon(255, 239) code is also investigated. Our design is based on an oversampling algorithm and can be operated in two configurations: BM-CPA with a SONET CDR and BM-CPA with a local oscillator. We conclude by discussing various possible extensions to the device, based on the promising results.

# Résumé

La fibre optique jusqu'au domicile (“fiber-to-the-home” - FTTH) s'avère être un médium efficace pour le transert de la voix, du son et de la vidéo. Les réseaux optiques passifs (“passive optical network” - PON) sont présentement à l'étude pour s'attaquer au problème de la bande passante limitée dans les réseaux locaux sur base de FTTH, puisqu'ils constituent une solution bon marché et peuvent facilement être mise à jour dans le futur. Dans un PON, des utilisateurs multiples se partagent l'infrastructure de fibre optique dans une topologie de réseau point à multipoint (P2MP). La nature P2MP des réseaux fait en sorte que les paquets de données provenant de chaque utilisateur subissent des variations de leur amplitude, de leur phase et de leur fréquence, ce qui crée un trafic par paquet (“burst-mode traffic”) atteignant la portion réceptrice du réseau. Ceci entraîne de nouveaux défis pour la conception de récepteurs optiques.

Nous présentons le design et la démonstration expérimentale d'un circuit pouvant aligner la phase de l'horloge pour les données en paquet (“burst-mode clock phase aligner” - BM-CPA) s'exécutant à une vitesse de 5 Gb/s avec acquisition automatique de la phase et correction d'erreur sans voie de retour (“forward-error correction” - FEC) via des codes Hamming (64,57). L'implémentation de ce BM-CPA est accomplie avec des cartes d'évaluation disponibles sur le marché et permet l'acquisition de phase instantanée avec un taux de paquets perdus $< 10^{-6}$ et un taux de bits erronés $< 10^{-10}$ pour toute phase de $-2\pi$ à $+2\pi$ radians entre paquets consécutifs. L'implémentation d'un code Reed-Solomon (239,255) est aussi étudiée. Notre concept est basé sur un algorithme de suréchantillonage qui peut être opéré de deux façons : un BM-CPA avec un SONET CDR ou un BM-CPA accompagné d'un oscillateur local. Nous concluons l'étude en discutant de possibles voies futures pour la continuation de nos recherches en nous basant sur les résultats déjà très prometteurs.

## Acknowledgments

First and foremost, I would like to thank Professor David Plant for giving me the opportunity to conduct advanced research at the forefront of today's technologies. His constant support and insight have been invaluable to the completion of this Master's Thesis.

Secondly, I would like to thank my lab partners, Bhavin and Nick, for sharing their efforts and expertise towards our common research goals. I also thank them for sharing great times in and outside the lab, be it in Montreal or away on conferences. I also extend my thanks to all the members of the McGill Photonic Systems Group.

I would also like to thank my boyfriend Ben for his constant support through very stressful times, and for his help in the preparation of this thesis. My gratitue also goes to my friends Dan, Jing, Guannan, Jifang, Mittu, Yasha, Dani, and many others with whom I spent such a great time during my graduate studies. Thank you all very much!

Last but not least, I would like to thank my parents and grandparents, first for teaching me the meaning of hard work and the joy of learning, and also for giving me the opportunity to study abroad and see the world. I have the pleasure of finalizing the writing of this thesis from my home in Wuhan, China, in the company of my family and my boyfriend, happy to have all the people I love the most under the same roof as I wrap up the past two years of my life as a graduate student.

Thank you all!

# Contents

**1 Introduction 1**

1.1 Motivation . . . 2

1.2 Access Networks . . . 2

1.3 Optical Networks . . . 3

1.3.1 Passive Optical Networks . . . 4

1.3.2 Gigabit Passive Optical Networks . . . 7

1.4 Research Challenges . . . 9

1.4.1 High Speed Receiver Design Challenges . . . 9

1.4.2 High Speed Burst Mode Receiver Test Bed . . . 9

1.5 Thesis Organization . . . 10

**Bibliography 12**

**2 Review of the State of the Art 14**

2.1 Optical receivers . . . 14

2.1.1 Front-End . . . 14

2.1.2 Clock and Data Recovery . . . 15

2.2 Jitter . . . 17

2.3 Continuous-Mode CDR . . . 19

2.3.1 Continuous-Mode Challenges . . . 19

2.3.2 Continuous-Mode Clock and Data Recovery Performance Optimization 19

2.4 Burst-mode CDR . . . 20

2.4.1 Burst-mode Challenges . . . 20

2.4.2 Burst-mode Clock and Data Recovery . . . 20

2.5 Summary . . . 25

**Bibliography** **26**

**3 Forward Error Correction** **28**

3.1 Error Coding Theory . . . 28

3.2 Forward Error Correction . . . 29

3.2.1 Hamming Codes . . . 30

3.2.2 Single Error Correction Double Error Detection Codes . . . 31

3.2.3 Reed-Solomon Codes . . . 32

3.3 Summary . . . 36

**Bibliography** **37**

**4 Design and Experimental Demonstration of a Burst-Mode Clock Phase Aligner for GPON - Solution 1: Broadband CDR** **38**

4.1 Introduction . . . 38

4.2 Modeling Methodology of a Broadband CDR . . . 41

4.2.1 Basic Linear Continuous-time PLL model . . . 42

4.2.2 Linear Discrete-time Charge-Pump PLL model . . . 44

4.2.3 Nonlinear Discrete-time Charge-Pump PLL model . . . 45

4.3 Simulation Results . . . 47

4.3.1 Basic Linear Continuous-time PLL model Simulation Results . . . 47

4.3.2 Linear Discrete-time Charge-Pump PLL model Simulation Results . 48

4.3.3 Nonlinear Discrete-time Charge-Pump PLL model Simulation Results 49

4.3.4 System Simulation Results . . . 49

4.4 Summary – Potential Problems with Solution 1 . . . 51

**Bibliography** **53**

**5 Design and Experimental Demonstration of a Burst-Mode Clock Phase Aligner for GPON - Solution 2: 2×Oversampling Clock Phase Aligner** **54**

5.1 Introduction - Burst Mode CPA Design Overview . . . 54

5.2 2×Oversampling Algorithm . . . 55

5.3 Local Oscillator vs. SONET CDR . . . 57

5.4 Mode of Operation . . . 57

5.5 Data Deserialization . . . 57
5.6 SEC-DED Decoder Implementation . . . 59
5.7 Hardware Implementation . . . 60
5.8 Burst-Mode Clock Phase Aligner Test Setup . . . 62
5.8.1 Burst-Mode Packet Generator . . . 62
5.8.2 Customly Designed Bit Error Rate and Packet Loss Rate Tester . . . 62
5.8.3 Electrical Test Bed for Burst-Mode Clock Phase Aligner . . . 63
5.8.4 Optical Test Bed for Burst Mode Clock Phase Aligner . . . 64
5.9 Experimental Results . . . 64
5.9.1 Electrical Test Bed Experimental Results . . . 64
5.9.2 Optical Test Bed Experimental Results . . . 66
5.10 Summary . . . 71

**Bibliography** **72**

**6 Summary and Other Solutions for Burst-Mode Clock Phase Aligner** **73**
6.1 Reed-Solomon (255, 239) Implementation . . . 74
6.1.1 Advantages . . . 74
6.1.2 Disadvantages . . . 74
6.2 nXoversampling Implementation, where $n > 2$ . . . 74
6.2.1 Advantages . . . 75
6.2.2 Disadvantages . . . 75
6.3 Clock Tapped Delay Sampling Technique . . . 75
6.3.1 Advantages . . . 75
6.3.2 Disadvantages . . . 76
6.4 ASIC Design . . . 76
6.4.1 Advantages . . . 76
6.4.2 Disadvantages . . . 77
6.5 Summary . . . 77

**A Reed-Solomon Decoder Implementation** **78**
A.1 Reed-Solomon Decoder Implementation . . . 78
A.1.1 Decoding Steps . . . 78
A.1.2 Hardware Implementation . . . 81

**Bibliography** **88**

# List of Figures

1.1 Passive optical network diagram . . . . . . . . . . . . . . . . . . . . . . . 5
1.2 GPON physical networks architecture . . . . . . . . . . . . . . . . . . . . 8

2.1 Data being sampled by a synchronized clock signal at the optimal sampling point . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
2.2 A PLL-based CDR in an optical network . . . . . . . . . . . . . . . . . . 16
2.3 Hogge phase detector and its operational timing diagram . . . . . . . . . . 17
2.4 Correlation algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
2.5 Gated oscillator . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22
2.6 Clock recovery scheme using matched gated oscillators . . . . . . . . . . . 23
2.7 Burst-mode clock and data recovery based on a 2×Oversampling and phase picking algorithm . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

3.1 Composition of a Hamming codeword . . . . . . . . . . . . . . . . . . . . 30
3.2 Example of parity bits generation . . . . . . . . . . . . . . . . . . . . . . 30
3.3 Structure of a Reed-Solomon codeword . . . . . . . . . . . . . . . . . . . 33
3.4 A general architecture of a Reed Solomon $(n, k)$ encoder . . . . . . . . . . 35

4.1 Graphical demonstration of a passive optical network . . . . . . . . . . . 39
4.2 A typical packet in a PON . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.3 CDR performance at various phase steps . . . . . . . . . . . . . . . . . . 41
4.4 A circuit block diagram of a PLL . . . . . . . . . . . . . . . . . . . . . . 42
4.5 Circuit diagram of two low pass filters . . . . . . . . . . . . . . . . . . . 43
4.6 A circuit block diagram of a charge-pump PLL . . . . . . . . . . . . . . . 44
4.7 Timing diagrams of the phase detector . . . . . . . . . . . . . . . . . . . 45
4.8 Timing diagram of case $\tau(k) > 0$, $\tau(k+1) > 0$ . . . . . . . . . . . . . . . . . 46

4.9 Step response of a PLL . . . 48
4.10 Time domain phase step response for control voltage $\nu(k)$ . . . 49
4.11 ADS simulation schematic circuit diagram . . . 50
4.12 Circuit simulation results . . . 51
4.13 Comparison of the effects of a 1st order and a 2nd order loop filters . . . 52

5.1 Block diagram of the BM-CPA . . . 55
5.2 Clock and data phase recovery using a single CDR vs. using the 2×Oversampling algorithm . . . 56
5.3 Circuit layout on a Virtex IV FPGA board. . . . 59
5.4 Block diagram of a (64, 57) SEC-DED decoder . . . 60
5.5 Hardware implementation of the BM-CPA . . . 61
5.6 Electrical test-bed of the BM-CPA . . . 63
5.7 Optical test-bed of the BM-CPA . . . 64
5.8 PLR performance for the BM-CPA in the electrical test bed . . . 65
5.9 BER performance for the BM-CPA in the electrical test bed . . . 66
5.10 BER and PLR performances at different BM-CPA operation modes at the data rate of 1.25 Gb/s . . . 67
5.11 BER and PLR performances of BM-CPA with a local oscillator at different data rates . . . 68
5.12 Burst mode penalty . . . 69
5.13 Comparison of CID immunity of a CDR with the BM-CPA . . . 70
5.14 Data input and output waveforms for dynamic range measurements . . . 70

A.1 The Berlekamp-Massey algorithm . . . 80
A.2 System level block diagram of a RS decoder . . . 82
A.3 Hardware implementation of a syndrome generator . . . 82
A.4 Block diagram of a syndrome calculator block with parallel syndrome generators . . . 83
A.5 Berlekamp-Massey State Diagram . . . 84
A.6 Chien search block diagram . . . 86
A.7 Hardware implementation of error magnitude polynomial computation . . . 87
A.8 Hardware Implementation of Forney Algorithm . . . 87

# List of Tables

1.1 Comparison of different types of PONs . . . . . . . . . . . . . . . . . . . . . 7

2.1 Comparison of four techniques for burst-mode CDR solutions . . . . . . . . . 25

4.1 Optimal PLL design parameters . . . . . . . . . . . . . . . . . . . . . . . . . 47

5.1 Summary of the BM-CPA performance . . . . . . . . . . . . . . . . . . . . . 71

# List of Acronyms

| | |
|---|---|
| ADS | advanced design system |
| ARPANet | advanced researchpProjects agency network |
| ARQ | automatic repeat request |
| ATM | asynchronous transfer mode |
| BER | bit error rate |
| BERT | bit error rate tester |
| BM-CPA | burst-mode clock phase aligner |
| BMR | burst mode receiver |
| BPON | broadband PON |
| CDMA | code division multiple access |
| CDR | clock and data recovery |
| CO | central office |
| DBA | dynamic bandwidth allocation |
| DCM | digital clock manager |
| DDR | double-data rate |
| DSL | digital subscriber line |
| DWDM | dense wavelength division multiplexing |
| EPON | ethernet PON |
| FE | front-end |
| FEC | forward error correction |
| FPGA | field programmable gate array |
| FTTC | fiber-to-the-curb |
| FTTCab | fiber-to-the-cabinet |
| FTTH | fiber-to-the-home |
| FTTP | fiber-to-the-premise |

| | |
|---|---|
| FTTx | fiber-to-the-home/building/curb |
| GEM | GPON encapsulation method |
| GPON | gigabit PON |
| GTC | GPON transmission coverage |
| ITU-T | international telecommunication union - telecommunication standardization sector |
| LA | limiting amplifier |
| LPF | low-pass filter |
| MAC | medium access control |
| OLT | optical line terminal |
| ONT | optical network terminal |
| OOK | on-off keying |
| P2MP | point-to-multipoint |
| P2P | point-to-point |
| PC | power combiner |
| PFD | phase/frequency detector |
| PLL | phase-locked loop |
| PLR | packet lost ratio |
| PMD | physical medium dependent |
| PONS | passive optical networks |
| POTS | plain old telephone service |
| QoS | quality of service |
| RF | radio frequency |
| SCM | subcarrier multiplexing |
| SDH | synchronous digital hierarchy |
| SEC-DED | single-error correction and double-error detection |
| SMF | single-mode fiber |
| SNOET | synchronous optical networks |
| SNR | signal-to-noise ratio |
| TDM | time division multiplexing |
| TIA | transimpedance amplifier |
| VCO | voltage controller oscillator |
| VOA | variable optical attenuators |

# Chapter 1

## Introduction

The history of communication is a significant chapter of human evolution. It dates back to the earliest signs of life and witnesses the development of human intelligence. Human communication was revolutionized with speech about 200,000 years ago. Symbols were developed about 30,000 years ago, and writing about 7,000 years before our time [1]. Telecommunication, on the other hand, has a much shorter history, beginning only a few centuries ago. However, it constitutes perhaps the most fascinating chapter in the history of human communication.

The earliest known means of telecommunication was drums in Africa, which were used to send signals to neighboring tribes and groups. This invention used sound patterns to carry information for long-distance communication. Around 1200 BC, fire signals were used until smoke signals were later deployed. In 1835, Samuel Morse invented the “Sam Signal” which was later renamed to “Morse Code”. It did not find any significant use for 19 years until the first telegraph was sent from Melbourne to Williamstown in Australia. Alexander Graham Bell registered his first telephone in 1876, and started an era of ringtones. In the 1960’s, the US military developed the very first network, Advanced Research Projects Agency Network (ARPANet), to connect their computers. Ethernet came around in 1974 and the World Wide Web (WWW) platform has thrived on the Internet since the 1990’s [2].

## 1.1 Motivation

When the Internet was first used, file transfer and e-mail were the most popular services, which resulted in mostly text-based traffic. However, during the last decade, the graphical nature of the WWW has brought the challenge of “bandwidth” into network design. More recently, the need for bandwidth is growing rapidly as more video portals, such as YouTube, continue with their rapid growth, together with the exponential growth in the number of Internet users. Based on the current bandwidth trends, it can be projected that by 2010 to 2015, the global Internet backbone will have to handle bandwidths higher than 1000 Tb/s [3]. To solve this problem with the least cost, various communication networks are being proposed and designed. How will these communication networks affect our lives? It is a question to be answered in the near feature.

The most promising technology to answer the problem of increasing bandwidth in local access networks is that of optical multiaccess networks for the deployment of fiber-to-the-home/building/curb (FFTx). Passive optical networks (PONs) have attracted a lot of attention in recent years for their potential to offer economical and high-bandwidth data transfer. The challenge in implementing this type of network arises from the point-to-multipoint (P2MP) network topology which requires the design of a receiver to deal with burst data from multiple users. This has been the focus of this research project, and a novel design for a burst-mode receiver is presented in this thesis.

## 1.2 Access Networks

Through almost a century-long evolution, modern telecommunication advanced from a friendly local operator to a network that is transmitting the equivalent of thousands of encyclopaedia per second. Nowadays, more and more telecommunication service providers are promoting “triple play” services to increase their revenue opportunities. Triple play, including voice, video, and high-speed data access, requires a higher bandwidth to satisfy the growing number of applications on the networks, such as video streaming, interactive gaming, and cable TV services [4]. The challenge of designing a new network that meets the nowaday application demands lies in solving the bottleneck of bandwidth availability problem in the access networks.

Access networks fall into three categories: wireless, copper, and fiber. Wireless has the lowest deployment cost. The standards for wireless access and broadband access are WiFi (802.11) and WiMAX (802.16) respectively. WiFi has a useful range of about 5 km at a data rate of 70 Mb/s and WiMAX has a range of only 100 m at 10-50 Mb/s [5]. The bandwidth shared by multiple users can support Web surfing, but is not sufficient for video applications.

Another access technology is copper wires in digital subscriber line (DSL). DSL can support a point-to-point (P2P) architecture that can provide 50 Mb/s to each subscriber. However, it is limited by the length of the copper loop due to noise in the network. DSL is capable of 50 Mb/s for loop lengths below 300 ft, but only provides 10 Mb/s at 10,000 ft [5]. For economical reasons, coaxial cables are generally shared by many subscribers, dividing available bandwidth into various channels, thus only providing a fraction of the bandwidth to each subscriber.

The final option is fiber. Fiber access networks can be either P2P or P2MP. In the P2P architecture, each subscriber has a dedicated fiber from the central office (CO). In the P2MP architecture, on the other hand, several users share one fiber from the CO, thus sharing the bandwidth on one fiber. Due to the virtually unlimited bandwidth of fiber, optical networks built on fibers are brought to the horizon as the answer to the “bandwidth” problem.

## 1.3 Optical Networks

Stimulated by rapid advances in the optical technologies in the 1980’s, the first generation of fiber-to-the-home (FTTH) was built by directly replacing existing copper wires with fibers. An optical network terminal (ONT) is installed at each subscriber’s premises and connected to an optical line terminal (OLT) by fibers. Later, fiber-to-the-curb (FTTC) is deployed to reduce costs. In this configuration, multiple subscribers share one ONT in order to reduce the number of optical components and to preserve the copper loops from the curb to the homes where DSL technology is used. Other variations of FTTH such as fiber-to-the-cabinet (FTTCab) and fiber-to-the-premise (FTTP) are also used in optical networks, depending on the cost effectiveness of specific applications [4].

### 1.3.1 Passive Optical Networks

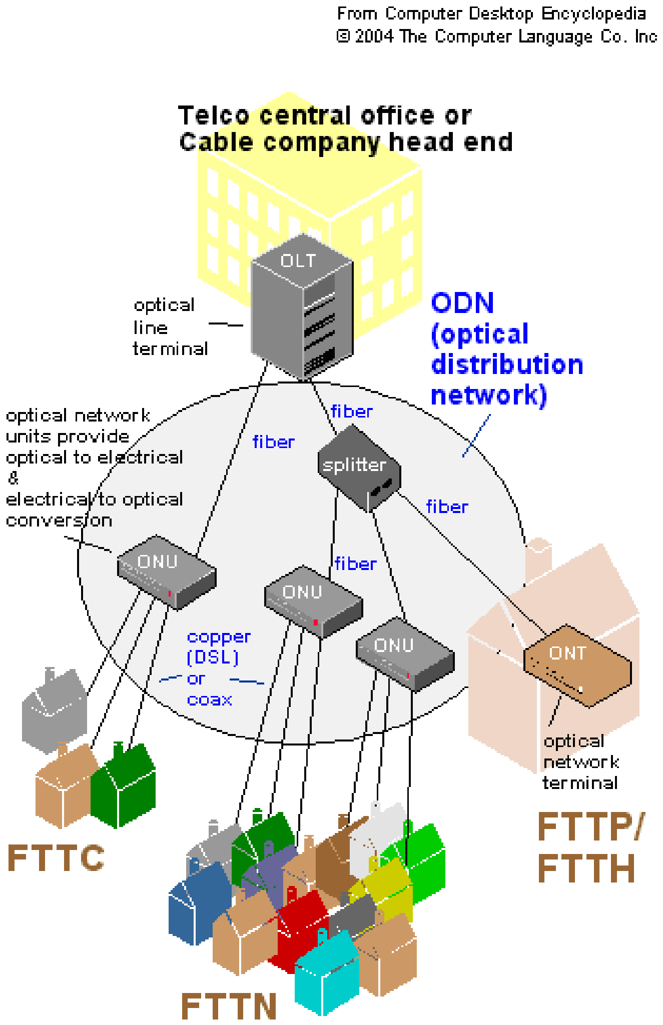
The first passive optical network (PON) was developed by British Telecom around 1989 [6]. Compared to its previous generation of optical networks, a PON has three major benefits: cost effectiveness, future safety and easy maintenance. Since all optical network units (ONUs) share the same CO, professional optical components such as connectors, lasers, and photodiodes are shared to reduce costs. Since PONs are entirely passive, they do not contain (opto)eletronics. In this way, when upgrading a PON in the future, only equipments at the CO, the ONUs, or both need to be replaced. Moreover, the passive nature also ensures minimal and straightforward maintenance of the networks [7].

Figure 1.1 shows the topology of a PON architecture. An ONT typically provides the subscribers with a plain old telephone service (POTS) interface and a high-speed interface that may be Ethernet or DSL. An OLT, on the other hand, provides an interface with the metro network and is located in the CO. In downstream, the OLT broadcasts continuous data to all ONUs. In upstream, data from all ONUs are sent back to the OLT and a medium access control (MAC) protocol is used to share the PON.

One drawback of PONs when compared to a conventional P2P architecture is that multiplexing is needed to transport respective ONU data upstream towards the CO. This can be done in either the electrical or the optical domain. Three of the most studied multiplexing techniques are time division multiplexing (TDM), dense wavelength division multiplexing (DWDM), and subcarrier multiplexing (SCM) [7].

#### TDM

In PONs, TDM is performed in the electrical domain because TDM in the optical domain is too expensive. In this technique, each ONT is granted a time slot in which to transmit its upstream data. Due to the optical path difference between each ONT and the CO, the signal propagates through the fiber with various amounts of time, which results in both amplitude and phase variations between data from all ONTs. Modern PON systems usually use a limiting amplifier to align the amplitude and a ranging protocol to measure the time variations such that the ONT bursts can be aligned when they reach the OLT. The advantage of the TDM technique is that it only requires inexpensive optical components, since TDM can be performed by digital electronics. Some disadvantages are that it is relatively less secure and less bandwidth-efficient because one wavelength channel is shared among all



From Computer Desktop Encyclopedia
© 2004 The Computer Language Co. Inc

**Figure 1.1** Passive optical network. OLT: optical line terminator; ONT: optical network terminator; ONU: optical network unit; FTTC: fiber-to-the-curb; FTTN: fiber-to-the-neighborhood; FTTH: fiber-to-the-home; FTTP: fiber-to-the-premise.

the subscribers. To use bandwidth more efficiently, dynamic bandwidth allocation (DBA) is usually deployed in upstream data transmission [4]. Also, a burst-mode receiver needs to be designed to deal with the amplitude and phase problems quickly at the beginning of each incoming burst [8]. In practice, a TDM PON uses different wavelengths for upstream and downstream transmissions. Coarse wave division multiplexing filters are used to split the wavelengths to be fed to respective receivers or transmitters. In this manner, the directional couple in one wavelength scenario is eliminated to increase the power budget and reduce the sensitivity to reflections [7].

### DWDM

In DWDM, each ONU is assigned a wavelength to transmit and/or receive data. Three main advantages are the arguments in favour of this technique: firstly, each user has the complete bandwidth of each assigned wavelength; secondly, no burst-mode receiver is needed for data synchronization since each wavelength channel is completely independent; thirdly, it is highly secure since one can not access the wavelength assigned to the other subscribers. However, along with technical difficulties imposed by wavelength management and tracking, it is very expensive to place a wavelength-tunable laser at each ONU. That is why this technique is not widely applied [7].

### SCM

SCM is a technique used in the electrical domain, where each ONU is assigned a unique carrier frequency for data transmission. In the upstream transmission, data are impressed upon each subcarrier using frequency shift keying, and the resulting radio frequency (RF) spectrum modulates the light intensity of the laser at each ONU. At the CO, signals from each ONU are selected using an RF-splitter and a bandpass filter centered on the respective channel. The advantage of the technique is that the bandwidth of the laser source is used much more efficiently when compared to digital modulation methods. However, this solution is also very expensive due to the need for high quality lasers with good modulation linearity [7].

A couple of popular types of PONs are broadband PON (BPON), Ethernet PON (EPON), gigabit PON (GPON), WDM PON, and code division multiple access (CDMA) PON. BPONs are based on Asynchronous Transfer Mode (ATM), which provides a convenient protocol for chopping the upstream data into blocks to transmit upstream bursts. EPONs, on the other hand, exploit Ethernet technology and the bursts are in the form of Ethernet frames. GPONs support ATM payloads, and also introduce a new payload adaptation mechanism called GPON Encapsulation Method (GEM) to optimize for carrying Ethernet frames at the same time. WDM PONs and CDMA PONs use WDM and CDMA respectively instead of TDMA. In CDMA PONs, each ONU uses a different signal rate and format corresponding to the subscriber's native client signal. Optical CDMA can also be used in conjunction with WDM for increased bandwidth capabilities. CDMA carries multiple client signals with their transmission spectrum spread over the same channel. The symbols from the constituent signals are encoded in a manner that is only recognizable by its own decoder, thus increasing security in the networks. However, ONU/OLT splitter ratios without amplifiers are only in the 2:1 to 8:1 range due to losses in the additional receiver splitter tree, circulators and filters. WDM and CDMA PONs have not been proven to be as cost effective as some of the other alternatives [4].

TDMA PONs, including BPON, EPON and GPON are the most popular PONs built for commercial purposes due to their relatively low installation and maintenance costs compared to other types of PONs. With similar costs for different TDMA PONs, the efficiency becomes the most dominant factor for the networks [9]. The efficiencies for different PONs are summarized in Table 1.1.

| | **BPON** | **EPON** | **GPON** |
|---|---|---|---|
| **Total bandwidth** | 622 Mb/s DS<br>155 Mb/s US | 1.25 Gb/s DS<br>1.25 Gb/s US | 2.5 Gb/s DS<br>1.25 Gb/s US |
| **Efficiency** | 72% | 49% | 94% |
| **Revenue throughput** | 448 Mb/s DS<br>112 Mb/s US | 612 Mb/s DS<br>612 Mb/s US | 2.36 Gb/s DS<br>1.18 Gb/s US |
| **Transmission format** | ATM | Ethernet | ATM + Ethernet |

**Table 1.1** Comparison of different types of PONs [9]. DS: downstream; US: upstream.

From Table 1.1, GPON has an apparent higher efficiency for a more economical solution. Moreover, with mature technology available for this network, GPON has been attracting more and more research attention.

### 1.3.2 Gigabit Passive Optical Networks

GPONs with different FTTx scenarios can offer efficient gigabit-scale data transmission to support "triple play" service for voice, video, and data. Their quality of service (QoS)

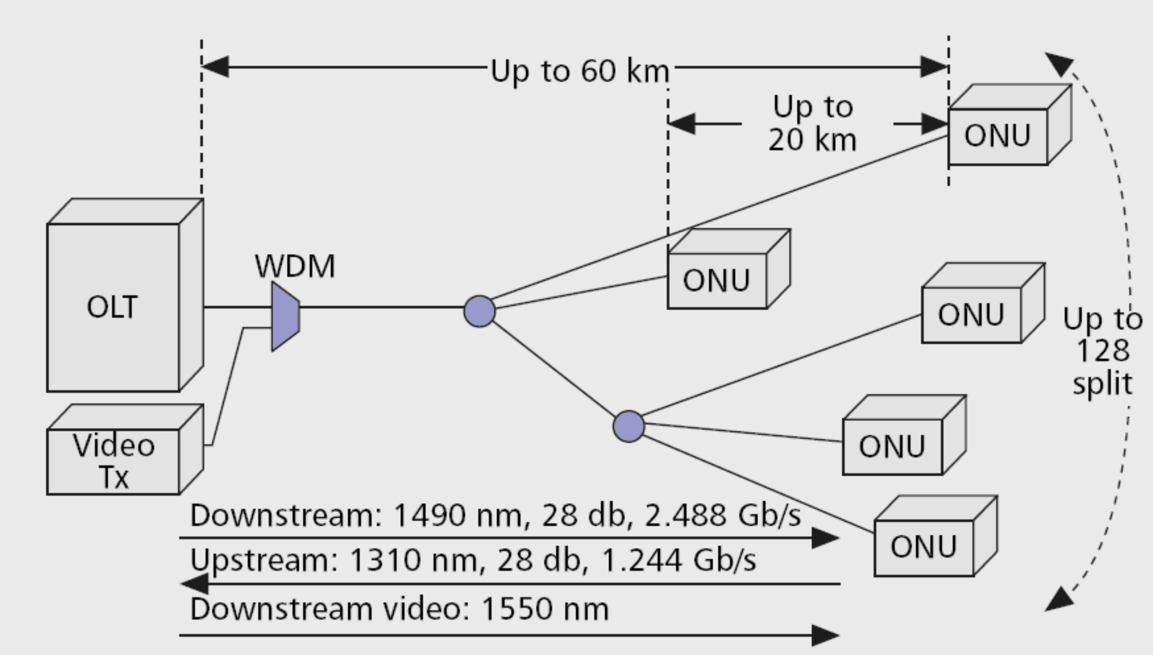

Up to 60 km
Up to 20 km
ONU
OLT
WDM
ONU
ONU
Up to 128 split
Video Tx
ONU
ONU
Downstream: 1490 nm, 28 db, 2.488 Gb/s
Upstream: 1310 nm, 28 db, 1.244 Gb/s
Downstream video: 1550 nm

**Figure 1.2** GPON physical networks architecture [5]

and bandwidth management are suitable for business users. They are specified by the International Telecommunication Union - telecommunication Standardization Sector (ITU-T) G.984 series [10, 11, 12, 13].

The physical medium dependent (PMD) layer of GPONs is specified in ITU-T G.984.2[11], covering the range of GPONs' upstream and downstream bit rates, and the optical parameters for the various rate combinations. Figure 1.2 shows the physical network architecture of a GPON. It supports different wavelengths for downstream and upstream data transmissions. One extra wavelength is allocated in the downstream direction for analog video service distribution. The network has a maximum logical reach of 60 km and 20 km differential reach between ONUs. The split ratio supported by the standard is 128. However, according to the optical budget, the actual reach and split ratio are generally lower.

The GPON transmission coverage layer (GTC) is specified by ITU-T G.984.3 [12]. This layer performs the adaptation of user data onto the PMD layer and provides basic management of the network. Two standard adaptation methods are ATM and GEM. In GPON, the OLT broadcasts data in the downstream direction to all ONUs and the data transmission is P2P. Each downstream frame begins with a physical layer operations, administration and maintenance (PLOAM) header that includes framing information and the bandwidth map. The bandwidth map is used to specify the bandwidth granted for the ONU in the next upstream frame. The PLOAM overhead field is followed by a payload area consisting of GEM frames and/or STM cells. In the upstream, bursty data from each ONU is sent to the OLT in a P2MP TDMA scheme. Each burst begins with a physical layer overhead field that consists of a guard time period, a preamble, a delimiter, and a summary of the ONT's bandwidth requests [4]. The preamble field allows the OLT to recover the timing information and signal levels. The delimiter indicates the end of the overhead and it is followed by the payload.

This thesis focus on designing an efficient high-speed burst-mode clock and data aligner at the OLT to receive upstream data from ONUs in a GPON.

## 1.4 Research Challenges

### 1.4.1 High Speed Receiver Design Challenges

Due to the bursty nature of upstream traffic in GPONs, a conventional continuous receiver can no longer be used. Arriving bursts at the OLT from each ONU vary in both amplitude and phase because they are attenuated and delayed by various amounts due to different lengths of optical paths between each ONU and the OLT. In order to sample the arriving data accurately, the amplitude and phase of each data packet have to be realigned. The fewer the bits needed for realignment, the shorter the preamble length in the overhead, which directly reflects the efficiency of the network. Moreover, with the growing data transmission rate desired by the network applications, the processing speed of the receiver also has to be scaled up accordingly. Therefore, the design of a burst-mode receiver in GPONs is challenged by both efficiency and speed.

### 1.4.2 High Speed Burst Mode Receiver Test Bed

An experimental test bed is important for both characterizing and debugging the test subject. Two difficulties in designing a test bed for the burst-mode receiver are emulating GPON upstream traffic and measuring the packet lost ratio (PLR) and the bit error rate (BER) at the receiver output.

GPON upstream traffic consists of bursts of data with various amplitudes and random phase steps between consecutive bursts. Moreover, each burst follows a specific format, including a header and a payload. Therefore, to emulate GPON upstream traffic, two high-speed pattern generators which can generate correlated specific data patterns are needed. However, high-speed pattern generators are usually very expensive with very limited access to controlled data correlation and pattern.

On the receiving end, a commercially available bit error rate tester (BERT) is usually used to measure the BER, but it does not support the PLR measurement. However, to test the realignment ability of the receiver, PLR is an essential character. Therefore, we need to look for a way to measure PLR.

## 1.5 Thesis Organization

The thesis is organized as follows:

**Chapter 2: Review of the State of the Art**

In this chapter, we review the challenges in designing an optical receiver in the applications of current network configurations and the existing techniques for burst-mode clock phase alignment present in the literature.

**Chapter 3: Forward Error Correction**

This chapter introduces a brief theoretical background of two error coding schemes: Hamming codes (including SEC-DED) and Reed-Solomon codes, which are considered as candidates in the design of the BM-CPA to increase the optical link budget.

**Chapter 4: Design and Experimental Demonstration of a Burst-Mode Clock Phase Aligner for GPON – Solution 1: Broadband CDR**

Design and simulations of a 1.244 Gb/s broadband CDR are presented in this chapter as the first solution of a BM-CPA. This design achieves 35 ns lock acquisition time at the cost of poor performance in jitter transfer and jitter tolerance.

**Chapter 5: Design and Experimental Demonstration of a Burst-Mode Clock Phase Aligner for GPON – Solution 2:2×Oversampling Clock Phase Aligner**

This chapter presents the implementation details of a 5 Gb/s burst-mode clock phase aligner (BM-CPA) with forward-error correction using a (64, 57) SEC-DED code for GPONs. This design features a 2×Oversampling algorithm and achieves instantaneous phase acquisition.

**Chapter 6: Other Solutions for Burst-Mode Clock Phase Aligner**

The last chapter concludes the thesis with a brief review of the main design and discussions of various possible extensions to the device for future research.

**AppendixA: Reed-Solomon Decoder Implementation**

In the appendix, an implementation of a Reed-Solomon decoder is explained in details. Although it is not compatible with the BM-CPA discussed in this thesis, it can be modified to be intergrated in the receiver for a better system performance.

# Bibliography

[1] http://en.wikipedia.org/wiki/History_of_communication, *History of communication.*

[2] F. J. Watamba, "A Subjective Selection of Crucial Moments in the History of (Tele-) Communication," *http://watamba.com/history_ communication.html*, 2006.

[3] B. J. Shastri, "Burst-Mode Clock and Data Recovery with FEC for Passive Optical Networks," Master's thesis, McGill University, 2007.

[4] S. S. Gorsche, "FTTH/FTTC technologies and standards," *China Communications*, vol. 3, no. 6, pp. 104–114, 2006.

[5] F. Effenberger, D. Cleary, O. Haran, G. Kramer, R. D. Li, M. Oron, and T. Pfeiffer, "An introduction to PON technologies," *IEEE Commun. Mag.*, vol. 45, no. 3, pp. S17–S25, 2007.

[6] T. Rowbotham, B. Ritchie, and C. Hoppit, "Plans for the Bishops Stortford (UK) Fibre To The Home Trials," *Proc. of IEEE Globecom'89*, pp. 1320–1325, Nov. 1989.

[7] P. Ossieur, X. Qiu, J. Bauwelinck, D. Verhulst, Y. Martens, J.Vandewege, and B. Stubbe, "An Overview of Passive Optical Networks," *International Symposium on Signal, Circuits, and Systems*, pp. 113–116, 2003.

[8] Y. Ota, R. Swartz, V. Archer, S. Korotky, M. Banu, and A. Dunlop, "High-speed, Burst-mode, Packet-capable Optical Receiver and Instantaneous Clock Recovery forOoptical Bus Operation," *J. Lightwave Technol.*, vol. 12, pp. 325–331, 1994.

[9] Z. Lou, *Designing an Embedded System for the Evaluation of the Burst-mode Transmission in a Gigabit PON Network.* PhD thesis, Ghent Univ, 2006.

[10] S. . ITU-T G.984.1, *Gigabit-Capable Passive Optical Networks (G-PON): General Characteristics*, March 2003.

[11] S. . ITU-T G.984.2, *Gigabit-Capable Passive Optical Networks (G-PON): Physical Media Dependent (PMD) Layer Specification*, March 2003.

[12] S. . ITU-T G.984.3, *Gigabit-Capable Passive Optical Networks (G-PON): Transmission Convergence Layer Specification*, July 2005.

[13] S. . ITU-T G.984.4, *Gigabit-Capable Passive Optical Networks (G-PON): ONT Management and Control Interface Specification*, June 2005.

# Chapter 2

# Review of the State of the Art

An optical communication system consists of three components: an electro-optical transducer, a fiber link and a photodetector. An electro-optical transducer, such as a semiconductor laser diode, is used to convert electrical data to optical signals. A fiber link allows the transmission of light which carries information. A photodetector converts the light from the fiber to an electrical signal to process the information received. However, transmitting the data over the fiber leads to signal attenuation and dispersion, which increases the difficulty in receiving the signal. Attenuation reduces the signal amplitude, and distortion leads to closure of the "data eye". Consequently, the system performance in terms of the bit error rate (BER) and packet lost rate (PLR) will degrade due to the low signal-to-noise ratio (SNR) sustained by the data [1]. Therefore, an optical receiver is designed at the output of the photodector in order to recover the information transmitted in the system.

## 2.1 Optical receivers

Two main components of an optical receiver are a front-end (FE) and a clock and data recovery (CDR) circuit:

### 2.1.1 Front-End

A front-end has three main parts: a photodiode, a transimpedance amplifier (TIA), and a limiting amplifier (LA). A photodiode performs optical-to-electrical conversion on the signal received from the fiber. It is followed by a TIA which amplifies the output current of the photodiode into a voltage signal. The voltage signal is passed to the LA to compensate for the limited output voltage swing of the TIA to provide sufficient logic levels.

### 2.1.2 Clock and Data Recovery

Intersymbol interference and additive noise introduced from the FE will degenerate the received signal. Hence, sampling at the optimal point will guarantee a recovery of the data with minimum BER. In modern networks, the BER requirement is extremely stringent. For example, the BER requirement in synchronous optical network (SONET) is below $10^{-15}$, which means that, on averag,e only one errorneous bit is allowed in $10^{15}$ bits. To meet such a strict BER requirement, a very demanding design process has been imposed on the CDR circuit.

Clock extraction and data retiming are the two main tasks for a CDR circuit. Accuracy in both frequency and phase of an extracted clock signal is essential since the clock signal is used to synchronize with the decision circuit in order to sample the data at its optimal point. The optimal sampling point is at the midpoint of each bit, where the signal level difference between logical one and logical zero is the largest. Therefore, it is desired to extract the clock signal with its rising clock edge at the midpoint of the received bit, as shown in Figure 2.1, to optimize the performance of the optical receiver. Other important aspects of a CDR circuit are jitter performance, phase acquisition time, and CID immunity, to name a few.

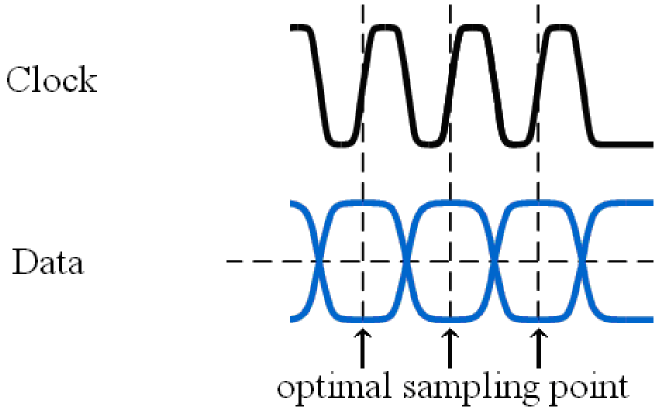


**Figure 2.1** Data being sampled by a synchronized clock signal at the optimal sampling point.

A few topologies have been proposed for a CDR circuit, such as open-loop CDRs and phase locking CDRs. Open loop CDRs have low complexity, but provide a limited phase tracking ability and are difficult to fabricate due to the highly selective band-pass filter needed in the design. CDRs based on a phase-locked loop (PLL), on the other hand, are versatile and self-sufficient, providing great tracking ability and therefore good jitter tolerance. The research focus of this thesis involves studies of the PLL-based CDRs, thus the review of the current litterature focuses on this specific type. A PLL-based CDR used in optical networks is shown in Figure 2.2.

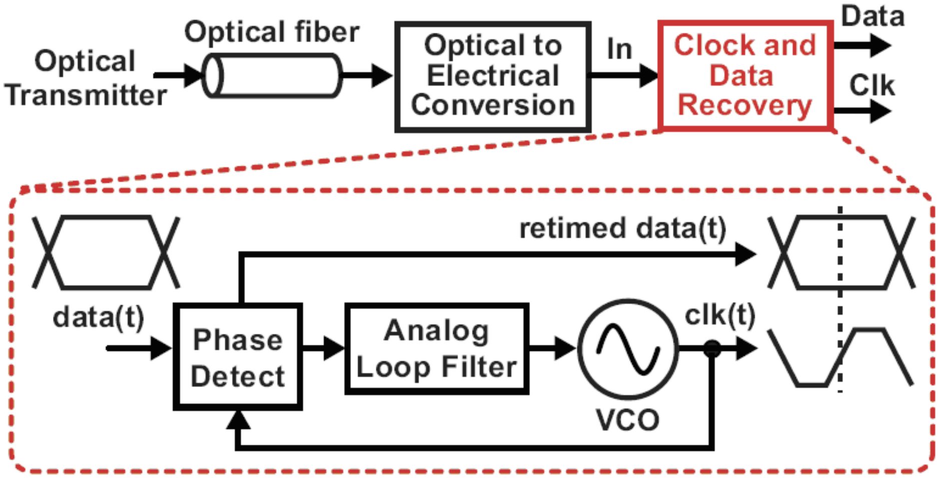


**Figure 2.2** A PLL-based CDR in an optical network.

### Phase-Locked Loop in CDR

A PLL is a feedback system that responds to the excess phase of an input signal and achieves steady-state when the frequency is matched and a constant phase variance is achieved. This is done by adjusting its own oscillator frequency and phase according to the instantaneous phase error [2]. A general PLL is composed of three main parts, a phase/frequency detector, a loop filter, and a voltage-controlled oscillator (VCO), shown in Figure 2.2. The major difference between a general PLL and the PLL in CDR is their phase/frequency detector (PFD) architectures. For a general PLL, the primary task is to lock both the frequency and phase of the VCO output with respect to the input clock signal; while in a CDR, the PFD is responsible for locking the VCO output with a random bit stream rather than a deterministic clock. As such, the PFD in CDR has a special architecture. The Hogge Phase Detector is one of the most popular PFDs among its family [3]. A Hogge phase detector is designed to detect phase errors between a periodic signal (clock) and a non-periodic signal (data) by providing a reference (REF) signal and an error (ERR) signal, which together provide a linear indication of the phase error. Figure 2.3 shows a high-level schematic diagram of a Hogge phase detector and a timing diagram to illustrate its operation.

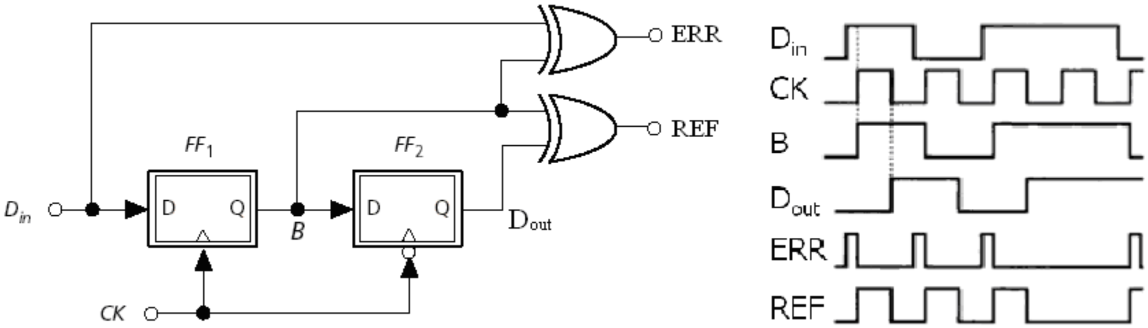


**Figure 2.3** Hogge phase detector and its operational timing diagram [4]; $D_{in}$: input data; CK: clock signal; FF: flip-flop; $D_{out}$: output data.

The ERR signal turns high at every transition of $\text{D}_{in}$ and its width is equal to the phase difference between CK and $\text{D}_{in}$. At the same time, a pulse is generated at REF by XORing signal B and $\text{D}_{out}$. The width of the REF pulse is half a cycle of the clock signal, CK. The difference between ERR and REF is taken as a linear indication of phase difference between the clock and the input data. In a phase-locked situation, both ERR and REF exhibit a pulse width of half a clock cycle. Since a Hogge Phase detector has the property of automatically retiming the input data by half a period during the phase detection process, the output data, $\text{D}_{out}$, will be sampled at its optimal point by the output clock signal.

## 2.2 Jitter

Jitter is defined as a measure of the short-term time variations of the significant instances of a digital signal from its ideal position in time [5]. The significant instances, for example, can be the optimum sampling instants as discussed earlier. Jitter is a random modulation of pulse position that is deviated from its ideal time window of reception [6].

In a communications system, the accumulation of jitter will eventually lead to data errors. Information is extracted from serial data streams by sampling the data signal ideally at the center of a data bit time, equidistant between two adjacent edge transition points. The presence of jitter changes the edge positions with respect to the sampling point. An error will then occur when a data edge falls on the wrong side of a sampling instant.

The two main classes of jitter are random jitter and deterministic jitter. Random jitter, also known as Gaussian jitter is caused by thermal noise in the system, and thus follows a Gaussian or Normal distribution. Its value is unbounded and unpredictable. Deterministic jitter, on the other hand, is predictable and reproducible with bounded peak-to-peak values.

Different methods are used to measure the jitter present in a system. Random jitter measurement and cycle-to-cycle jitter measurement are the two main means of jitter characterization. A random jitter measurement measures the difference in time between an actual clock edge and its intended position, while a cycle-to-cycle measurement is based on the difference in the period measured between one clock cycle and an adjacent clock cycle. Although measuring the same source, the two measurements are not equivalent because cycle-to-cycle measurements are frequency dependent, while random measurements are not [7].

The SONET standard specifies three measures of the jitter performance in a receiver to maintain acceptable jitter in a network [1]:

**Jitter generation** is a measure of the maximum allowable jitter generated by a system. In CDRs, jitter is mainly generated by noise in the local oscillator and the ripple in its control line. Due to the feedback loop of a PLL, noise can be cancelled to a certain degree, depending on the loop parameters.

**Jitter transfer** is a measure of the amount of jitter suppressed from the input to the output by a CDR. It is defined as the ratio of the output jitter to the input jitter at a specific jitter frequency [8]. A CDR is able to cancel jitter transfer through its feedback mechanism, and the efficiency is dependent on PLL parameters.

**Jitter tolerance** is a measure of the ability of a CDR to correctly detect incoming data with the presence of jitter. It can be defined as the amplitude of the incoming jitter that causes the BER of the recovered data to exceed a specified limit [8]. Jitter tolerance quantifies the ability of the CDRs to respond to changes in data phase by quickly and effectively altering the phase of the clock signal from synchronized sampling [9].

## 2.3 Continuous-Mode CDR

Point-to-point (P2P) optical links, where signals are transmitted synchronously, create a tightly synchronized network with data transmitted at the exact same rate. Therefore, efficient signal transmission between each element in the network is ensured. The most widely used standards for synchronous optical networking are SNOET and synchronous digital hierarchy (SDH); SONET in the U.S. and Canada, SDH in the rest of the world [10].

### 2.3.1 Continuous-Mode Challenges

Continuous-mode CDRs are digital in nature and use on-off keying (OOK). Their three main performance criteria are phase and frequency acquisition, tracking of data phase and frequency shift, and system jitter impression on the recovered clock and data [11]. Acquisition is measured in terms of time, or number of bits, and is desired to be as short as possible to minimize latency. It is the main performance criteria for a continuous-mode receiver. As the data travels over the same optical link, it undergoes equal delay over the link, such that the phase of the received data never varies significantly. Therefore, the phase tracking requirement of a continuous receiver is relaxed. The noise generated in the receiver has to be as low as possible to achieve satisfactory BER performance.

### 2.3.2 Continuous-Mode Clock and Data Recovery Performance Optimization

Acquisition and tracking are strongly related, where tracking can be thought of as acquisition on a small scale to shifts in phase or frequency [9]. However, if the tracking ability is strongly enhanced in the receiver design, the noise performance of the system is compromised. Since jitter simply consists of small modulations of the data's phase, a receiver that is very sensitive to data phase shifts can not distinguish jitter from an actual data phase shift; thus incoming jitter can not be suppressed and is passed to the output clock signal. In short, jitter tolerance performance is traded away for jitter transfer performance in designs with good phase tracking ability. Vice versa, optimizing jitter transfer for a cleaner output results in a slow tracking performance and a long phase acquisition time. Moreover, a better jitter transfer performance also trades-off internal jitter suppression, resulting in a poor jitter generation performance [2]. The challenge of designing a continuous-mode PLL-based CDR as an optical receiver mainly lies in choosing loop parameters to optimize the system performance.

## 2.4 Burst-mode CDR

Burst-mode optical receivers are needed in the point-to-multipoint (P2MP) network where each source is placed at varying distances along the fiber medium. Also, the relative clock phases of each source are not synchronized with respect to one another. Therefore, each data burst experiences a different degree of dispersion [12]. Burst-mode CDRs must recover data under a dynamic condition of inconsistent phase from one burst to another.

### 2.4.1 Burst-mode Challenges

For burst-mode communication, conventional clock recovery methods, such as the ones designed for SONET applications, and proposed in [13, 14] are not applicable. Theses PLL-based CDR circuits, with stringent jitter transfer specifications and tolerance to long sequences of identical bits, impose a narrow-band PLL with a long acquisition time [15]. Burst-mode CDRs must be designed for an acquisition time on the order of nanoseconds, as opposed to milliseconds for the minimum network latency [16], while having a jitter performance comparable to those used in continuous-mode networks.

### 2.4.2 Burst-mode Clock and Data Recovery

Several techniques have been experimentally demonstrated for fast clock recovery from burst-mode data.

#### Burst-mode CDR Based On Correlation Algorithm

This approach is based on correlating phase-delayed versions of the local oscillator against the phase of incoming data, and selecting the phase that provides the best match [17]. This technique requires a 3-bit preamble of ‘0 1 0’ at the beginning of each burst. Clock signals with various delayed phases are used to sample the incoming data. The first clock signal that samples ‘0 1 0’ correctly is considered to be synchronized with the data and used to sample the rest of the burst. Figure 2.4 shows a block diagram explaining the correlation algorithm.

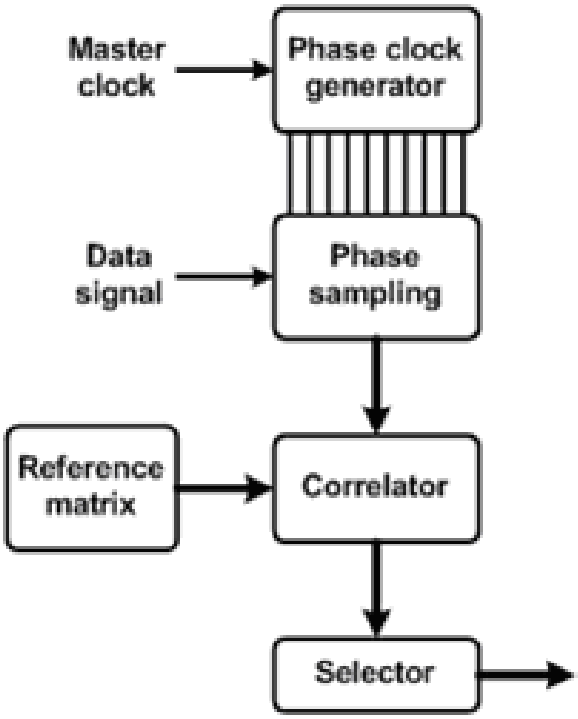


**Figure 2.4** Correlation algorithm [10]

The two advantages of this technique are low latency and high jitter rejection. Allowing only a 3-bit preamble for phase alignment greatly reduces the overhead needed in the case of continuous CDR. Since data is retimed by a clock signal generated by a local oscillator, jitter accumulated at the input data can not be passed to the output signal, therefore achieving a high jitter rejection. However, only a finite number of clock signals can be practically generated in a CDR for correlation. In [17], for example, 10 phases are generated. Thus, there is no guarantee that the selected clock signal samples at the optimal point. In this case, the SNR and therefore the BER performance of the CDR output can vary from burst to burst, depending on the relative phase difference between the incoming data and the generated clock signals.

### Burst-mode CDR Based on Gated Oscillators

This technique was first proposed in [18], in which a "gate stage" is added in the signal path of ring oscillators. These oscillators can be rapidly started at a constant phase or stopped by a digital signal applied on their start-stop-gate input, making it possible to instantly align the clock phase with data without the need for preamble bits [19]. This solution is simple and only requires low power, but has limited jitter performance and stability.

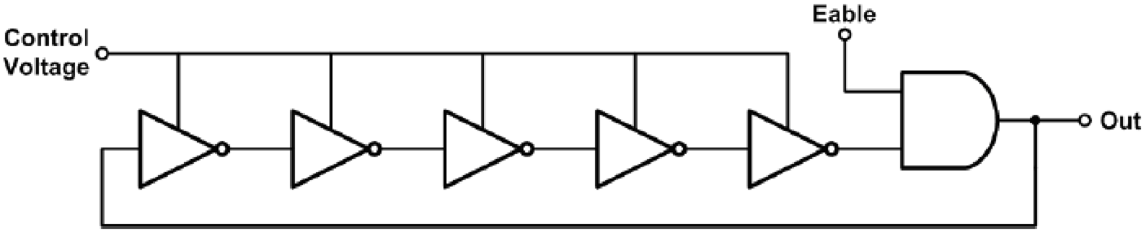
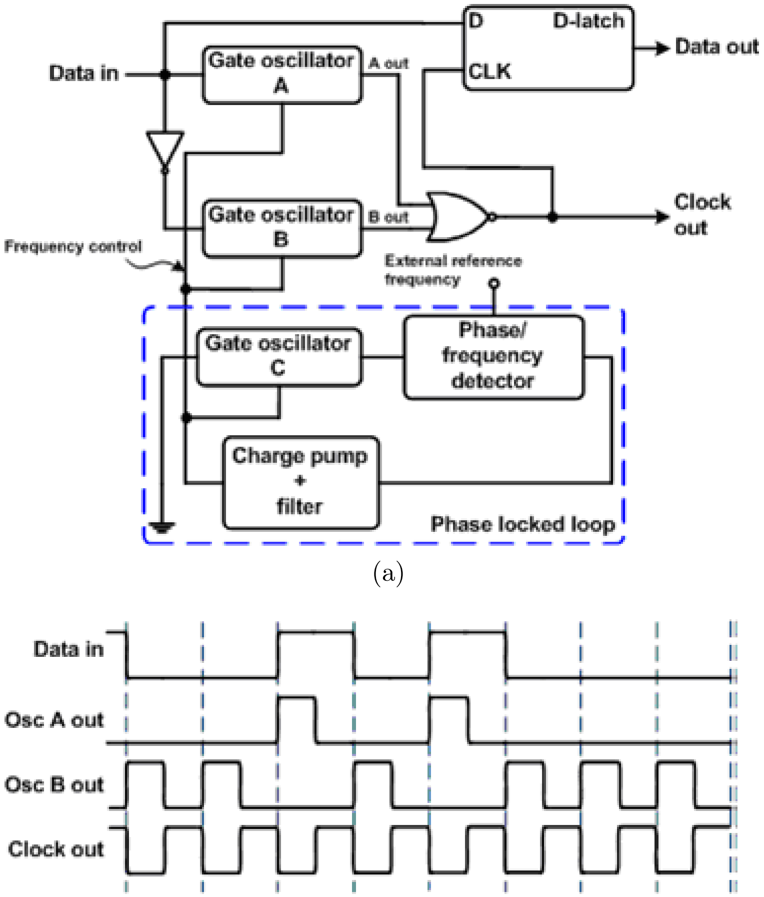


**Figure 2.5** Gated oscillator.

Figure 2.5 shows a gated ring oscillator. An AND gate acts as the gate and is used to turn the oscillation on and off. When the signal Enable is low, the oscillation stops and once Enable is high, the oscillation starts again. In addition, the oscillation always begins with the signal falling from high to low. Consequently, the output clock phase can be set to a desired value by the Enable signal. In a gated-oscillator CDR, two gated oscillators are enabled by two data signals from the same source with opposite phases such that only one oscillator is operated at any given time. The two output signals from the gated oscillators, whose initial phases of oscillation are forced into phase synchronization with the input signal, are added by a NOR gate, as shown in Figure 2.6 (a). As a result, the final clock output is always aligned with the incoming data each time a transition occurs at the input, as shown in Figure 2.6 (b).

This solution makess use of the data's own transition to align the clock signal for data retiming. It is efficient in power and simple in implementation. However, the jitter accumulated in the input data is inherited by the output clock which is used for retiming, and thus the jitter is inevitably passed from the input to the output. Jitter generation is completely traded away for jitter tolerance. The author improved his design in [20] with a separate jitter-rejection unit. However, the new design involves complex post-processing and thus loses the advantage of simplicity. Another concern about this design



(a)

(b)

**Figure 2.6** Clock recovery scheme using matched gated oscillators [10].

is its instability due to the lack of a feedback control signal. If there are inaccuracies in frequency between the clock and the incoming data, they can not be corrected and may lead to undesired BER performance.

### Burst-mode CDR with Broad Bandwidth

Continuous-mode CDRs are optimized for jitter suppression, assuming little need for tracking and almost no need for acquisition, due to the continuous nature of the data. It is argued in [9] that a PLL-based CDR can be optimized for fast acquisition and tracking, instead of jitter suppression, in order to realize burst-mode application. However, the disadvantages of this technique are the increased jitter bandwidth due to the large loop bandwidth for fast acquisition, and the increased difficulty in sampling data with low transition density. This solution is further studied in Chapter 4 where modeling and simulation of a broadband PLL based on [9] is investigated in details.

### Burst-mode CDR Based on Oversampling Algorithm

In [21, 22], a BM-CDR based on sampling the data at twice the bit rate (oversampling in time) is proposed with a novel phase-picking algorithm. Figure 2.7 shows a block diagram of such a BM-CDR.

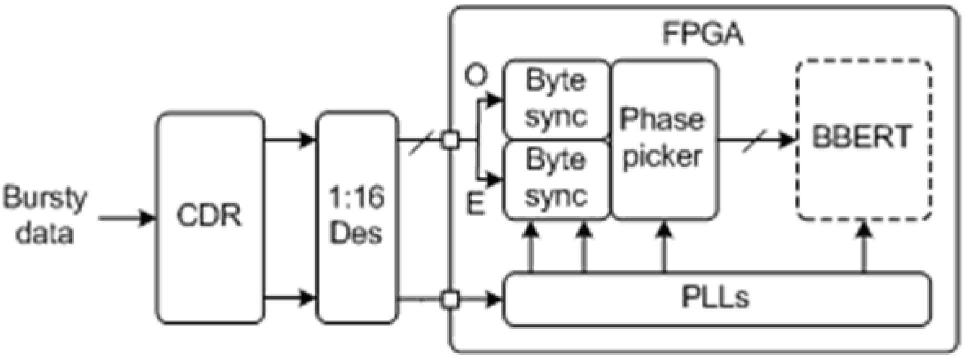


**Figure 2.7** Burst-mode clock and data recovery based on a 2×Oversampling and phase picking algorithm [21]; CDR: clock and data recovery; Des: deserializer; PLLs: phase-locked loops; BBERT: burst bit error rate tester.

In this technique, a clock signal of exactly twice the speed of the data rate is used to sample the incoming data. Thus, two information bits, which are sampled by odd and even alternate clock rising edges, are produced for every data period. The sampled bits are selected by the phase-picking algorithm for the data path with synchronized phase relation. This oversampling algorithm is deployed in the design of a burst-mode clock phase aligner discussed in this thesis. This technique is discussed in great details in Chapter 5.

## 2.5 Summary

The respectively continuous and bursty nature of data in P2P and P2MP optical links pose different challenges on the design of receivers in these two types of networks. Techniques are developed to target each challenge, but each solution has its own flaws. The technology for a continuous optical receiver has become mature over the past ten years of research. The search for an optimized burst-mode optical receiver with the desired speed, jitter performance, power consumption, and stability, is still attracting more research attention. In this chapter, an introduction of the four techniques used for burst-mode CDRs gives a peek of the main trend of research in this field. The advantages and short-comings of each technique are summarized in Table 2.1.

| **Performance** | **Correlation algorithm** | **Gated oscillator** | **Broad-band** | **2×Oversampling algorithm** |
|---|---|---|---|---|
| **Acquisition** | 3 bit | 0 bit | ∼100 bit | 0 bit |
| **Complexity** | High | Simple | Moderate | Simple |
| **Jitter rejection** | Medium | None | Low | High |
| **Sampling** | Not optimum | Optimum | Optimum | Not optimum |
| **Tracking** | Limited | Limited | Good | Excellent |

**Table 2.1** Comparison of four techniques for burst-mode CDR solutions. [10]

The research focus of this thesis is on designing a high-speed burst-mode clock phase aligner. In Chapter 4 and Chapter 5, two solutions are proposed, implemented, and analyzed, looking for the most suitable receiver technique for today's optical networks.

# Bibliography

[1] J. Savoj and B. Razavi, *High-Speed CMOS Optical Receivers.* Kluwer Academic Publishers, 2001.

[2] B. Razavi, *Monolithic Phase-Locked Loops and Clock Recovery Circuits: Theory and Design.* IEEE Press, 1996.

[3] B. Razavi, *RF Micoelectronics.* Prentice-Hall PTR, 1997.

[4] B. Razavi, *Design of Integrated Circuits for Optical Communications.* McGraw-Hill, 2002.

[5] Bell Communications Research, Inc (Bellcore), *Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria*, tr-253-core ed., 1994.

[6] P. R. Trischitta and E. L. Varma, *Jitter in Digital Transmission Systems.* Artech House Inc, 1989.

[7] Maxim Application Note 1916, *An Introduction to Jitter in Communications Systems*, 2003.

[8] H.-F. C. Group, "Jitter in Digital Communications Systems, Part 1," tech. rep., Maxim Integrated Products, Los Angeles, CA, 2001.

[9] A. Li, "Design of a Broadband PLL Solution for Burst-Mode Clock and Data Recovery in All-Optical Networks," Master's thesis, McGill University, Montreal, Canada, 2005.

[10] B. J. Shastri, "Burst-Mode Clock and Data Recovery with FEC for Passive Optical Networks," Master's thesis, McGill University, 2007.

[11] R. CO and J. Mulligan, "Optimization of Phase-Locked Loop Performance in data Recovery Systems," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 9, 1994.

[12] C. Su, L. K. Chen, and K. W. Cheung, "Inherent Transition Capacity Penalty of Burst-Mode Receiver for Optical Multiaccess Networks," *IEEE Photonics Technology Letters*, vol. 6, no. 5, pp. 663–667, 1994.

[13] I. Dorros, J. M. Sipress, and F. U. Walohauer, “An experimental 224 Mb/s digital repeated line,” *Bell Syst. Tech. J.*, pp. 993–1043, 1966.

[14] I. L. Maione, D. D. Sell, and O. H. Wolaver, “Practical 45-mbk regenerator for lightwave transmission,” *Bell Syst. Tech. J.*, pp. 1837–1879, 1978.

[15] B. Analui and A. Hajimiri, “Instantaneous Clockless Data Recovery and Demultiplexing,” *IEEE Transactions on Circuit and Systems II: Express Briefs*, vol. 52, no. 8, 2005.

[16] C. Su, L. K. Chen, and K. W. Cheung, “Theory of Burst-Mode Receiver and Its Applications in ptical Multuaccess Networks,” *IEEE Journal of Lightwave Technology*, vol. 15, no. 4, pp. 590–606, 1997.

[17] C. A. Eldering, “Theoretical Determination of Sensitivity Penalty for Burst Mode Fiber Optic Receivers,” *IEEE Journal of Lightwave Technology*, vol. 11, pp. 2145–2149, 1993.

[18] M. Banu and A. E. Dunlop, “Clock recovery circuits with instantaneous locking,” *Electronics Letters*, vol. 28, no. 23, pp. 2127–2130, 1992.

[19] Y. Ota, R. G. Swarts, V. D. Archer, S. K. Korotky, K. Banu, and A. E. Dunlop, “High-Speed, Burst-Mode, Packet Capable Optical Receiver and Instantaneous Clock Recovery for Optical Bus Operation,” *Journal of Lightwave Technology*, vol. 12, no. 2, pp. 325–331, 1994.

[20] M. Banu, A. Dunlop, W. C. Fischer, and T. Gabara, “150/30 Mb/s CMOS Non-oversampled Clock and Data Recovery Circuits with Instantaneous Locking and Jitter Rejection,” *IEEE International Solid-State Circuits Conference*, pp. 44–45, 1995.

[21] J. Faucher, M. Mukadam, A. Li, and D. V. Plant, “622/1244 Mb/s Burst-mode Clock Phase Aligner for GPON Using Commercial SONET CDRs in 2x Over Sampling Mode,” *IEEE Trans. Circuits and Systems I*, 2006.

[22] J. Faucher, M. Mukadam, A. Li, and D. V. Plant, “622/1244 Mb/s Burst-mode CDR for GPONs,” *IEEE Conf. Laser and Electro Optics*, pp. 420–421, 2006.

# Chapter 3

## Forward Error Correction

There are various error sources in an optical communication system that can corrupt the transmitted data and compromise the signal quality. In optical communication systems, two main error sources are dispersion and attenuation. Dispersion causes spreading of light pulses as they travel along the length of the fiber. The broadening of the signals results in intersymbol interference due to pulses overlapping. Attenuation exponentially reduces the optical power with the distance. It can be caused by scattering, absorption, and even fiber bending. For efficient and reliable data transmission, much effort has been expended on developing encoding and decoding methods for error control in a noisy environment.

### 3.1 Error Coding Theory

Coding theory, first presented by Claude Shannon [1], is a branch of mathematics concerned with transmitting data across noisy channels and recovering the message [2]. Two fundamental coding techniques are source coding and channel coding. Source coding is a way to compress information for efficient data transmission, whereas channel coding is a method to increase reliability through reducing information rate over the channel.

Channel coding is widely used in the design of optical receivers to recover the data at the output of an optical channel. It is achieved by adding redundancy to the information symbol vector prior to transmission, resulting in a longer coded vector of symbols that are distinguishable at the receiver. Two categories of channel coding techniques are automatic repeat request (ARQ) and forward error correction (FEC). With ARQ, receivers use a back channel to the sender to request the retransmission of lost packets. FEC, on the other hand, introduces a known structure into a data sequence which enables a receiving system to detect and possibly correct errors caused by corruption from the channel without requesting retransmission of the original information.

ARQ requires much simpler decoding equipment than performance error correction in FEC. However, in high data rate transmissions, the ARQ technique causes considerably lower system throughput due to its need for retransmissions [3]. The optical receiver of interest to us needs to operate at speeds up to 5 Gb/s, and is designed to achieve high bandwidth data transmission. Therefore, FEC is a better choice for error control in the receiver under discussion.

## 3.2 Forward Error Correction

FEC codes are a subclass of linear block codes. Block codes operate on blocks of data in which information is divided into frames, and the encoder uses only the current frame to produce its output. Linear codes can be defined as those codes in which the sum of two codewords is another codeword, and the product of any codeword by a scalar (filed element) is also a codeword. An important subclass of linear block codes is cyclic codes, which include Reed-Solomon (RS) codes. A code is cyclic if by cyclically shifting the components of a codeword one place to the right, another valid codeword is obtained [4].

The use of FEC codes is a classical solution to improve the reliability of multicast and broadcast transmissions. FEC trades-off efficiency for reliability by adding redundant check bits to a block of data using a predetermined algorithm for error detection and correction. The first class of linear block codes for error correction was invented by Richard W. Hamming in 1950 [5]. Hamming codes have a minimum distance of 3, which means that it takes 3 bit changes to move from one valid codeword to the other. This property allows the code to correct any single error in a code block. Decoding for Hamming codes is easily done through a look-up table. By properly shortening Hamming codes, a code with a minimum distance of 4 can be obtained, which is known as single-error correction and double-error detection (SEC-DED). Due to their support for high-rate transmissions and decoding simplicity, Hamming codes and their shortened versions have been used widely for error control in digital communication and data storage systems. Another FEC code that has become very popular is the RS code. It works by oversampling a polynomial constructed from the data, and is well-suited to applications where errors occur in bursts.

The three coding schemes: Hamming, SEC-DED, and RS are explained in greater details in the following sections as options to improve the performance of the burst-mode receiver discussed in this thesis.

### 3.2.1 Hamming Codes

A Hamming code is a linear error-correcting code often referred to as a systematic code because the data is left unchanged and the parity symbols are appended. More specifically, a $(n, k)$ Hamming codeword has a length of $n$ bits, out of which $k$ bits are information bits, and $n - k$ are check bits, also know as parity bits. A Hamming codeword is shown in Figure 3.1. Each parity bit is generated by XORing a smaller, overlapping portion of the data as shown in Figure 3.2. An error bit in the data is identified as a parity error in the overlapping groups of which it was a member and not in the other groups.

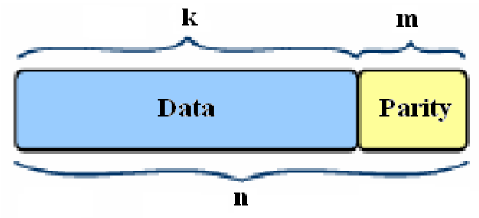


**Figure 3.1** Composition of a Hamming codeword.

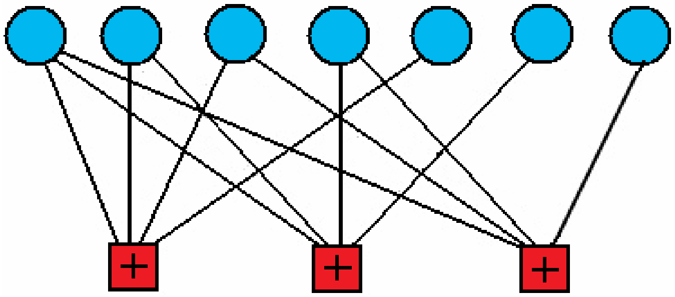


**Figure 3.2** An example of parity bits generation through overlapping portions of the data for a (10,7) Hamming code.

For any positive integer $m \geqslant 3$ (where $m$ is the number of parity-check symbols), there exists a Hamming code with the following parameters:

| | |
|---|---|
| Code length: | $n = 2^m - 1$ |
| Number of information symbols: | $k = 2^m - m - 1$ |
| Number of parity-check symbols: | $n - k = m$ |
| Error-correction capability: | $t = 1(d_{min} = 3)$ |

The parity matrix **H** is generated by combining two matrices $\mathbf{I}_m$ and **Q**, where $\mathbf{I}_m$ is an $m \times m$ identity matrix and **Q** consists of $2^m - m - 1$ columns of $m$-tuples of weight 2 or more.

$$H = [\mathbf{I}_m \ \mathbf{Q}] \tag{3.1}$$

The columns of **Q** may be arranged in any order without affecting the distance property and weight distribution of the code. From the properties of a Hamming parity matrix **H**, the minimum distance of the code is exactly 3. Hence, the code is capable of either correcting a single error in a codeword or detecting two or fewer errors, but not both simultaneously, since a codeword with a double-bit error is indistinguishable from a different codeword with a single-bit error. Thus, if a Hamming code is used for single-bit error correction, double-bit error is undetectable. Instead, when a double-bit error occurs in a codeword, the Hamming decoder attempts to "correct" the "single error bit" by adding one more error bit in the original codeword.

The generator matrix of a Hamming code is $\mathbf{G} = [\mathbf{Q}^T \ \mathbf{I}_k]$ where $\mathbf{Q}^T$ is the transpose of **Q** and $\mathbf{I}_k$ is a $k \times k$ identity matrix. Row vectors in a generator matrix describe the correlation between the party bits and their respective data bits. A codeword is generated by multiplying data with **G**. Upon reception of the codeword, an error vector known as the syndrome is obtained by XORing **H** with the codeword. If there is no error in the codeword, the syndrome vector is zero, otherwise it is nonzero [3].

### 3.2.2 Single Error Correction Double Error Detection Codes

Shortened Hamming code, also known as SEC-DED, with a minimum distance of 4 are capable of correcting codewords containing a single error bit and simultaneously detecting thoes codewords containing double-bit errors. These codes improve performance, cost and reliability with the same number of check bits. The SEC-DED parity-check matrix is constructed by properly deleting certain columns from the conventional Hamming parity-check matrix so that it satisfies the following three constraints [6]:

1. There are no all-0 columns.
2. Every column is distinct.
3. Every column contains an odd number of 1's (hence odd weight)

The first two constraints give a distance-3 code. The third constraint guarantees the code to have distance of 4, which means that single-bit error correction and double-bit error detection can be performed at the same time. Besides the three constraints, another two conditions should be satisfied for optimum performance: 1. the total number of 1's in the matrix should be minimized; 2. the number of 1's in each row of the matrix should be as close to each other as possible. Since each 1 in the matrix needs to be checked by an XOR operation, condition 1 insures minimum hardware consumption. Condition 2 is set to provide a uniform delay in the error correction process. Thus, the shortened parity-check matrix greatly reduces the complexity of the decoder and features more cost-effective hardware implementation.

When a single error occurs in one codeword during data transmission, the resultant syndrome is nonzero, and it contains an odd number of 1's. The pattern of the syndrome bits identifies the position of the single bit error and a mask is generated from a look-up table for error correction. A mask is a binary vector with value '1' at the erroneous bit location and value '0' for all the other bits. It is XOR'ed with the transmitted data to invert the error bit in the codeword. When double errors occur in a codeword, the syndrome is also nonzero, but it contains an even number of 1's. However, there is no information about the positions of the error bits. The errors will be passed to the system without correction. In other words, the Hamming distance between the transmitted and received codewords must be zero or one to guarantee reliable communication.

### 3.2.3 Reed-Solomon Codes

The properties of RS codes make them especially well-suited to applications where errors occur in bursts. This is because it does not matter to the code how many bits in a symbol are erroneous - if multiple bits in a symbol are corrupted, it only counts as one erroneous symbol to be corrected.

### Reed-Solomon Codes Basics

RS codes operate on blocks of data with symbols of $m$ bits sequences, where integer $m \geqslant 2$. For any m, there exists RS $(n, k)$ codes for all $n$ and $k$ ($n$ and $k$ are integers) for which [7]

$$0 < k < n < 2^m + 2 \tag{3.2}$$

where $k$ is the number of data symbols being encoded, and $n$ is the total number of code symbols in the encoded block. Also,

$$(n, k) = (2^m - 1, 2^m - 1 - 2t) \tag{3.3}$$

where $t$ is the symbol-error correcting capability of the code, and $n - k = 2t$ is the number of parity symbols as shown in Figure 3.3.

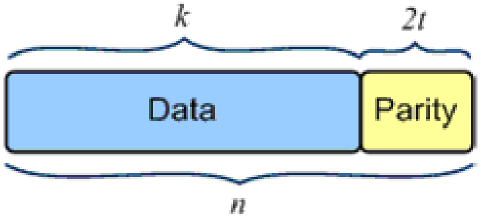


**Figure 3.3** Structure of a Reed-Solomon codeword: RS$(n, k)$ with $k$ data symbols, $2t$ parity check symbols, and total number of encoded symbols, $n$, in the block.

One important property of RS codes is that they can achieve the largest possible "code minimum distance" for any linear code with the same lengths for encoder input and output blocks. The code minimum distance is defined as $d_{min} = n - k + 1$. Therefore, the code can correct any combination of $t$ or fewer errors, where $t$ is

$$t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor = \left\lfloor \frac{n - k}{2} \right\rfloor \tag{3.4}$$

where $\lfloor x \rfloor$ means the largest integer not to exceed $x$.

The equation above leads itself to the following intuitive reasoning. One can say that the decoder has $n - k$ redundant symbols to "spend", which is twice the amount of correctable errors. For each error, one redundant symbol is used to locate the error, and the other redundant symbol is used to find its correct value [7].

From the structure of RS codes, one can also see the trade-off between reliability and efficiency. For a more reliable transmission, more parity bits can be added to obtain a higher error correction capability at the cost of fewer information bits being transmitted.

### Encoding RS code

The most commonly used method to generate a RS code is through a generator polynomial. A cyclic RS $(n, k)$ code can be defined by a polynomial

$$g(x) = g_0 + g_1 x + g_2 x^2 + \cdots + g_{n-k} x^{n-k}. \tag{3.5}$$

Each codeword can be interpreted as a codeword polynomial:

$$(c_0, c_1, c_2, \ldots, c_{n-1}) \Rightarrow c_0 + c_1 x + c_2 x^2 + \cdots + c_{n-1} x^{n-1} \tag{3.6}$$

Let $m = (m_0, m_1, \ldots, m_{k-1})$ be a block of $k$ information symbols, and its corresponding information polynomial is

$$m(x) = m_0 + m_1 x + m_2 x^2 + \cdots + m_{k-1} x^{k-1}. \tag{3.7}$$

The codeword polynomial is encoded through multiplication by $g(x)$:

$$c(x) = m(x) \cdot g(x) \tag{3.8}$$

To build a $t$-error-correcting RS code, the generator polynomial has $2t$ roots of consecutive powers of $\alpha$.

$$g(x) = \prod_{j=1}^{2t} (x - \alpha^j) \tag{3.9}$$

The general architecture of a RS $(n, k)$ encoder is shown in Figure 3.4. Since each codeword polynomial is obtained by multiplying a codeword with $g(x)$ , every codeword polynomial must have the same $2t$ consecutive powers of $\alpha$ as roots. This property provides a very convenient means for determining whether a received word is a valid codeword [8].

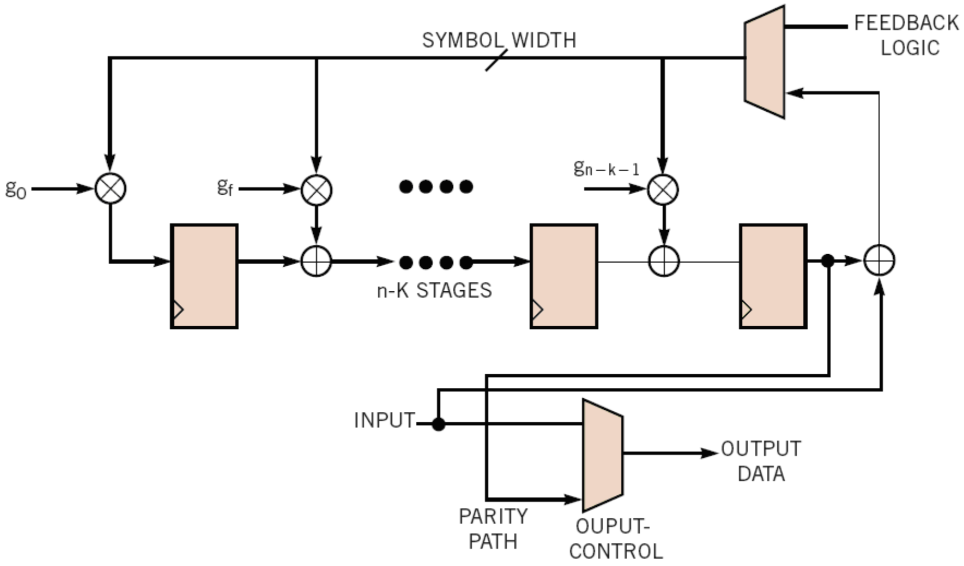


**Figure 3.4** A general architecture of a Reed Solomon $(n, k)$ encoder [9].

### Decoding of RS code

After the discovery of Reed-Solomon codes, a search began for an efficient decoding algorithm. In 1960, Reed and Solomon proposed a decoding algorithm in [10]. Although it is much more efficient than a look-up table, their algorithm is only useful for the smallest RS codes. In 1967, Berlekamp proposed his efficient decoding algorithm for both nonbinary BCH and RS codes [11]. BHC stands for the names of the inventors, Bose, Chaudhuri, and Hocquenghen. One year later, Massey demonstrated a fast-shift register-based decoding algorithm that is equivalent to Berlekamp's algorithm, which now is commonly referred to as the Berlekamp-Massey algorithm [12].

Unlike Hamming codes which are binary codes, RS codes are nonbinary. Thus, in an RS decoder, once the error locations are obtained, the errors can not be simply inverted as in the case of a Hamming decoder. The correct symbol values at the error locations need to be determined as well. The decoding of RS codes can be divided into the following steps:

1. **Syndrome Computation**

   A syndrome is generated during parity check to determine whether the received codeword is a valid member of the codeword set. If the received codeword is a member, the syndrome is zero; otherwise, the syndrome is nonzero, indicating the presence of an error or errors in the codeword. The computation can be expressed as $S_i = r(X)\Big|_{X=\alpha^i} = r(\alpha^i), \quad i = 1 \ldots n-k$, where $r(X)$ is a received codeword, and $\alpha$ is a primitive element.

2. **Error Location Calculation**

   Once a nonzero syndrome vector is computed, it is necessary to learn the location of the error or errors, which is represented by the error-locator polynomial $\sigma(X)$ : $\sigma(X) = (1+\beta_1 X)(1+\beta_2 X)\ldots(1+\beta_\nu X) = 1 + \sigma_1 X + \sigma_2 X^2 + \cdots + \sigma_\nu X^\nu$, where $\beta_1, \beta_2, \ldots, \beta_\nu$ are the error-location numbers, whose inverses are the error locations.

3. **Error Value Calculation**

   Suppose that there are $\nu$ errors in a codeword at location $X^{j_1}, X^{j_2}, \ldots, X^{j_\nu}$, then the error polynomial $e(X)$ is expressed as $e(X) = e_{j_1}X^{j_1} + e_{j_2}X^{j_2} + \cdots + e_{j_\nu}X^{j_\nu}$, where 1, 2, and $\nu$ refer to the first, second and $\nu^{\text{th}}$ error, $j$ are the error location, and $e_{j_i}$ are error values. Since the error locations are calculated in Step 2, the error values can be obtained by solving this polynomial.

## 3.3 Summary

Error detection and correction are widely used in communication and storage systems to improve transmission performances. Numerous error codes have been studied and deployed in noisy channels and less-than-reliable storage media. Selection of error coding schemes depends on various factors such as data rate, design complexity, decoding efficiency, and latency, to name a few. This chapter introduces a brief theoretical background of two error coding schemes: Hamming codes (including SEC-DED) and RS codes. They are both valid candidates to take part in the design of the burst-mode receiver presented in this thesis, and the details of their implementations will be discussed and compared in later chapters.

# Bibliography

[1] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell Syst. Tech. Journal*, pp. 379–423; 623–56, 1948.

[2] R. Pinch, “Coding Theory: The First 50 Years,” *Plus Magazine*, vol. 3, 1997.

[3] S. Lin and D. J. Costello, *Error Control Coding (2nd Edition)*. Prentice Hall, 2004.

[4] R. E. Blahut, “Theory and Practice of Error Control Codes,” *Addison Publishing Company*, 1983.

[5] I. S. Reed, “A Class of Multiple-Error-Correcting Codes and the Decoding Scheme,” *IRE Trans.*, vol. IT-4, pp. 38–49, 1954.

[6] M. Y. Hsiao, “A class of Optimal Minimum Odd-weight-column SEC DED Codes,” *IBM J. Res. Develop.*, pp. 395–401, 1970.

[7] B. Sklar, *Digital Communications: Fundamentals and Applications, 2nd ed.* Upper Saddle River, NJ: Prentice-Hall, 2001.

[8] S. B. Wicker and V. K. Bhargava, *Reed-Solomon codes and their applications*. IEEE Press, 1994.

[9] S. S. Shah, S. Yaqub, and F. Suleman, “Self-correcting Codes Conquer Noise Part 2: Reed-Solomon Codecs,” *EDN Magazine*, pp. 107–120, March 2001.

[10] I. S. Reed and G. Solomon, “Polynomial Codes Over Certain Finite Fields,” *SIAM J. Applied Math.*, vol. 8, pp. 300–304, 1960.

[11] E. Berlekamp, “Factoring Polynomials Over Finite Fields,” *Bell System Tech. Journal*, vol. 46, pp. 1853–1859, 1967.

[12] J. L. Massey, “Shift-Register Synthesis and BCH Decoding,” *IEEE Trans. Inform. Theory*, vol. IT-15, no. 1, pp. 122–127, 1969.

# Chapter 4

## Design and Experimental Demonstration of a Burst-Mode Clock Phase Aligner for GPON - Solution 1: Broadband CDR

In this thesis, two solutions for the design of a burst-mode clock phase aligner (BM-CPA) are discussed. Chapter 4 provides a solution based on a regular clock and data recovery (CDR) circuit. With carefully selected parameter values, this circuit achieves broadband performances and is able to quickly react to the instant phase variations between data bursts. However, the broadband nature of this solution has inherent disadvantages, such as poor jitter performances, which makes it impractical for applications in GPON. Therefore, another solution, based on a 2×Oversampling algorithm, is discussed in Chapter 5. This solution tackles the problem with the help of a simpler design and offers very promising experimental results.

### 4.1 Introduction

Fiber-to-the-Home (FTTH) has been proven to be an efficient medium for voice, audio, and video transfer. Passive Optical Networks (PONs) are being studied as an upgradeable and low-cost solution to the problem of limited bandwidth in local access networks in the medium of FTTH. Figure 4.1 is a graphical demonstration of such a PON where there are two transmission streams: dowstream and upstream.

The downstream path is point-to-point (P2P): continuous data is broadcasted from the optical line terminal (OLT) to the optical network units (ONUs). The transmitting side of the OLT and the receiving sides of the ONUs can therefore use continuous-mode optical and electrical components.

The upstream path is point-to-multi-point (P2MP): bursts of data are transmitted through optical fiber from the ONUs of various distances away from the OLT. Consequently, the amplitude and phase of each data burst vary from one another – bursty traffic.

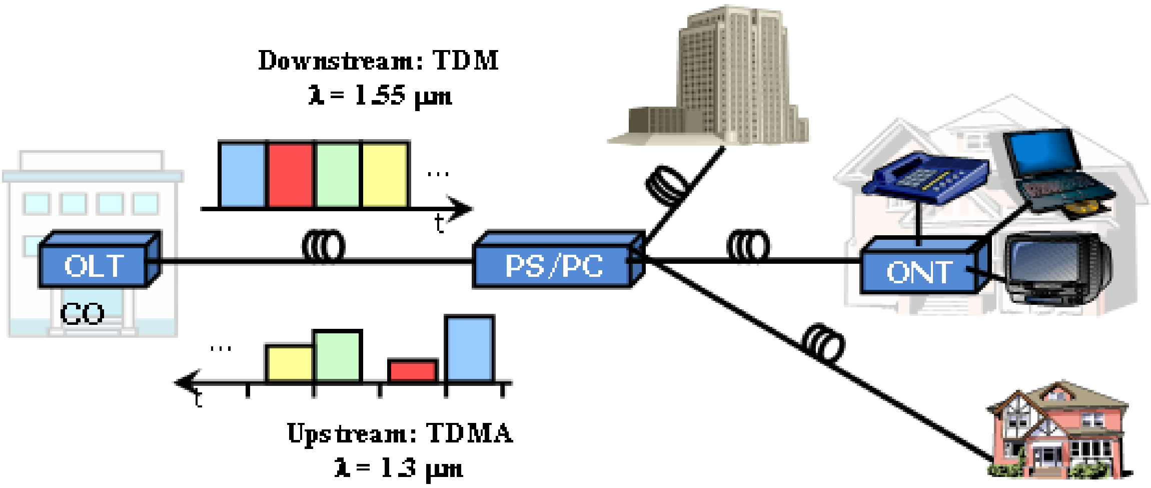


**Figure 4.1** Graphical demonstration of a passive optical network.

Due to the multi-access nature of PONs, many network nodes can interchangeably share the optical transport medium, hence the burst-mode transmission is highly preferable over the continuous-mode transmission. In the case of burst-mode transmission, transceivers used in PONs must have the capability of handling burst-mode data; more specifically, the optical receiver should have a relatively wide dynamic range and fast phase acquisition, ideally within nanoseconds, to support a short packet length at gigabit rates.

A burst-mode receiver (BMR) is in demand at the receiving end to recover a clock signal from each data burst. A BMR consists of two main components: a burst-mode front end (BM-FE) and a burst-mode clock phase aligner (BM-CPA). A BM-FE normalizes the amplitude of incoming packets, while a BM-CPA generates a clock signal that is synchronized with the incoming data in both frequency and phase. The time needed to align both amplitude and phase should be on the order of nanoseconds, while having a jitter performance comparable to continuous-mode networks. This thesis focuses on the aspect of designing the BM-CPA.

Figure 4.2 shows a typical data packet in a PON. It has four fields: a preamble, a delimiter, a payload, and a comma. The payload carries the actual information from a user; while all the other fields are overhead to ensure proper operations. The preamble is a stream of bits at the beginning of each packet to allow the system to reset the sampling threshold and generate a synchronized clock signal. The delimiter and comma are two sets of predetermined unique patterns to respectively indicate the start and end of the payload. In the upstream traffic in a PON, guard bits are also inserted between consecutive packets to avoid collisions.

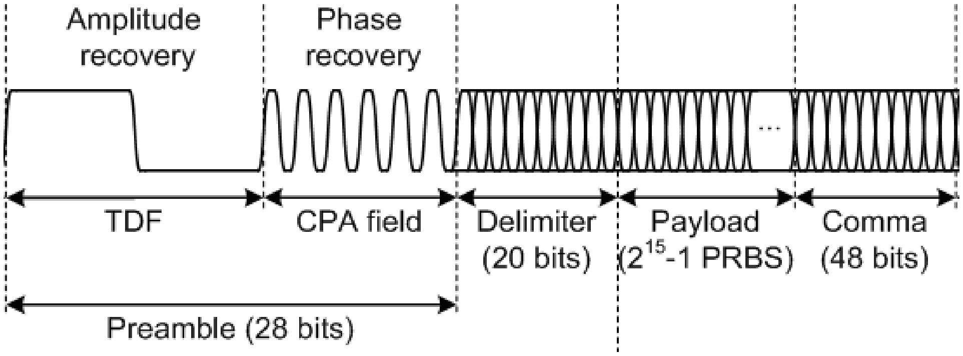


**Figure 4.2** A typical packet in a PON. TDF: threshold determination field; CPA: clock phase alignment; PRBS: pseudo-random binary sequence.

The most natural solution to extract a clock signal from a stream of data is to use a CDR circuit. Figure 4.3 shows the packet loss ratio (PLR) performance of a SONET CDR as a function of the phase difference between consecutive packets for different preamble lengths at 1.25 Gb/s. The reason the bell-shaped curves center around 400 ps is that this is the half bit period corresponding to the worst-case phase step ($\pi$ rads) and therefore the CDR is sampling exactly at the edge of the eye diagram. Preamble bits (“1010..." pattern) can be inserted at the beginning of the packets to help the CDR acquire lock. As the preamble length is increased, there is an improvement in the PLR. We observe an error-free operation (PLR $< 10^{-6}$) for any phase step after 28 preamble bits. However, the use of the preamble reduces the effective throughput and increases delay. Furthermore, this does not satisfy the 28-bit requirement for both phase and amplitude recovery, specified in the G.984.2 standard [1].

Since a SONET CDR can not deal with bursty packets in PONs due to its slow reaction to phase variations, it is desired to design a broadband CDR that is able to realign the clock phase with the current packet in the nanosecond scale.

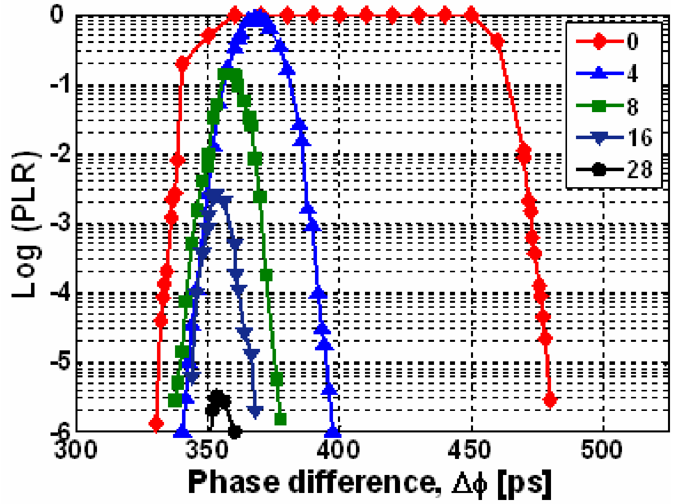


**Figure 4.3** CDR performance at various phase steps between two consecutive packets with increasing length of the preamble field. PLR: packet lost ratio.

## 4.2 Modeling Methodology of a Broadband CDR

The general behavior of a PLL is highly nonlinear during phase acquisition. However, assuming a small phase error at the input yields a near-linear behavior and allows one to assume a linear loop behavior. Moreover, if the loop bandwidth of a PLL is narrow compared to the input frequency, the state of the PLL changes only by a small amount on each cycle of the input. This allows the assumption of a time-invariant system and one only needs to consider the average behavior over many cycles [2]. However, these two assumptions can not be made while modeling the PLL in a broadband CDR. The broadband CDR targets to operate at a data rate of 1.244 Gb/s. It is also expected to respond to an instant phase step as large as $\pi$ rads, and to settle close to the steady-state within a nanosecond time frame. In order to satisfy our design specifications, a nonlinear, discrete, time domain model is needed. The nonlinearity of the model comes from the acquisition of large phase steps, and consequently, it has to be in the time domain. The model is discrete due to the sampling nature of a PLL and the broadband requirement of the CDR. However, this model alone does not provide sufficient insight into the problem, and therefore three different PLL models are used in the designing phase, aiming to tackle different aspects of the problem.

- Model 1: Basic Linear Continuous-time PLL Model

- Model 2: Linear Discrete-time Charge-Pump PLL Model
- Model 3: Nonlinear Discrete-time Charge-Pump PLL Model

Model 1 ensures that the response is not too oscillatory and settles to the steady-state quickly; once these requirements are met by the selected parameters, model 2 is used to check the stability of the resultant discrete system while responding to a small phase error; once a stable system is obtained, model 3 is applied to ensure the performance of the PLL with a large phase error ($\pm\frac{\pi}{2}$) in terms of both the stability and settling time. This design process is iterated until the design specifications are met by all three models simultaneously. Even though none of the three models alone provides a feasible solution, each model contributes to the design process and together they provide a truthful prediction of a broadband PLL.

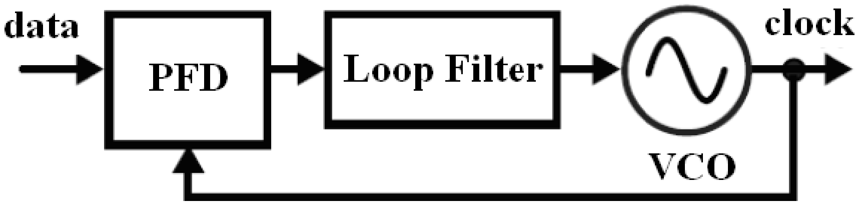


**Figure 4.4** A circuit block diagram of a PLL; PFD: phase/frequency detector; VCO: voltage-controlled oscillator

### 4.2.1 Basic Linear Continuous-time PLL model

The basic linear, continuous-time PLL model has two assumptions: small phase error and narrow loop bandwidth compared to the input frequency. Even though neither assumption is true in the broadband PLL we are designing, this model still provides a scaled approximation of the response. More specifically, this model is used as a starting point to ensure that the response is stable and settles within a nanosecond time frame. The closed-loop transfer function for this model is

$$H_c(s) = \frac{\phi_{out}(s)}{\phi_{in}(s)} = \frac{K_{PD}K_{VCO}F(s)}{s + K_{PD}(s)K_{VCO}F(s)} \tag{4.1}$$

where $K_{PD}$ and $K_{VCO}$ denote the linearized gains of the phase/frequency detector (PFD) and voltage-controlled oscillator (VCO) respectively; $F(s)$ is the transfer function of the loop filter, which significantly impacts the loop behavior. A block diagram of a PLL circuit is shown in Figure 4.4. Two options for the low-pass filter (LPF) are shown in Figure 4.5. The 1$^{\text{st}}$ order LPF consists of a resistor and a capacitor. Adding a shunt capacitor to the 1$^{\text{st}}$ order LPF, which is equivalent to adding a pole in the filter transfer function $F(s)$, results in a 2$^{\text{nd}}$ order LPF.

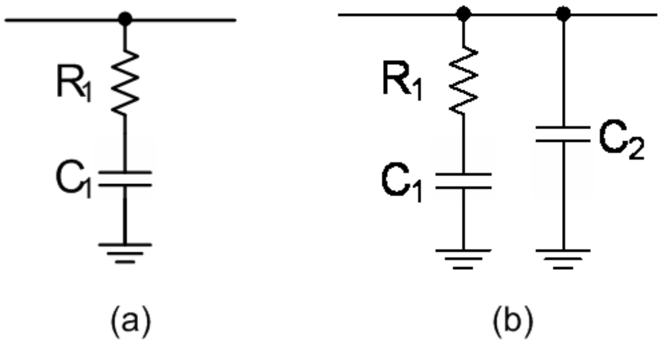


**Figure 4.5** Circuit diagram of two low pass filters (LFP); (a) 1$^{\text{st}}$ order LPF (b) 2$^{\text{nd}}$ order LPF.

The 2$^{\text{nd}}$ order loop filter is desired in the PLL circuit design because the extra shunt capacitor effectively mitigates the ripples in the control line of the VCO, thus avoiding the addition of spurious sidebands in the VCO output frequency spectrum. However, a 2$^{\text{nd}}$ order loop filter results in a 3$^{\text{rd}}$ order PLL system (it can be derived from Equation 4.1) which can not be easily analyzed. On the other hand, a 2$^{\text{nd}}$ order PLL system (with a 1$^{\text{st}}$ order loop filter) is well studied and allows easy adjustments of loop parameters to achieve a desired rise time, overshoot amplitude, and settling time. Research has shown that if the extra shunt capacitor is significantly smaller than the capacitor used in the 1$^{\text{st}}$ order filter, C1 $\gg$ C2, the 3$^{\text{rd}}$ order system loop behaviour can be accurately approximated by the 2$^{\text{nd}}$ order system [2]. Therefore, in this model, a 2$^{\text{nd}}$ order system is first used for loop parameter selections and a small shunt capacitor is added to form a 3$^{\text{rd}}$ order system for optimal loop behaviour. Their simulated performances are shown and compared in Section 4.3.1. This model provides insight into loop dynamics applicable to any PLL. However, the PLL used in the burst-mode CDR is digital and time-varying in nature; therefore, a second model needs to be considered.

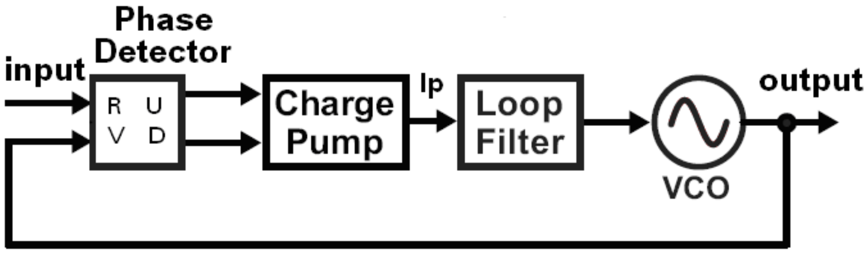


**Figure 4.6** A circuit block diagram of a charge-pump PLL; PD: phase detector

### 4.2.2 Linear Discrete-time Charge-Pump PLL model

A charge-pump PLL, as shown in Figure 4.6, has a phase detector which is modeled as a digital state machine accepting reference (R) and VCO feedback signal (V) to output *up* (U) and *down* (D) signals to a charge pump. The charge pump generates a current, $I_p$, which drives the VCO control voltage through the loop filter to adjust the VCO output frequency. This charge-pump PLL is used in the next two models.

A charge-pump PLL is a sampled system, which must be analyzed using a discrete-time model. The continuous approximation of a PLL model is valid when the input signal frequency is ten times greater than the loop bandwidth, according to the designer's bandwidth "rule-of-thumb" [2]. A bandwidth beyond 1/10 of the input frequency introduces significant discrete effects, and the stability of the PLL system must be re-evaluated. The second model does not assume a narrow bandwidth, but does assume a small phase error at the input. A Z-transform is used to characterize the discrete linear model to find an upper bound of the loop gain through pole locations of the closed-loop transfer function. The stability measure for the 3$^{\text{rd}}$ order PLL is given as

$$K < \frac{1+a}{\frac{\pi(b-1)}{b\omega_i}\left[\frac{\pi(1+a)}{\omega_i R_1 C_1} + \frac{2(1-a)(b-1)}{b}\right]} \tag{4.2}$$

where and $b = 1 + \frac{C_1}{C_2}$ and $a = e^{\frac{-2\pi b}{\omega_i R_1 C_1}}$ [5]. This model is used to further shape our design choice by imposing an upper bound on the loop gain. However, this stability measure only ensures the stability of a linear system (small phase error), whereas a nonlinear model is still needed to predict the behavior of the charge-pump PLL in the case of a large phase errors at the input.

### 4.2.3 Nonlinear Discrete-time Charge-Pump PLL model

The previous two linear models together provide a fairly reasonable choice for the parameters used in a PLL. However, the actual behavior of the broadband PLL responding to a large phase error ($-\frac{\pi}{2}$ to $\frac{\pi}{2}$ ) must be predicted by a nonlinear model. A nonlinear, discrete model for PLL was proposed by Paemel [3]. Since this model does not assume a narrow bandwidth or a small step size, it is appropriate for the burst-mode solution. This model assumes a 2$^{\text{nd}}$ order system with a 1$^{\text{st}}$ order loop filter, which drastically simplifies the otherwise very complex calculations. The dynamic behavior of the 2$^{\text{nd}}$ order system is fully characterized by two nonlinear difference equations with two state variables: $\nu(k)$, the voltage stored on the capacitor of the loop filter, and $\tau(k)$, the output pulse width of the PD. Two timing diagrams of the phase detector in the charge-pump PLL are shown in Figure 4.7. In the mathematical model, $\tau(k)$ is positive when U is high and negative when D is high.

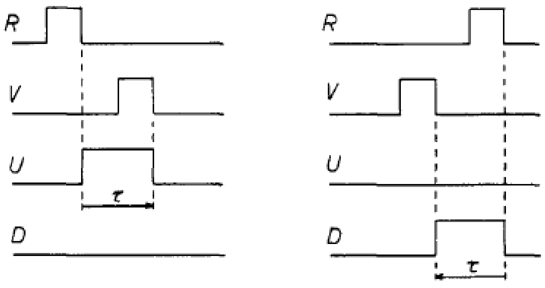


**Figure 4.7** Timing diagrams of the phase detector [3].

These two state variables are updated at every period ($T$) of the input signal. The difference equation to obtain $\nu(k+1)$ is given by

$$\nu(k+1) = \nu(k) + \tau(k+1)\frac{I_p}{C_1} \tag{4.3}$$

since a net charge of $\tau(k+1)I_p$ is added to or subtracted from the capacitor. The equation to calculate $\tau(k+1)$ is determined by equating $[T + \tau(k) - \tau(k+1)]$ to one VCO period, or simply

$$\int_0^{T+\tau(k)-\tau(k+1)} f_{VCO}(t) = 1 \tag{4.4}$$

Assuming linear operation between the VCO control voltage and its frequency, $f_{VCO}(t) = K_{VCO}\nu(t)$, equation 4.4 can be rewritten as

$$\int_0^{T+\tau(k)-\tau(k+1)} \nu(t) = \frac{1}{K_{VCO}} \tag{4.5}$$

A graphical representation of the wave forms is shown in Figure 4.8

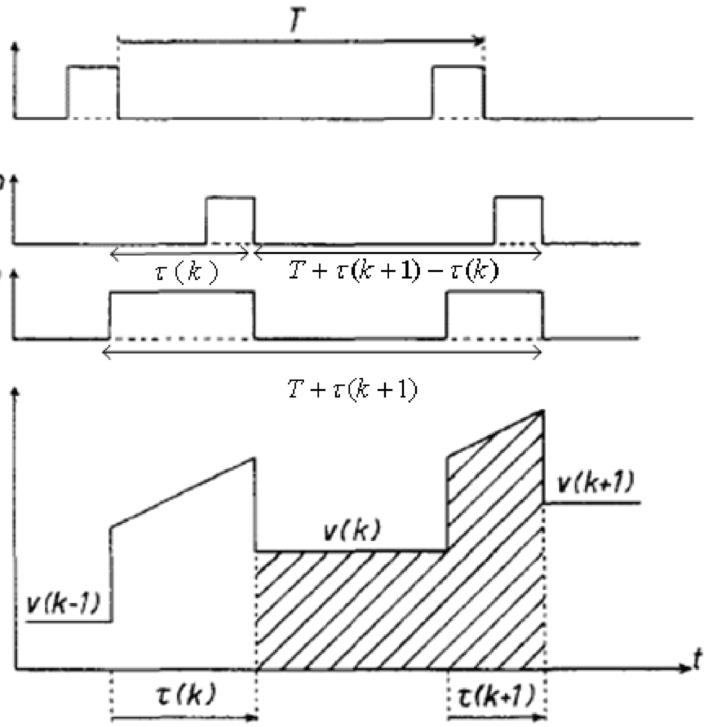


**Figure 4.8** Timing diagram of case $\tau(k) > 0$, $\tau(k+1) > 0$; $T$: input period; $V_i$ : input data; V$_{VCO}$ : VCO output voltage; $i_p$ : charge pump output current; $V_c$ : control voltage for the VCO [3].

Equation 4.5 is equivalent to putting the shaded area in Figure 4.8 to be $1/K_{VCO}$. However, depending upon the sign of both $\tau(k)$ and $\tau(k+1)$, the shape of the area varies. There are totally four possible cases. Figure 4.8 shows the timing diagram of the first case when both $\tau(k)$ and $\tau(k+1)$ are positive, representing that the input is leading the VCO output in both present and future steps. The area of the shaded region in Figure 4.8 is given by $\nu(k)\left[T+\tau(k)-\tau(k+1)\right] + I_P\left[R_1\tau(k+1) + \frac{\tau^2(k+1)}{2C_1}\right] = \frac{1}{K_{VCO}}$. Solving this equation for $\tau(k+1)$ in terms of $\tau(k)$ and $\nu(k)$, together with 4.3, we obtain a set of difference equations relating the current state to the next state for the first case. A similar argument applies to all the other three cases when at least one of $\tau(k)$ and $\tau(k+1)$ is negative. Based on the above derivation, a complete nonlinear discrete model for a broadband PLL is established. The loop parameter $\mathrm{K}_{PD}$ can be derived from the charge pump current $I_P$. Moreover, this model is important for checking the stability of the PLL when a large phase error occurs at the input. One parameter that was omitted in the analysis of this model is the second capacitor in the loop filter used to mitigate the ripples on the control line. The effect of this capacitor on the loop dynamic is negligible if it is much smaller than the capacitor in the 1st order LPF. This argument is confirmed by our simulation results shown in the next section.

## 4.3 Simulation Results

The simulation results presented in this section are obtained using the optimal PLL design parameters listed in Table 4.1. The models are simulated with the optimal values to show that the desired performances are satisfied by the three models simultaneously. A system simulation is also demonstrated in a design automation software, Advanced Design System (ADS), to predict the behavior of a broadband PLL with the parameters derived from the three models.

| **Parameters** | $R_1$ | $C_1$ | $C_2$ | $K_{PD}$ | $I_P$ | $K_{VCO}$ |
|---|---|---|---|---|---|---|
| Value | 5 kΩ | 2.5 pF | 0.3 pF | 50 mV/Rad | 20 μA/Rad | 2 GHz/V |

**Table 4.1** Optimal PLL design parameters.

### 4.3.1 Basic Linear Continuous-time PLL model Simulation Results

In the simulation of the basic linear continuous-time PLL model, a unit step function is applied as the input to Equation 4.1. The output behavior is simulated in Matlab and shown in Figure 4.9. Both a 1st order and a 2nd order loop filter transfer function $F(s)$ are used in the simulations and their effects on the system's behavior are also investigated.

Theoretically, the fastest response of a 2nd order system is critically damped. However, during the iterative design process, a critically damped 2nd order system results in an unstable system when it is simulated in model 2. The simulation results show a system that is not too oscillatory and settles to the final value quickly. It is also observed that

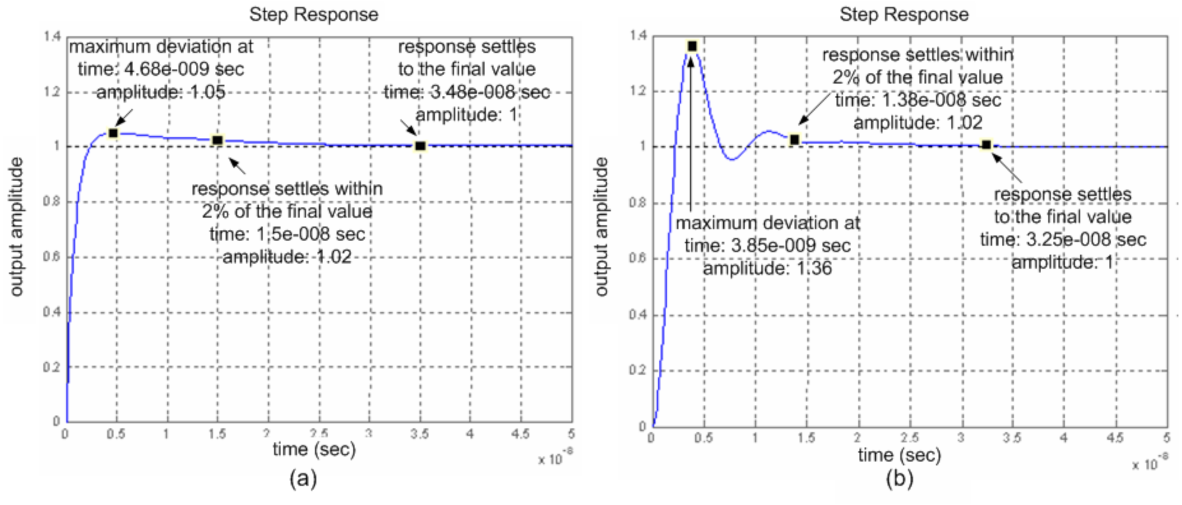


**Figure 4.9** Step response of a PLL; (a) step response of a 2nd order system with a 1st order LPF; (b) step response of a 3rd order system with a 2nd order LPF.

even though the 2nd and 3rd order systems have a slightly different step response, both of their responses settle to within 2% of the final value within 15 ns (38 bits), as shown in Figure 4.9. This settling time is in the range of the targeted specification of nanoseconds.

### 4.3.2 Linear Discrete-time Charge-Pump PLL model Simulation Results

Simulations for the linear, discrete-time charge-pump PLL model are to check the stability of the broadband PLL when the input phase step is small. Since the input frequency is 1.244 GHz and the loop bandwidth is calculated to be 200 MHz, the ratio between these two values is 6.22 < 10. Hence, according to the designer's bandwidth "rule-of-thumb", the PLL is broadband and therefore its stability needs to be evaluated in the discrete frequency domain. Equation 4.2 is used to check the stability upper bound on the loop bandwidth of the third order system, which is 1.517 GHz (> 200 MHz) for our design choices. The result indicates that the discrete charge-pump PLL is stable while responding to small phase error.

### 4.3.3 Nonlinear Discrete-time Charge-Pump PLL model Simulation Results

To analyze the stability and the phase acquisition time of the broadband PLL under large phase steps, the nonlinear discrete-time PLL model is simulated in Matlab.

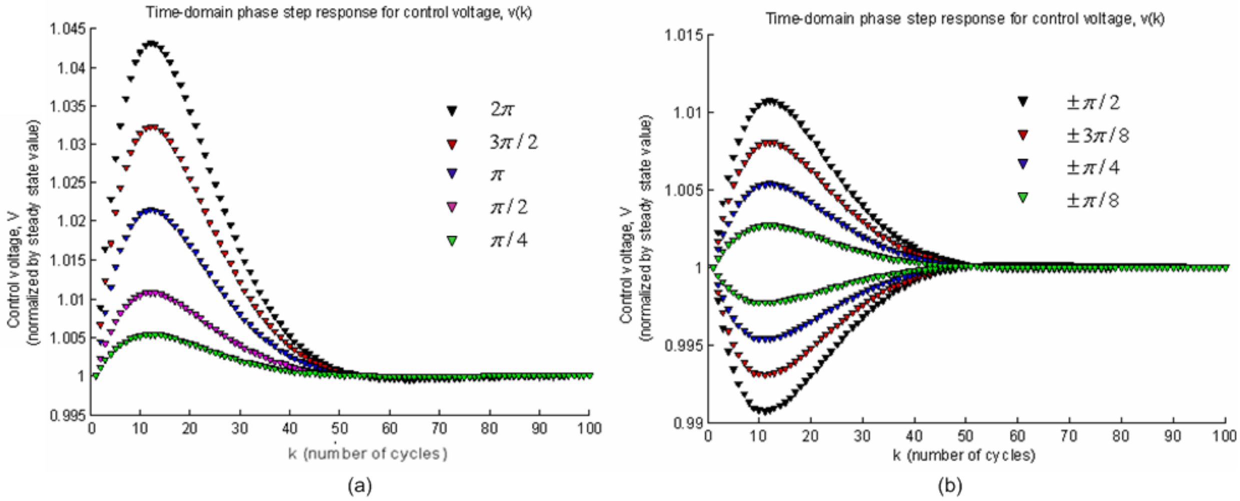


**Figure 4.10** Time domain phase step response for control voltage $\nu(k)$; (a) 0 to $2\pi$ response; (b)$\frac{-\pi}{2}$ to $\frac{\pi}{2}$ response.

From the simulation results shown in Figure 4.10, it is confirmed that the loop is stable for phase steps up to $2\pi$ and the responses are symmetric from phase step $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. The responses also settle to within 2% of the steady-state in 50 bits, which is around 40 ns for a 1.244 GHz input. As expected, the settling time in this model is in the same degree of magnitude as in the basic linear model.

### 4.3.4 System Simulation Results

At last, both $2^{\text{nd}}$ and $3^{\text{rd}}$ order systems are simulated in ADS. In this simulation, the behavior of each intermediate signal in the circuit can be further studied for a better insight of the system performance. The advantage of a $3^{\text{rd}}$ order system over a $2^{\text{nd}}$ order system is also shown in this section.

Figure 4.11 shows the circuit diagram used to simulate the designed PLL system in ADS. Since there is no Hogge detector available in the ADS library and the simulation time to acquire phase-lock in a transistor level circuit is unreasonably long, a high-level regular PD with a charge pump from the ADS library are used. Therefore, instead of

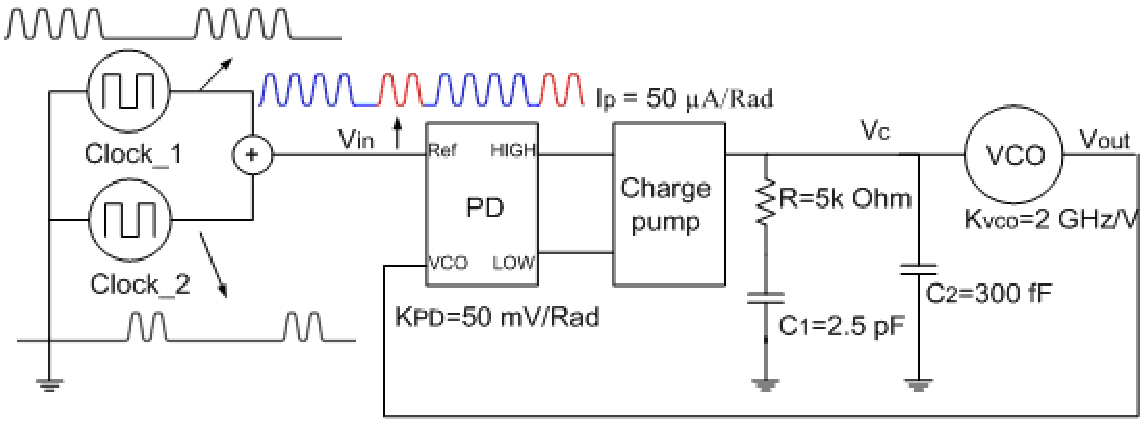


**Figure 4.11** ADS simulation schematic circuit diagram.

random input data, deterministic clock signals are needed as input. The system consists of two ideal clock sources to emulate two bursts of incoming data, a PD together with a charge pump, a 2$^{\text{nd}}$ order loop filter and an ADS built-in VCO. The two clock signals are combined to form the input of the PLL system, $V_{in}$, which has various phase differences between bursts, as shown in Figure 4.11. Since all components deploy the parameter values derived in the previous section, this circuit should truthfully demonstrate the performance of the system designed from the previous three PLL models.

Simulation results are shown in Figure 4.12. Figure 4.12 (a) shows that when a new burst is injected into the system, the VCO control voltage, $V_c$, quickly reacts to the change of input and settles to the steady-state in 35 ns. It drives the output to lock to the input signal in both frequency and phase. The "locking" state is broken only when the input burst ends. As shown in Figure 4.12 (b) and (c), during the "quiet period" of the input ($V_{in} = 0$ V), the output, $V_{out}$, gradually loses lock due to the absence of a reference signal at the input. As soon as the next burst arrives, the system starts to acquire lock again and stays locked within the burst period as shown in Figure 4.12 (b) and (d). From the circuit level simulations, the design is further proved to provide a stable performance with 35 ns lock acquisition time, which is comparable to Matlab simulations.

Figure 4.13 shows the improvement in ripple mitigation of a 3$^{\text{rd}}$ order system compared to a 2$^{\text{nd}}$ order system. When a 1$^{\text{st}}$ order loop filter is used, big ripples are present in the control line due to the large resistor in the filter. These ripples modulate the output signal of the VCO, which can lead to spurious sidebands and affect the output spectral purity [4].

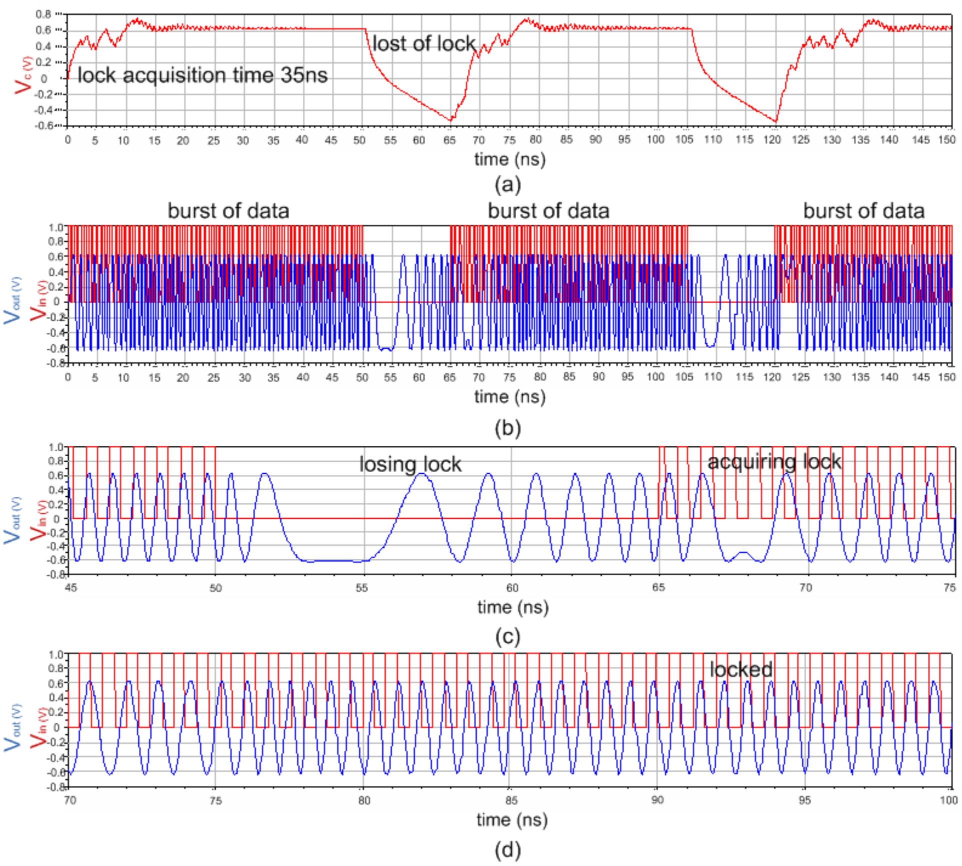


**Figure 4.12** Circuit simulation results; (a) behavior of the VCO control voltage signal $V_c$; (b) behavior of the input and output signals of the system, $V_{in}$ and $V_{out}$; (c) magnified signal behavior of $V_{in}$ and $V_{out}$ during “losing lock” and “acquiring lock” states from 45 to 75 ns; (d) magnified signal behavior of $V_{in}$ and $V_{out}$ during “locked” state from 70 to 100 ns.

A 2nd order loop filter, obtained by adding a shunt capacitor, significantly reduces ripples for a much smoother control voltage of VCO as shown in Figure 4.13 (b).

## 4.4 Summary – Potential Problems with Solution 1

This solution is a modification of a traditional phase-locked CDR to achieve burst-mode functionalities. It is accomplished by remodeling a conventional narrowband PLL to a broadband PLL which has a high feedback gain to quickly respond to phase variations for fast phase acquisition. However, a large bandwidth inevitably reduces the loop’s jitter-

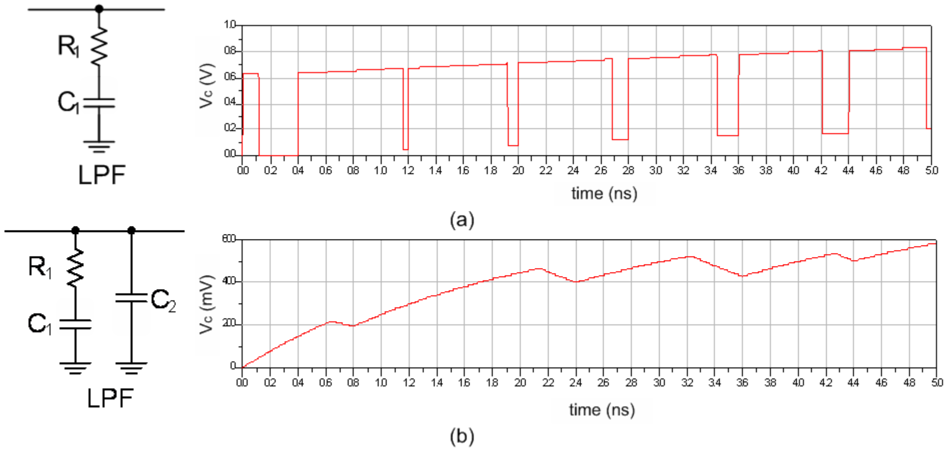


**Figure 4.13** Comparison of the effects of a 1st order and a 2nd order loop filters; (a) VCO control voltage, $V_c$, behavior with a 1st order loop filter; (b) VCO control coltage, $V_c$, behavior with a 2nd order loop filter.

suppression ability. More jitter is allowed to propagate from the incoming data through to the recovered clock, as the broadband PLL is unable to distinguish a true phase step from cycle-to-cycle data jitter. It is also more prone to clock phase drift when receiving low transition density data. Therefore, the trade-offs for a fast nonlinear PLL are a lower jitter rejection and the requirement for line coding to increase incoming data transition [5]. Since a broadband PLL is not suitable in practical applications, Chapter 5 introduces another solution that solves the random phase alignment problem in bursty incoming data while providing satisfactory jitter performances.

# Bibliography

[1] S. . ITU-T G.984.2, *Gigabit-Capable Passive Optical Networks (G-PON): Physical Media Dependent (PMD) Layer Specification*, March 2003.

[2] F. M. Gargner, “Charge-pump Phase-lock Loops,” *IEEE Trans. Commun.*, vol. 128, no. 11, pp. 1849–1858, 1980.

[3] M. V. Paemel, “Analysis of a Charge-pump PLL: A New Model,” *IEEE Trans. Commun.*, vol. 42, no. 7, pp. 2490–2498, 1994.

[4] R. E. Best, *Phase-Locked Loops: Design, Simulation, and Applications*. McGraw-Hill Professional, 4th bk&cdr ed. ed., 1999.

[5] A. Li, “Burst-Mode Clock and Data Recovery in Optical Multiaccess Networks Using Broad-Band PLLs,” *IEEE PTL*, vol. 18, no. 1, 2006.

# Chapter 5

# Design and Experimental Demonstration of a Burst-Mode Clock Phase Aligner for GPON - Solution 2: 2×Oversampling Clock Phase Aligner

## 5.1 Introduction - Burst Mode CPA Design Overview

To deal with the phase difference between two consecutive data packets, both a single SONET clock and data recovery (CDR) and a broadband CDR can be used to extract the clock signal at the beginning of every packet. However, a SONET CDR can generate clean data signals but needs too many bits in the preamble field for phase acquisition; while a broadband CDR is able to acquire the phase quickly, but passes too much jitter through to the output data. The proposed burst-mode clock phase aligner (BM-CPA) makes use of a SONET CDR and the delimiter field in every data packet to achieve instantaneous phase acquisition. The SONET CDR samples the data twice every bit period, generating redundant data bits to form two data paths. The delimiter field is originally a predetermined bit pattern indicating the start of information bits. It is used as a reference pattern to be compared with the two sampled data paths for path selection. This solution allows a SONET CDR to generate clean data signals while reducing the bits needed in the preamble field for phase acquisition. In this chapter, we demonstrate a 5 Gb/s BM-CPA that achieves instantaneous (0-bit) phase acquisition for any phase step ($\pm 2\pi$) between consecutive packets, with a packet loss ratio (PLR) $< 10^{-6}$ and bit error rate (BER) $< 10^{-10}$.

A block diagram of the BM-CPA is shown in Figure 5.1. A local oscillator (LO) or a CDR can be used to either generate a clock signal, or recover the clock from the incoming bursty data, respectively. The CDR/LO is followed by a 1:16 deserializer from Maxim-IC (MAX3995). The lower rate parallel data is then brought onto a Virtex IV field programmable gate array (FPGA) from Xilinx for further processing. On the board, it is first necessary to further parallelize the data and clock to a lower frequency that will ensure proper synchronization and better stability of these signals before they can be sent to the CPA for automatic phase acquisition. Thus, an integrated double-data rate (DDR) 1:8 deserializer is implemented on the FPGA. The CPA can be turned ON or by-passed to operate at different modes for experimental purposes. The realigned data is then sent to the (64, 57) Hamming decoder which can be turned ON for BER measurements with FEC. The decoder is followed by a custom-designed FPGA based BER tester (BERT), to selectively perform BER and PLR measurements on the payload of the packets.

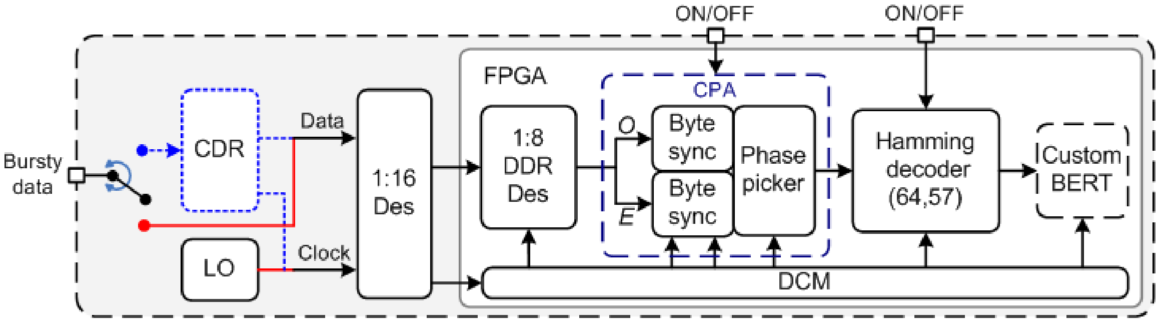


**Figure 5.1** Block diagram of the BM-CPA.

## 5.2 2×Oversampling Algorithm

The idea behind the CPA is based on a simple, fast, and effective algorithm. As shown in Figure 5.2, when the phase step between two packets increases, the clock signal from the previous packet, extracted by a CDR at normal operation mode, falls closer to the edge of the current packet “data eye”. In this case, data can not be accurately sampled by the clock rising edge that is off its optimal sampling point. The 2×Oversampling algorithm, on the other hand, produces a clock signal which has exactly twice the speed as the data rate. Using this double-rate clock signal to sample the incoming data, two information bits, which are sampled by odd and even alternate clock rising edges respectively, are produced every data period, as shown in Figure 5.2 (g). Then, the only question left is how to choose the data path that is sampled by the clock edge within the data eye opening.

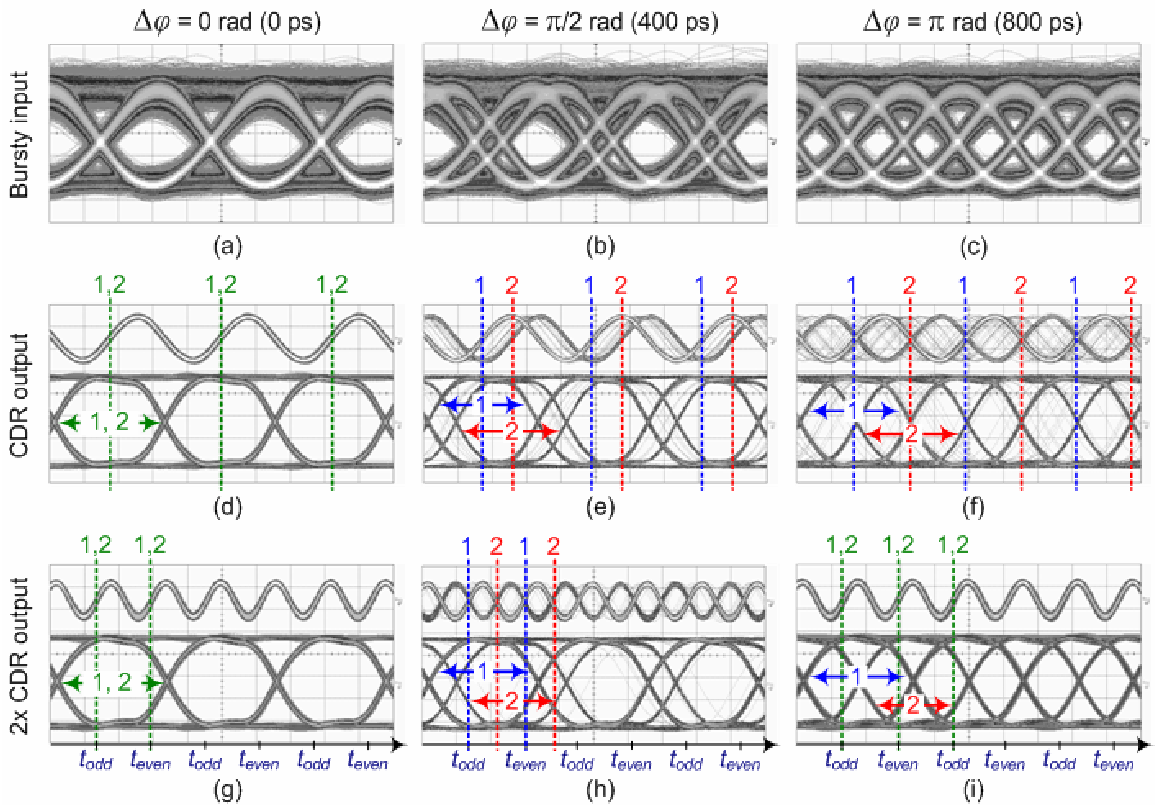


**Figure 5.2** Clock and data phase recovery using a single CDR vs. using the 2×Oversampling algorithm; (a)-(c) BM-CPA input signal for 0, $\pi/2$ and $\pi$ rad phase steps, respectively. (d)-(f) Clock and data recovered by a single CDR. (g)-(i) Clock and data recovered by the BM-CPA operated on 2× Oversampling mode.

The odd and even samples resulting from sampling the data twice, on the alternate (odd and even) clock rising edges, are forwarded to path O and to path E, respectively. The two byte synchronizers attempt to detect the delimiter on both the odd and even samples of the data. Since the clock rate is twice the data rate, there is always at least one clock (odd or even) edge that samples the data correctly, regardless of the phase step. Thus, the phase picker can use feedback from the byte synchronizers to select the right path. The realigned data is then sent to the FPGA-based BERT, implemented to selectively perform BER and PLR measurements on the payload of the packets.

## 5.3 Local Oscillator vs. SONET CDR

Our 2×Oversampling algorithm guarantees that at least one of the even or odd clock edges samples the data correctly at any phase steps between two consecutive packets. In this case, an LO can be employed to provide a constant-rate clock signal that runs at exactly twice the data rate, thus eliminating the need for complex and expensive CDR circuits based on phase-locked loops (PLLs) for a much more cost-effective solution. However, there are two trade-offs for the LO solution: 1. To apply the 2×Oversampling algorithm, the clock signal rate needs to be exactly twice the data rate. A CDR with a PLL can constantly synchronize its clock frequency with the incoming data, while an LO is not able to do so; therefore, frequency synchronization between the incoming data with the LO can be a potential problem. 2. A CDR circuit is designed to generate a clock signal which has its rising clock edge at the optimal sampling point of the data eye, while an LO has no such functionality, resulting in a lower output signal quality. The specific requirements of a system need to be carefully considered with regard to these trade-offs when choosing the appropriate configuration.

## 5.4 Mode of Operation

For experimental purposes, the BM-CPA supports three modes of operation: 1. conventional mode - essentially a SONET CDR, 2. burst mode with CDR - CPA turned on with CDR locking at twice the data rate, 3. burst mode with LO - CPA turned on with LO locking at twice the data rate. These modes of operation are useful in measuring the relative performances.

## 5.5 Data Deserialization

The main challenge in designing gigabit-capable receivers based on FPGAs lies in the limited processing speed of digital logic on commercially available FPGAs. For example, a key design component on the FPGA, the digital clock manager (DCM), provides multiple output clock signals, with various phases relative to the source clock, low clock skew, and a zero propagation delay, to be distributed throughout the board. The DCM is in essence a digital PLL, and is limited to an operating range of 24 MHz to 500 MHz. The latter frequency is 20 times slower than the targeted 10 Gb/s (2×Oversampling of the 5 Gb/s data). Thus, two stages of deserialization are employed, as shown in Figure 5.1.

The first deserialization stage is performed by the off-board 1:16 deserializer. The oversampled 10 Gb/s data and clock are deserialized to 34 parallel signals (32 differential data signals + 2 differential clock signals), each at 625 Mb/s. These signals are then brought on to the FPGA board through low voltage differential signalling (LVDS). However, the 625 MHz clock signal is 1.25 times faster than the maximum operating frequency of the DCM which is 500 MHz. Thus, a clock divider is used to reduce the frequency of the received clock to 312.5 MHz. This clock signal is then fed to a DCM block for further clock distribution throughout the system.

The second deserialization stage is based on the DDR signalling, and is accomplished by a 1:8 deserializer designed and implemented on the FPGA. It uses the 312.5 MHz DCM output clock signal to sample the 625 Mb/s incoming data at both the rising and the falling clock edges, i.e. DDR signalling. This way, each data signal is separated into two data lines by a half-rate clock signal. The same clock is then used to demultiplex these two lines of data into an 8-bit data path. In summary, the 16 input data signals are deserialized to 128 data lines at 78 Mb/s, which is eight times lower than 625 Mb/s. The advantage of this method is that the clock signal is well within the 24 MHz to 500 MHz operating range of the DCM, guaranteeing system synchronization while keeping the same harmonic content of the clock and data lines.

One concern in the circuit implementation is that the FPGA receives data at 625 Mb/s, which is the maximum speed that the digital logic can support. Processing at this speed on the FPGA board can result in stability and synchronization problems. Therefore, the DDR deserializer is manually located as close to the data input pins as possible, in order to minimize the distance over which the high-speed signals propagate on the board. The I/O pins are on the edge of the lower right corner of the board, as shown in Figure 5.3. The deserializer is located right beside them. In order to avoid synchronization issues, each high speed data path is routed to propagate approximately the same distance on the board. In this way, the risks of both stability and synchronization problems are minimized.

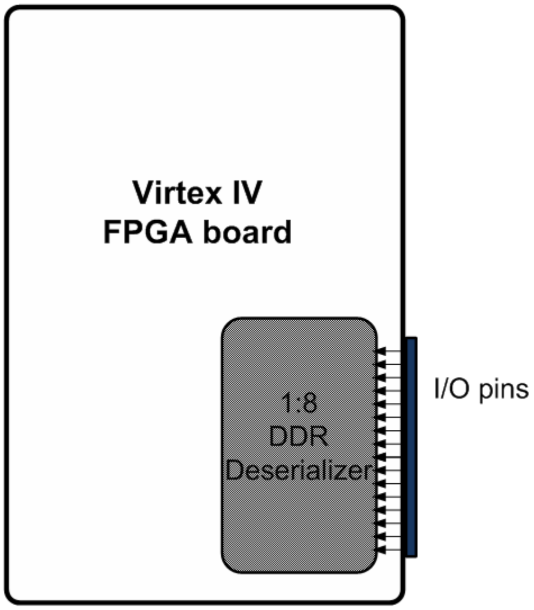


**Figure 5.3** Circuit layout on a Virtex IV FPGA board.

## 5.6 SEC-DED Decoder Implementation

We implement (64, 57) SEC-DED codes in the receiver design. In this case, every 57 bits of data is concatenated with 7 bits of parity to make a codeword of 64 bits in length. Each parity bit is generated by XORing a smaller, overlapping portion of the original data. An error bit in the data is identified as a parity error in the overlapping groups of which it was a member and not in the other groups. If a single-bit error occurs in the transmitted data block, several check bits show parity errors after decoding the retrieved codeword. The combination of these check bit errors identifies the position of any single-bit error. However, when there is more than one erroneous bit in the transmitted codeword, the decoder either passes the data without performing an error correction or miscorrects one bit in the data. In other words, the SEC-DED distance between the transmitted and received codewords must be zero or one for reliable communication.

The decoding process is done in three steps: syndrome generation, mask generation and data correction. Figure 5.4 shows the design of the decoder block. A 7-bit syndrome vector is generated through the modified parity-check matrix by XORing certain bits in the original 57-bit data with their corresponding parity bits. If the generated syndrome vector

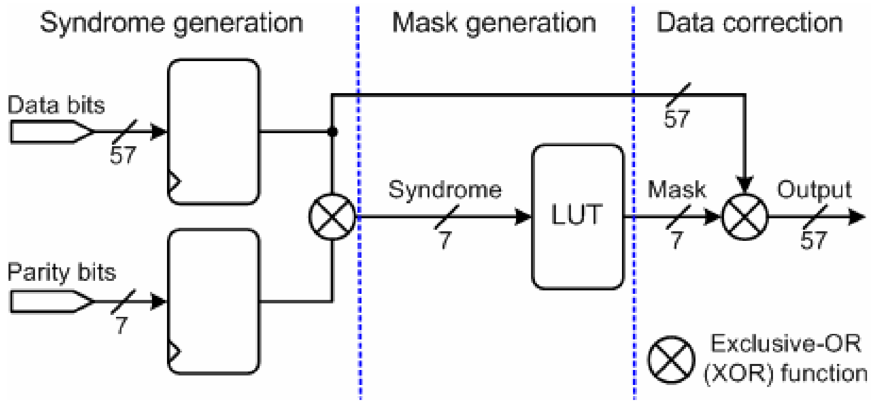


**Figure 5.4** Block diagram of a (64, 57) SEC-DED decoder.

is 0, no error occurred during data transmissions. A non-zero syndrome, on the other hand, corresponds to a single or more errors in the codeword.

When a single error is present in a data block, the generated syndrome also contains information about the exact location of the erroneous bit. A codeword with an even number of errors generates a syndrome with an even number of 1's, which signals the decoder to leave the data as it is with the errors uncorrected. If the number of errors in the data is an odd number greater than 1, there are cases in which the generated syndrome pattern is outside the columns of the code's parity-check matrix, which is a matrix consisted of syndrome vectors generated by all single-error codewords as columns. Under this condition, the errors are undetectable and no error correction is performed. In the cases where the generated syndrome pattern coincides with a column in the parity-check matrix, the decoder is forced to perform a miscorrection. However, the probability of occurrence of the last case is very low [1].

During the mask generation step, syndrome information is used to create a mask through a look up table (LUT). A mask is a 57-bit long binary vector with value '1' at the erroneous bit location and value '0' for all the other bits. It is used to XOR with the transmitted codeword to invert the error bit during the data correction step.

## 5.7 Hardware Implementation

Figure 5.5 shows the experimental setup in the lab. The setup consists of three discrete integrated circuits: a CDR, a 1:16 deserializer, and an FPGA board. They are mounted on

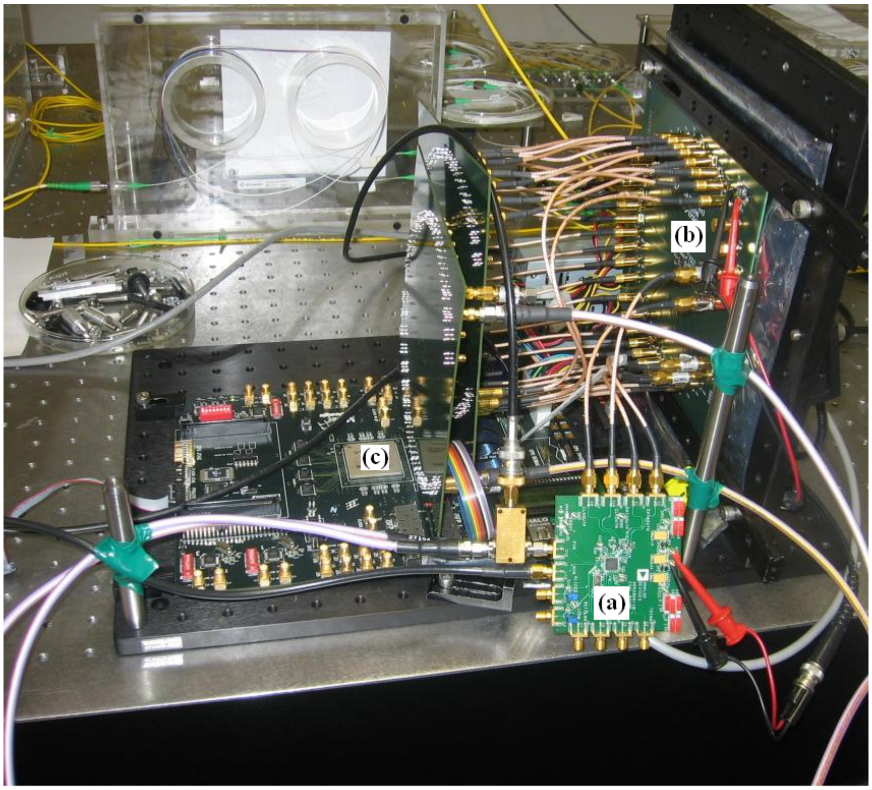


**Figure 5.5** Hardware implementation of the BM-CPA; (a) SONET CDR from Analog Devices; (b) 1:16 deserializer from Maxim; (c) Virtex IV FPGA board from Xilinx.

three separate evaluation boards as shown in Figure 5.5. The multirate CDR supports the following frequencies of interest: 622.08/1244.16 Mb/s for the conventional mode (Mode 1) and 1250/2488.32 Mb/s for the 2×Oversampling burst modes with a CDR (Mode 2). When experimenting at higher data rates, a local oscillator is employed to replace the CDR (Mode 3). The deserializer is rated up to 10 Gb/s and its main function is to reduce the bit rate by parallelizing the data. The parallel data and the recovered clock are brought onto the FPGA using a high-speed QSE connector from Samtec. However, the deserializer evaluation board uses SMB connectors. Since the outputs of the deserializer and the inputs of the FPGA both use LVDS logic, no conversion other than a connector conversion is needed at the interface between the two. The two vertical PCBs next to the FPGA, shown in Figure 5.5, serve as SMB-to-QSE connector converters.

## 5.8 Burst-Mode Clock Phase Aligner Test Setup

### 5.8.1 Burst-Mode Packet Generator

In order to emulate the upstream PON traffic, two Anritsu MP1800 pattern generators are used. The two pattern generators are programmed to output alternative PON standard packets with adjustable phase difference $|\Delta\phi| \leqslant 2\pi$ rads in between, on a 1-ps resolution. These packets are formed from guard bits, preamble bits, delimiter bits, $2^{15} - 1$ PRBS payload bits, and comma bits. At 622.08 Mb/s and 1.25 Gb/s, the guard, preamble, and delimiter bits correspond to the physical-layer upstream burst-mode overhead specified by the ITU-T G.984.2 standard [2]. However, there are no standards for PONs at data rate 5 Gb/s. Therefore, a delimiter length of 36 bits and a comma length of 48 bits are used by the author for experimental purposes. The guard bits provide some distance between two consecutive packets to avoid collisions. The preamble is used to perform amplitude and phase recovery. The delimiter is a unique pattern indicating the start of the packet to perform byte synchronization. Likewise, the comma is a unique pattern to indicate the end of the payload. The payload is the PRBS sequence where PLR and BER are measured.

The output of each pattern generator has two parts: a PON packet and a silent period with the length of the PON packet generated by the other pattern generator. Thus, once the two outputs are concatenated, each packet occupies the silent period in the other output, resulting in alternative PON packets. A silence period to emulate a phase step $|\Delta\phi| \leqslant 2\pi$ rads (with a 1-ps resolution) or a sequence of $m$ consecutive identical digits (CIDs) can be inserted between the packets during experiments.

### 5.8.2 Customly Designed Bit Error Rate and Packet Loss Rate Tester

A custom-designed designed burst bit error rate tester (BBERT) is implemented on the FPGA board to perform BER and PLR measurements. The reason a commercially available BERT is not used is because it does not support PLR measurement. However, PLR is an essential criterion to characterize the performance of the system.

The byte synchronizers constantly look for both delimiter and comma in the incoming data. If a pair of a delimiter and a comma are detected, a packet is considered received. On the other hand, if a comma is received without detecting a delimiter beforehand, a packet is considered lost. The numbers of packets received and lost are stored in the counters in the BBERT on the FPGA board.

The payload patterns of each packet sent by the pattern generators are stored in a memory block on the FPGA board. The BBERT compares the received payload with the expected data pattern in the memory block and keeps track of the total number of bits received and the number of erroneous bits. All the values stored in the counters in the BBERT are later sent to a computer via uart for BER and PLR calculations in Matlab. The reason the calculations are not done on the FPGA board is because division on hardware is slow and area inefficient.

### 5.8.3 Electrical Test Bed for Burst-Mode Clock Phase Aligner

Figure 5.6 shows the electrical test bed setup used to measure the phase acquisition time of the BM-CPA in the three modes of operation. Bursty upstream PON traffic is generated by adjusting the phase between alternating packets from two programmable ports of a pattern generator. The output of the two ports are combined via a power combiner (PC) and is then low-pass filtered by a fourth-order Bessel-Thomson filter whose 3 dB cutoff frequency is 0.7 × bit rate. The filtered data is fed to the BM-CPA and measurements stored in BBERT are sent to a computer for calculation. The computer also controls the BM-CPA to switch between different operational modes.

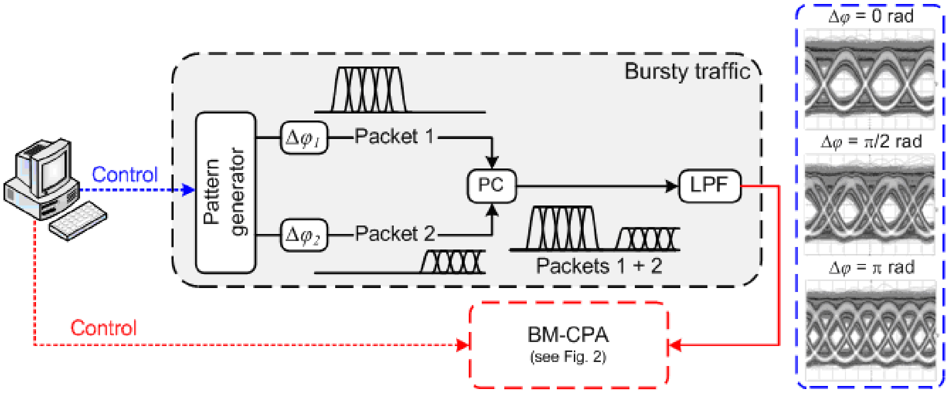


**Figure 5.6** Electrical test-bed of the BM-CPA with typical bursty traffic; PC: power combiner; LPF: low-pass filter.

### 5.8.4 Optical Test Bed for Burst Mode Clock Phase Aligner

In the optical test bed shown in Figure 5.7, the packet generators are programmed in the same way as for the electrical test bed. The two packet generators are used to drive their respective modulators (MOD). The amplitude of the packets is adjusted by employing variable optical attenuators (VOA) at the output of each laser. The upstream signals from the two ONUs are then coupled and sent over a 20-km uplink single-mode fiber (SMF). Prior to photodetection, a VOA serves to control the received power level. The output of the photodetector is passed to a transimpedance amplifer (TIA) to amplify the burst signal. The amplified signal with different voltage swings is then equalized by a limiting amplifier (LA) before being sent to the BMRx. In this test setup, imperfection of the optical components in a PON is brought into consideration and its effect on the power budget of the system is shown in the later sections.

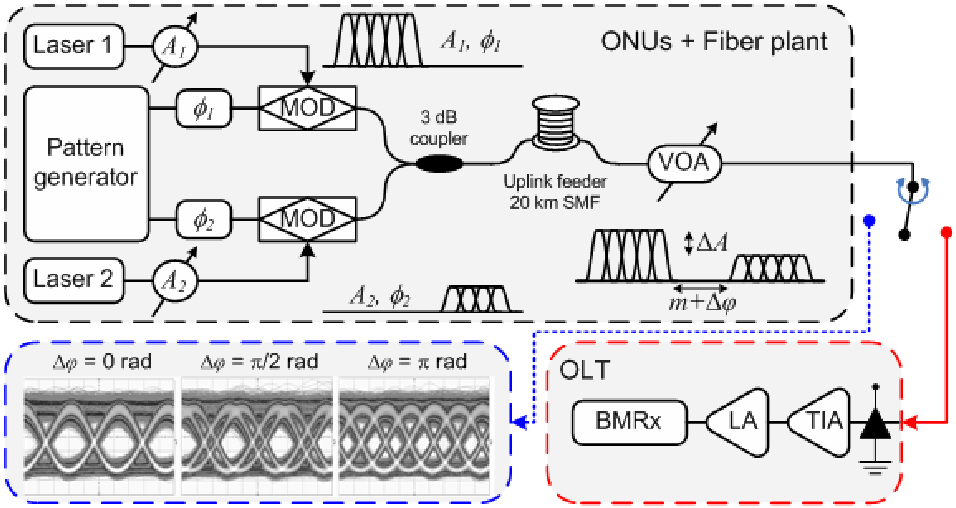
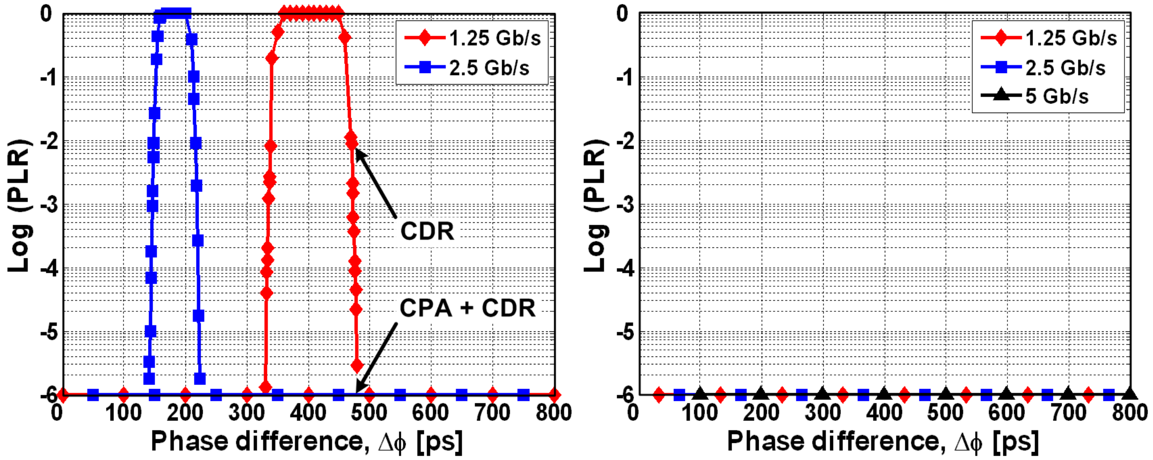


**Figure 5.7** Optical test-bed of the BM-CPA with typical bursty traffic.

## 5.9 Experimental Results

### 5.9.1 Electrical Test Bed Experimental Results

Figure 5.8 shows the PLR performance of the system as a function of the phase difference between consecutive packets in the electrical test bed. Figure 5.8 (a) depicts the phase step



(a)

(b)

**Figure 5.8** PLR performance for the BM-CPA in the electrical test bed.

response at both 1.25 Gb/s and 2.5 Gb/s data rate of the receiver at two modes of operation: 1. only the CDR and 2. the CDR followed by the CPA. When operating at mode 1, bell-shaped curves, centered at the half bit period, are obtained. A half bit period corresponds to the worst-case phase step ($\pi$ rads), and therefore the CDR is sampling exactly at the edge of the eye diagram. However, by switching on the burst-mode functionality of the receiver with the CPA (mode 2), we observe error-free operation for any phase step ($0 \leqslant \Delta\phi \leqslant 2\pi$ rads) with no preamble bits, allowing for instantaneous phase acquisition - well below the 28-bit specification.

By replacing the PLL-based CDR by the LO running at twice the data rate (mode 3), we also obtain error-free operation for any phase step with no preamble bits for data rates up to 5 Gb/s, as demonstrated in Figure 5.8 (b). The reason no experimental data is shown at 5 Gb/s while operation at mode 1 and 2 is because there is no commercially available CDR which can run at this speed. To the best of our knowledge, this is the first time that a BM-CPA is successfully implemented without CDR circuitry, resulting in a simpler and cheaper system. We note that a sensitivity penalty results from the quick extraction of the decision threshold and clock phase from a short preamble at the start of each packet [3]. However, by reducing the phase acquisition time, as demonstrated in this work, more bits are left for amplitude recovery, thus reducing the burst-mode sensitivity penalty. Alternatively, with the reduced number of bits, more bits can be used for the payload, thereby increasing the information rate.

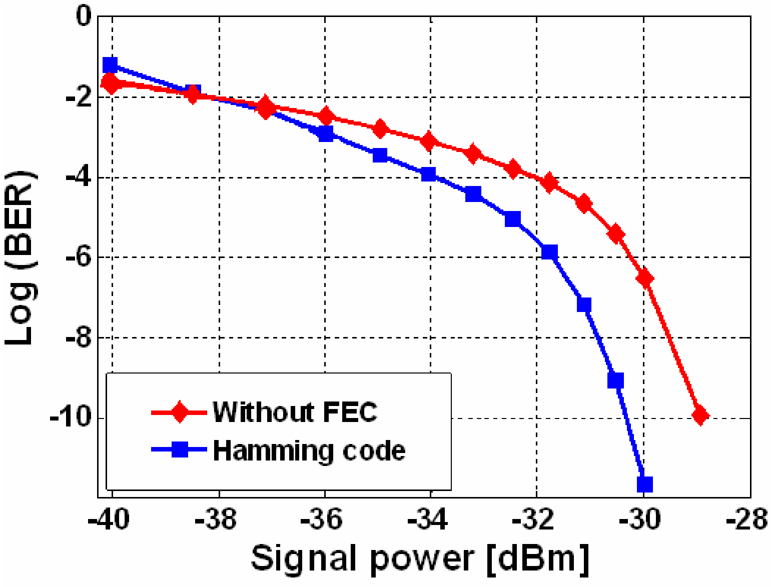


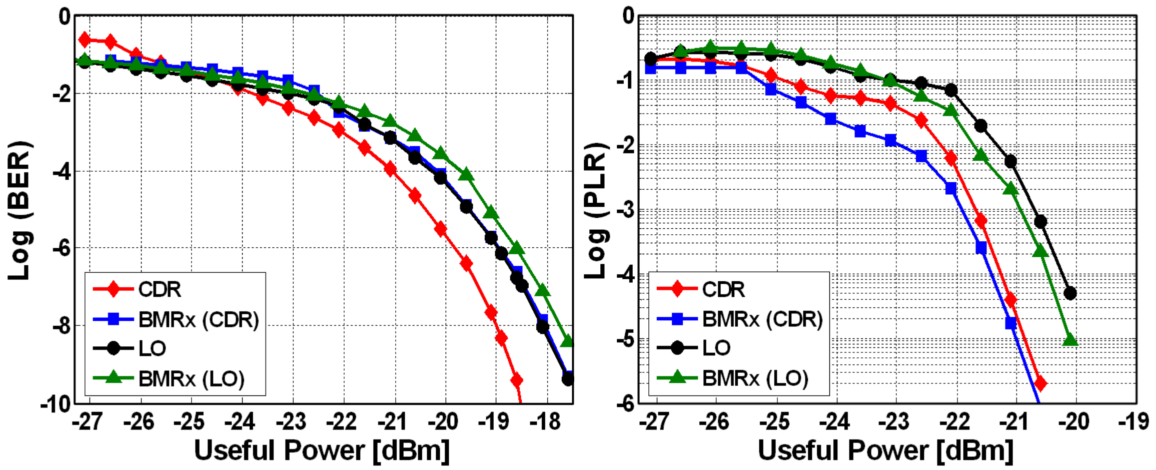
**Figure 5.9** BER performance for the BM-CPA in the electrical test bed.

To study the impact of FEC on the power budget of the GPON uplink, the BER performance of the system as a function of the received power, with and without FEC, is shown in Figure 5.9. According to the G.984.2 standard, coding gain is defined as the difference in input power at the receiver with and without FEC at a BER= $10^{-10}$. With the implemented (64, 57) SEC-DED codes, we observe a coding gain of $\sim$ 1.8 dB. The small coding gain is due to the single error correction nature of SEC-DED codes, which is not sufficient for bursty error correction.

### 5.9.2 Optical Test Bed Experimental Results

In the optical test bed, the same PLR performance at various phase differences between two consecutive packets, shown in Figure 5.8, are confirmed. In this experiment, the power budget required for error-free operation when operating at different modes of the BM-CPA is further investigated. The performances are compared to demonstrate the advantages and tradeoffs of each configuration.

Figure 5.10 shows the BER and the PLR measurements of the GPON uplink as a function of the received signal power at a data rate of 1.25 Gb/s. Performances are compared

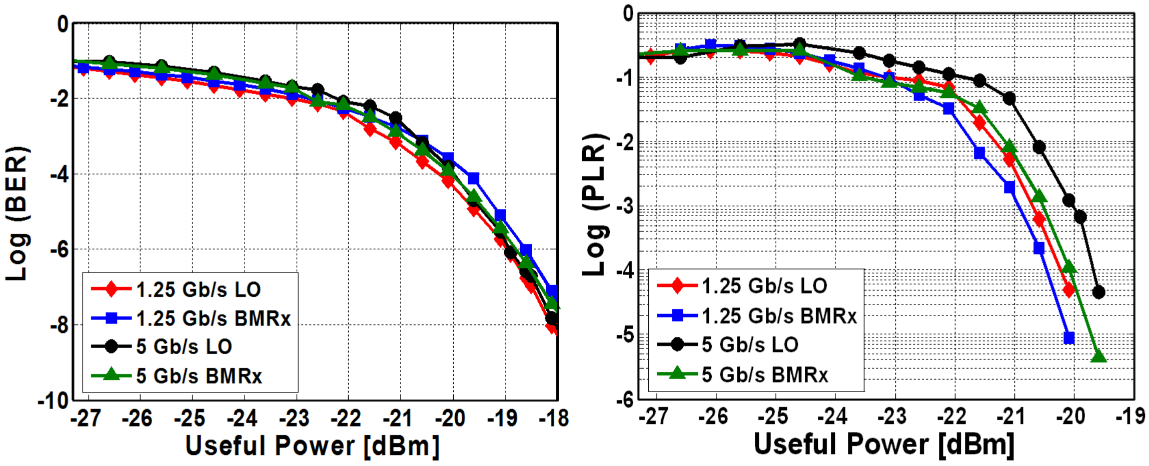


(a) (b)

**Figure 5.10** BER and PLR performances at different BM-CPA operation modes at the data rate of 1.25 Gb/s; CDR: receiver operating on non-oversampling mode with a SONET CDR; BMRx(CDR): burst mode receiver operating on 2×Oversampling mode with a SONET CDR; LO: receiver operating on non-oversampling mode with a local oscillator; BMRx(LO): burst mode receiver operating on 2×Oversampling mode with a local oscillator.

on both operation modes (non-oversampling and 2×Oversamplingg) and in both configurations (BM-CPA with a SONET CDR and with a LO). The reason the comparison is not done at 5 Gb/s is because there is no commercially available CDR which can operate at this speed. The receiver achieves a sensitivity of -18 dBm where it attains error-free operation: BER $< 10^{-10}$ and PLR $< 10^{-6}$. In comparing the BER and PLR performance by sampling at twice the bit rate for 2×Oversampling operation versus sampling at the data rate, there is a slight improvement in the case of PLR metric. This is expected due to the enabling of the 2×Oversampling algorithm and phase picking function of the receiver. As discussed earlier, the superior quality of the phase aligner with the oversampling algorithm compared to a SONET CDR alone is its insensitivity to phase variations between consecutive data packets. Moreover, the 2×Oversampling mode provides two samples of data streams for selection, which increases the probability of correctly sampling the received data packets, thus allowing the recognition of packets at a lower signal power, as shown in Figure 5.10 (b). However, when operating at twice the data rate, the CDR is less stable and inevitably introduces more noise to the system. Therefore, a power penalty of less than 1 dB must be paid at BER = $10^{-10}$, shown in Figure 5.10 (a).

Figure 5.10 also demonstrates an expected power penalty from replacing the CDR with an LO at high signal power. Since an LO does not have the phase tracking ability as a SONET CDR does, a CDR samples the data more accurately than an LO, leading to a better performance in both BER and PLR. More details are discussed in Section 5.3: Local Oscillator vs. SONET CDR. However, it can also be noticed that when the signal power is low, the BER performances are inversed: sampling with an LO gives a better BER than sampling with a SONET CDR. This is due to the fact that when the input signal power is low, it becomes more difficult for a CDR to distinguish data transitions and generate an accurate clock signal. Nevertheless, in the other configuration, the clock signal is provided by an LO with constant accuracy and power, resulting in a better BER at low input signal power.



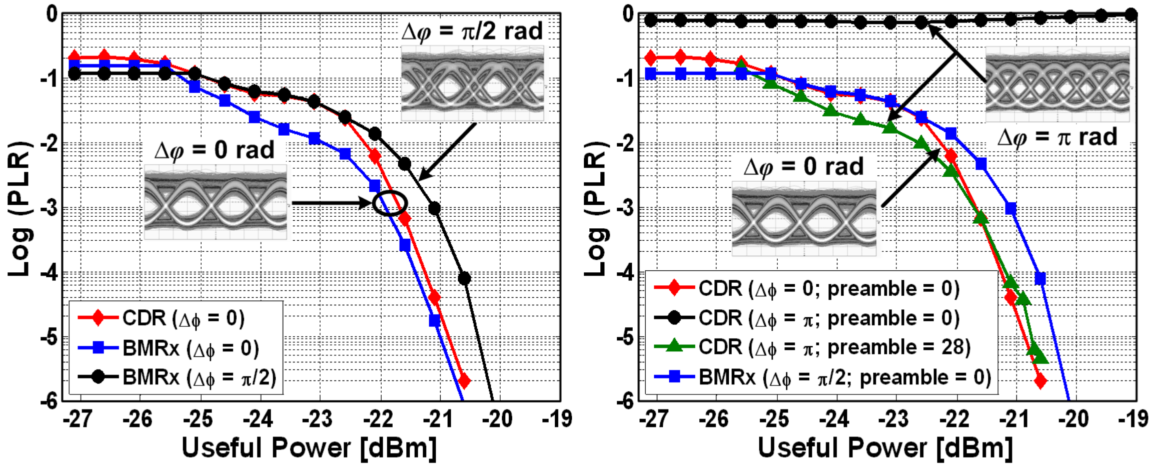
(a)

(b)

**Figure 5.11** BER and PLR performances of BM-CPA with a local oscillator at different data rates ; LO: burst-mode receiver on non-oversampling mode; BMRx: burst-mode receiver on 2×Oversampling mode.

In order to illustrate the effects of data rate on the BM-CPA, Figure 5.11 is plotted in the configuration when sampling is done with an LO at both 1.25 Gb/s and 5 Gb/s. It shows that there is a slight power penalty when the BM-CPA is operating at a higher data rate in both BER and PLR performances, due to faster electronics.

To determine the burst-mode penalty of the receiver, the PLR performances as functions of the received signal power are plotted in Figure 5.12. Figure 5.12 (a) shows the PLR



(a)

(b)

**Figure 5.12** Burst mode penalty; CDR: a SONET CDR operating at data rate is used as the receiver; BMRx: burs-mode receiver operating on 2×Oversampling mode with a SONET CDR.

performances of the CDR sampling continuous data at the bit rate (non-oversampling mode) with no phase difference, $\Delta\phi = 0$ rads, compared to the PLR performances of the BMRx (2×Oversampling mode) sampling bursty data at both zero phase difference and the worst-case phase difference, $\Delta\phi = \frac{\pi}{2}$ rads. All measurements are made for a 0-bit preamble. The same improvement as in Figure 5.10 (b) is confirmed in 2×Oversampling mode with no phase difference. However, a power penalty of less than 1 dB is observed at the worst-case phase difference. If there does exist a phase difference between the consecutive packets, the CDR alone will not be able to recover any packet, regardless of the signal power, resulting in a worst-case PLR $\sim 1$ as shown in Figure 5.12 (b). Only when a 28-bit preamble is appended to each packet does the PLR performance of the CDR become comparable to the PLR performance obtained by the CDR with zero preamble bits and no phase difference. Since phase steps in the GPON uplink are inevitable, the 1-dB power penalty may be a small price to pay to avoid not receiving any packet at all.

Figure 5.13 shows the CID immunity of a SONET CDR and the BM-CPA on 2×Oversampling mode. When consecutive ‘1’s or ‘0’s appear in the received data, the clock signal recovered by a SONET CDR starts to drift away from the optimal sampling point due to the low data transition density. Since the clock signal on 2×Oversampling mode is twice the data rate, there is a higher chance that a clock rising edge is around the optimal sampling point

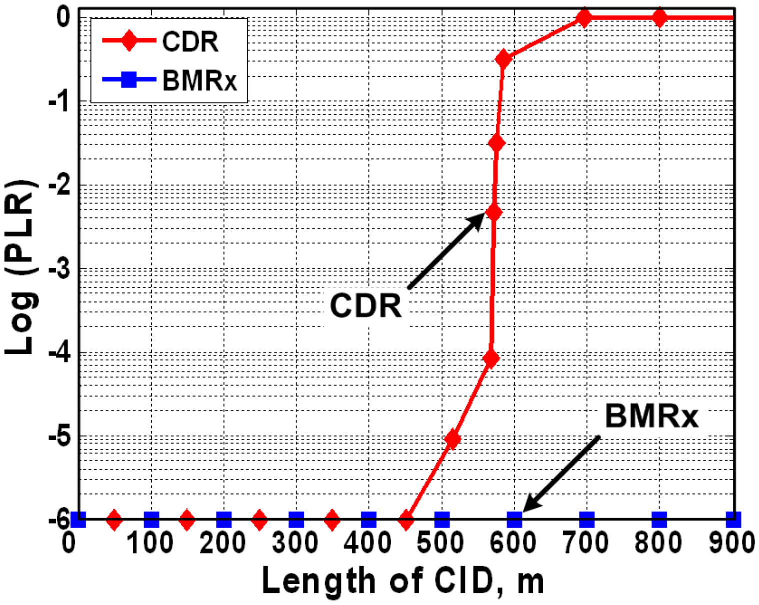


**Figure 5.13** Comparison of CID immunity of a CDR with the BM-CPA; CDR: a SONET CDR operating at data rate is used as the receiver; BMRx: burst mode receiver operating on 2×Oversampling mode with a SONET CDR.

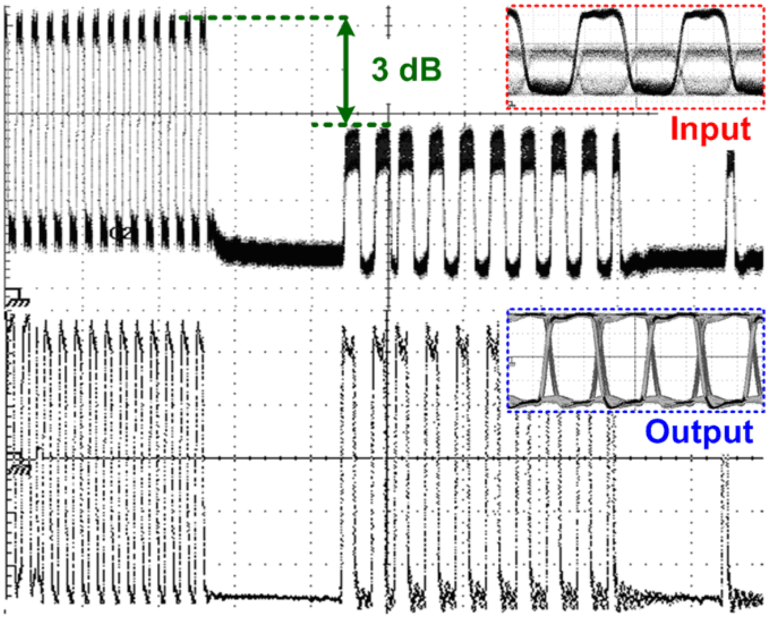


**Figure 5.14** Data input and output waveforms for dynamic range measurements.

to recover the data. Therefore, the BM-CPA is expected to have a larger maximum CID period than a CDR. The experiment shows that our receiver can support more than 1000 CIDs with error-free operation, which is $\sim 14\times$ more than the minimum 72 CIDs specified in G.984.2.

The dynamic range of the receiver is measured to be 3 dB, as shown in Figure 5.14. This relaxes the requirements on the output voltage swings/fluctuations from a front-end at high data rates. The dynamic range can easily be increased to 16 dB with a burst-mode amplitude recovery circuit as in [4]. At the end, Table 5.1 summarizes the overall performance of our receiver.

| | Bit rate (Gbs) | Sensitivity (dBm) | Preamble (bits) | CIDs |
|---|---|---|---|---|
| This work | 5 | -18 | 0 | > 1000 |
| S. Nishihara | 10 | -19 | 1000 | NA |
| GPON [ITU] | 1.25 | -23 | 44 | > 72 |

**Table 5.1** Summary of the BM-CPA performance [ITU-T Recommendation G.984.2.]

## 5.10 Summary

We have demonstrated a 5 Gb/s BM-CPA based on a 2×Oversampling LO and a phase picking algorithm. We performed PLR measurements and quantified it as a function of phase steps between packets, signal power, and CID immunity. We also assessed the trade-offs in power penalty and preamble length. The receiver achieves a PLR $< 10^{-6}$ and a BER $< 10^{-10}$ while featuring instantaneous (0-bit) phase acquisition for any phase step between packets, a sensitivity of -18 dBm, and supporting more than 1000 CIDs, with a power penalty of 1-dB. Our CDR-free BMRx greatly reduces the complexity of electronics, providing a cost-effective solution for GPON receivers.

# Bibliography

[1] M. Y. Hsiao, "A class of Optimal Minimum Odd-weight-column SEC DED Codes," *IBM J. Res. Develop.*, pp. 395–401, 1970.

[2] S. . ITU-T G.984.2, *Gigabit-Capable Passive Optical Networks (G-PON): Physical Media Dependent (PMD) Layer Specification*, March 2003.

[3] P. Ossieur, X. Z. Qiu, and J. Vandewege, "Sensitivity Penalty Calculation for Burst-Mode Receivers Using Avalanche Photodiodes," *IEEE J. Lightw. Technol.*, vol. 21, pp. 2565–2575, 2003.

[4] S. Nishihara, S. Kimura, T. Yoshida, M. Nakamura, J. Terada, K. Nishimura, K. Kishine, K. Kato, Y. Ohtomo, N. Yoshimoto, T. Imai, and M. Tsubokawa, "A Burst-Mode 3R Receiver for 10-Gbit/s PON Systems With High Sensitivity, Wide Dynamic Range, and Fast Response," *IEEE J. Lightw. Technol.*, vol. 26, pp. 99–107, 2008.

# Chapter 6

## Summary and Other Solutions for Burst-Mode Clock Phase Aligner

In this thesis, a clock phase aligner design is proposed to deal with burst-mode traffic in point-to-multipoint networks such as PONs. The nature of P2MP networks introduces optical path delays which inherently cause the data packets to undergo amplitude, phase, and frequency variations - burst-mode traffic which creates unique challenges for the design and testing of optical receivers.

The proposed solution uses a 2×Oversampling algorithm to process 5 Gb/s incoming data with commercially available electronic components and a custom-programmed field programmable gate array (FPGA) board to achieve instantaneous phase alignment with no need for preamble bits. It is the first time that instantaneous phase alignment is demonstrated at 5 Gb/s. This solution is relatively simple compared to other burst-mode phase alignment methods at the cost of higher electronics processing speed. We also attempt to implement error coding schemes in the receiver design for better system performance. SEC-DED codes lead to slightly better bit error rate (BER) and packet loss rate (PLR) performances but is insufficient in noisy channels with bursty errors.

In this chapter, we propose directions for future research that could be derived from the work presented in this thesis.

## 6.1 Reed-Solomon (255, 239) Implementation

Reed-Solomon (RS) codes are expected to significantly improve the performance of the system discussed in the thesis, due to its ability to correct bursty errors. In the Appendix A, an RS decoder implementation is introduced. However, the implemented RS decoder needs an input of an 8-bit symbol, which is not suitable for the current receiver design. This is because the data on the FPGA board has been deserialized and processed 64 bits at a time to relax the speed requirement on the digital logic. In order to pass 8 bits of data at a time to the RS decoder, serialization is needed. It inevitably increases the data speed by 8 times, which can not be handled by the currently available FPGA board. Therefore, more efforts can be made towards implementing a programmable RS decoder whose input width can be varied according to specific applications.

### 6.1.1 Advantages

RS codes are capable of correcting a block of data, regardless of the number of erroneous bits in the block. Therefore, it is favored to correct burst error in bursty channels with reliable and predictable BERs.

### 6.1.2 Disadvantages

The trade-offs of the error correction capability of RS codes are its complexity in design, consumption in hardware area, and latency in process. In order to correct errors in a codeword, all 255 symbols in the codeword need to be received beforehand. Therefore, all received data need to be stored and buffered until error correction is performed, which leads to consumption in both hardware and latency.

## 6.2 nXoversampling Implementation, where $n > 2$

The 2xOversampling algorithm provides two clock rising edges that sample each incoming data bit twice every bit period. At least one of these two clock edges (odd and even) falls within the data eye opening, such that the data can be sampled properly regardless of the phase steps between two packets. This algorithm provides two sampled data streams to be chosen from; however, there are times when neither of the two clock rising edges is at the optimal sampling point of the data eye. In these cases, the system has a lower jitter tolerance and the performance is degraded. nxOversampling, where $n > 2$, can provide more clock rising edges within each data bit, such that $n$ samples of the incoming data are collected to be chosen for further processing.

### 6.2.1 Advantages

The advantage of this method is that it increases the sampling accuracy. More sampling clock edges in one data eye opening means a higher probability of one clock rising edge falling at the optimal sampling point. The most accurately sampled data stream can be selected for its optimal jitter tolerance.

### 6.2.2 Disadvantages

A higher sampling rate poses a higher requirement on hardware speed and area consumption. To implement this high-rate oversampling algorithm, a multi-rate clock and data recovery (CDR) circuit running at the specific oversampled rate is needed. More deserialization is also necessary in order to lower the data speed to a rate that can be processed by the FPGA board. Since more data samples are brought onto the FPGA board to select the optimal data path, more hardware is consumed to store the data values. These disadvantages are more severe when the sampling rate is higher. Therefore, it should be carefully considered when choosing the value of $n$.

## 6.3 Clock Tapped Delay Sampling Technique

This technique is very similar to the correlation algorithm but in PON application. Versions of delayed clock signal are used to sample the incoming data. Unlike the correlation algorithm, which uses the first three bits to identify the synchronized version of the sampling clock, this technique can directly make use of the delimiter pattern at the beginning of each burst packet to identify the synchronized clock signal, resulting in instantaneous phase alignment.

### 6.3.1 Advantages

Similarly to the correlation algorithm, two advantages of this technique are low latency and high jitter rejection. Since only the delimiter bits are used for identification and no extra bits are needed for phase alignment, the latency is reduced to the minimum. In this technique, jitter accumulated at the input data is rejected while data is being retimed by a clock signal generated by a local oscillator; therefore, jitter rejection is realized. One significant advantage of this technique compared to the nxOversampling algorithms is its relaxed speed requirement on electronics. In this technique, multiple clock sampling edges within one data eye are provided by delaying one clock signal running at the same speed of the incoming data instead of generating a higher rate clock signal. Therefore, no high speed multi-rate CDR is required.

### 6.3.2 Disadvantages

In this technique, only a finite number of clock signals can be practically generated. Thus, there is no guarantee that the selected clock signal samples at the optimal point. In this case, the signal-to-noise ratio and therefore the BER performance of the CDR output can vary from burst to burst depending on the relative phase difference between incoming data and generated clock signals. This technique can not be realized with the hardware currently used for the design presented in this thesis. A high-speed analog to a digital converter is needed to convert the incoming analog data signal to multiple digital levels, in order for it to be sampled by delayed versions of clock signals. However, a high-speed analog to a digital converter is difficult to design and expensive to purchase.

## 6.4 ASIC Design

The current burst-mode clock phase aligner is built from commercially available off-the-shelf components and therefore rather bulky. If this system is to be deployed in passive optical networks in the near future, then it is important to scale the design down to an application-specific integrated circuit (ASIC).

### 6.4.1 Advantages

The two main advantages of designing an ASIC are lower power consumption and smaller physical size. A major financial restraint on consumers is the high cost associated with the level of power consumption. Designing this receiver on a chip will reduce the power consumption of the system. A smaller physical size is easy for device integration in PONs.

### 6.4.2 Disadvantages

The only disadvantage of scaling down this receiver design onto a chip is the inflexibility of any further change in the design. Unlike designing on an FPGA board, chip design needs to go through manufacturing cycles before being tested on the actual device. Once a chip is manufactured, the circuit design can not be easily altered. Therefore, extra care must be taken during the design phase of the ASIC.

## 6.5 Summary

Optical multiaccess networks, and more specifically PONs have opened a new era for telecommunications. The potential bandwidth PONs can bring to each user through deploying fiber-to-the-home/building/curb (FTTx) has attracted much research attention to further develop this promising technology. This thesis targets one aspect of PONs and presents an efficient solution to the problem of processing high-speed burst data at the receiving end of the network. With combined research efforts, one can expect PONs to emerge as a fast and reliable solution to the growing needs for higher bandwidth, and to dominate the next generation of telecommunication technologies.

# Appendix A

# Reed-Solomon Decoder Implementation

## A.1 Reed-Solomon Decoder Implementation

Hamming and SEC-DED have single error correction capability, but in the cases with multiple errors in a codeword, they can not provide any coding gain. Therefore, they are not efficient in communication systems that are noisy with bursts of errors. In those systems, the Reed-Solomon (RS) code is favored for its capacity to correct a block of data regardless of the number of errorneous bits in the block. However, the design and implementation of the RS code are much more complicated than the two previous coding schemes.

### A.1.1 Decoding Steps

Decoding the RS code involves determining the positions and magnitudes of the errors in the received polynomial $r(X)$ which can be written as:

$$r(X) = r_{n-1}X^{n-1} + r_{n-2}X^{n-2} + \cdots + r_1 X + r_0 \tag{A.1}$$

Positions $P(X)$ are represented by powers of $X$, in the received polynomials, whose coefficients are the corrupted symbols in a codeword. Magnitudes $e(X)$ are symbols to be added to the corrupted symbol to obtain the original symbol. Locations and magnitudes constitute the so-called error polynomial.

After going through a noisy transmission channel, the received data polynomial $r(x)$ can be represented as $r(X) = c(x) + \text{error}(x)$ where $c(x)$ is the original encoded data polynomial and error($x$) is the error polynomial.

The following steps show the decoding procedure and hardware implementations to determine error($x$) to recover $c(x)$:

***Syndrome Calculation***

The syndrome polynomial $S(X)$ is obtained by evaluating the received polynomial $r(X)$ at the $2t$ roots of the generator polynomial $g(X)$. Since each codeword polynomial is obtained by multiplying a codeword with $g(X)$, every codeword polynomial must have the same $2t$ consecutive powers of $\alpha$ as roots:

$$S(X) = \sum_{i=0}^{2t-1} r(\alpha^i) \tag{A.2}$$

$$g(x) = \prod_{j=1}^{2t} (x - \alpha^j) \tag{A.3}$$

***Key Equation Calculation***

To find the location and magnitude of each error, two other polynomials are needed: the error locator polynomial $\Lambda(X)$, and the error magnitude polynomial $\Omega(X)$. $\Lambda(X)$ is later used in Chien search algorithm to determine error positions, while $\Omega(X)$ is used in Forney algorithm to evaluate the magnitude of each error. The key equation describes the relationship between the three polynomials: $S(X)$, $\Lambda(X)$, and $\Omega(X)$. $S(X)$ is obtained in the previous step, and $\Lambda(X)$ can be evaluated through Berlekamp-Massey (BM) algorithm, shown in Figure A.1. The expression of $\Lambda(X)$ can be written as:

$$\Lambda(X) = \prod_{i=1}^{e} (1 - X_i x) = 1 + \lambda_1 x + \lambda_2 x^2 + \cdots + \lambda_e x^e \tag{A.4}$$

where $e$ is the number of errors.

Therefore, by knowing $S(X)$, $\Lambda(X)$, and the key equation, $\Omega(X)$ can be derived to

$$\Omega(X) = \sum_{i=1}^{e} Y_i X_i \prod_{j=1}^{e} (1 - X_j x) = \omega_0 + \omega_1 x + \omega_2 x^2 + \cdots + \omega_{e-1} x^{e-1} \tag{A.5}$$

***Error Position Evaluation***

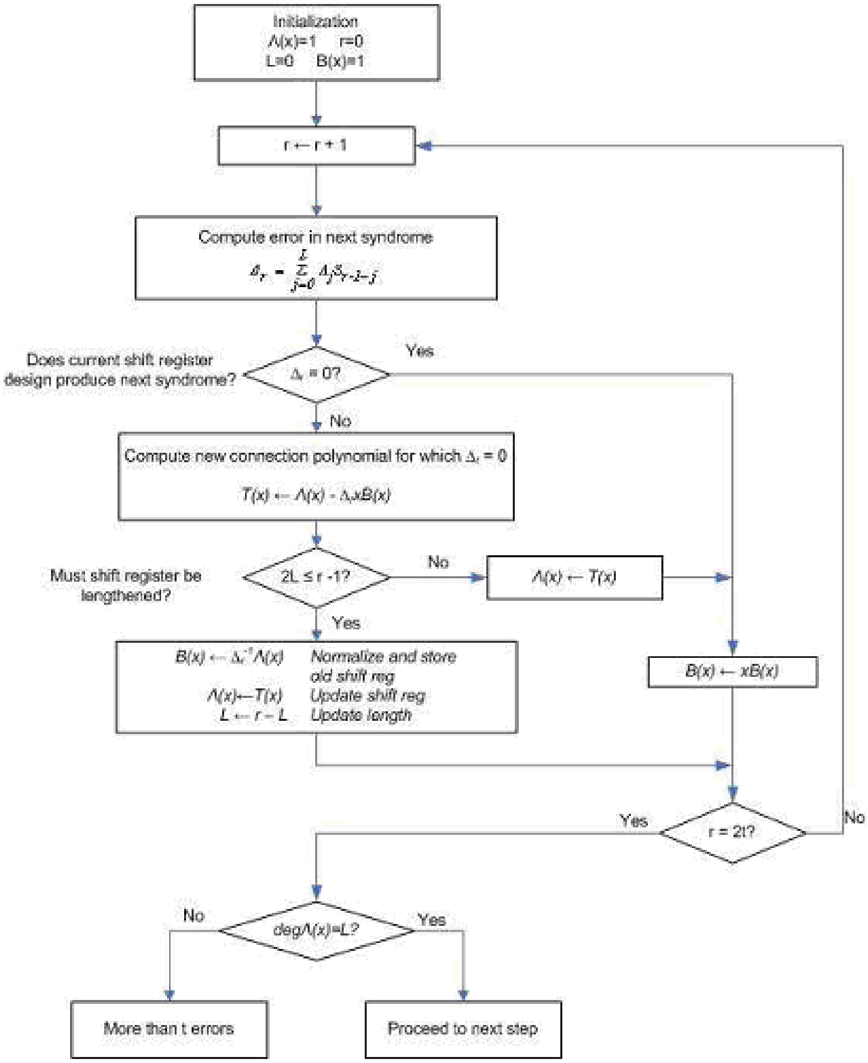

Initialization
Λ(x)=1 r=0
L=0 B(x)=1
r ← r + 1
Compute error in next syndrome
$\Delta_r = \sum_{j=0}^{L} \Lambda_j S_{r-1-j}$
Does current shift register design produce next syndrome?
Δr = 0?
Yes
No
Compute new connection polynomial for which Δr = 0
T(x) ← Λ(x) - ΔrxB(x)
Must shift register be lengthened?
2L ≤ r -1?
No
Λ(x) ← T(x)
Yes
B(x) ← Δr⁻¹Λ(x)
Normalize and store old shift reg
Λ(x)←T(x)
Update shift reg
L ← r – L
Update length
B(x) ← xB(x)
Yes
r = 2t?
No
No
degΛ(x)=L?
Yes
More than t errors
Proceed to next step

**Figure A.1** The Berlekamp-Massey algorithm; [1].

The Chien search algorithm takes the error locator polynomial $\Lambda(X)$ as the input to calculate the positions of errors. The algorithm evaluates $\Lambda(X)$ at each root of $g(X)$, $\alpha^j$ for $0 \leqslant j \leqslant 2t-1$. If $\Lambda(j) = 0$, the location of the error is $c$, where $c$ is derived based on $\alpha^{-j} = \alpha^c$ in the Galois Field.

***Error Magnitude Evaluation***

Rather than performing an entire inverse Fourier transform to find the time domain errors, we only need to find values coincident with the roots in the magnitude polynomial

$\Omega(X)$. It turns out, however, that is unnecessary. Forney algorithm allows direct calculation of the error patterns and has the following form [2]:

$$e_{i_k} = \frac{-X_k \Omega'(X_k^{-1})}{\Lambda(X_k^{-1})}, \quad \text{where } \Lambda'(X) \text{ is the derivative of } \Lambda(X) \tag{A.6}$$

***Error Correction***

By combining the error magnitudes and locations, an error polynomial is constructed to be added to the transmitted codeword to retrieve the original codeword.

### A.1.2 Hardware Implementation

The hardware architecture of the decoder is essentially based on adapting the equations outlined previously. Since RS (255, 239) code is being implemented, each symbol is 8 bits in length, and each codeword contains 255 symbols, out of which, 239 symbols are data and the rest are parity check bits.

The block diagram of the system implementation is shown in Figure A.2. From the received data, the syndrome calculator generates the syndrome vector $S(X)$ which is an input to both key equation solver and error magnitude polynomial calculator. The key equation solver, which is implemented based on the BM algorithm, evaluates the error locator polynomial $\Lambda(X)$ upon receiving $S(X)$. The output, $\Lambda(X)$, is taken as an input to the error locator block which takes use of Chien search to find both error positions $P(X)$ and the zero polynomial $X(X)$. Error magnitude polynomial calculator needs both $S(X)$ and $\Lambda(X)$ to evaluate the error magnitude polynomial $\Omega(X)$. Eventually, polynomials $X(X)$, $\Lambda(X)$, and $\Omega(X)$ are combined through a Forney algorithm to give the error magnitudes $e(X)$. $P(X)$ and $e(X)$ form the error polynomial which is added to the received codeword for error correction. Each bolck in Figure A.2 will be explained in details in the following section.

***Syndrome Calculator***

Syndrome equation calculation involves multiplication of the incoming symbol with powers of $\alpha$, and summations to compute one $S_j$. Parallel implementation requires $2t$ (16, in this specific case) units, as shown in Figure A.3. In this design, the syndrome generator uses constant multipliers, derived from the generator polynomial during encoding process, to optimize hardware consumption and calculation speed. When the start signal is high, 8-bit symbols are taken in one at a time until the counter reaches 255, which is the count

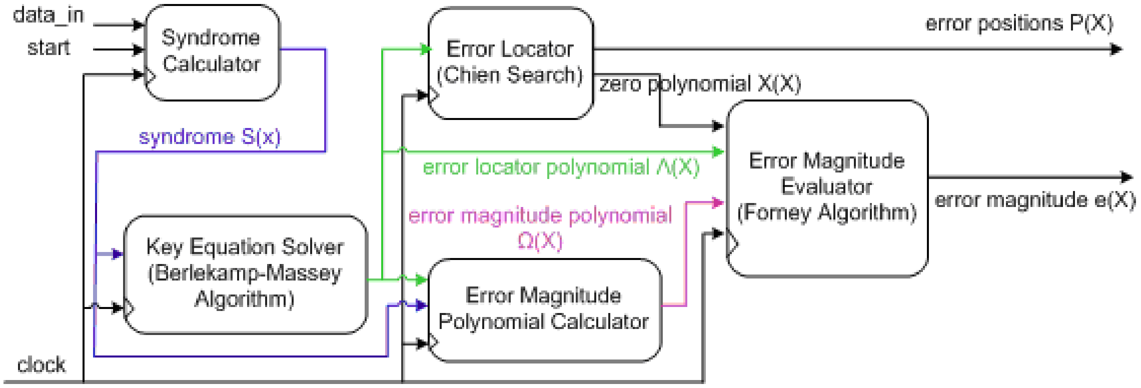

data_in
start
clock
Syndrome Calculator
syndrome S(x)
Key Equation Solver (Berlekamp-Massey Algorithm)
Error Locator (Chien Search)
error positions P(X)
zero polynomial X(X)
error locator polynomial Λ(X)
error magnitude polynomial Ω(X)
Error Magnitude Polynomial Calculator
Error Magnitude Evaluator (Forney Algorithm)
error magnitude e(X)

**Figure A.2** System level block diagram of a RS decoder.

of a codeword of 255 symbols.

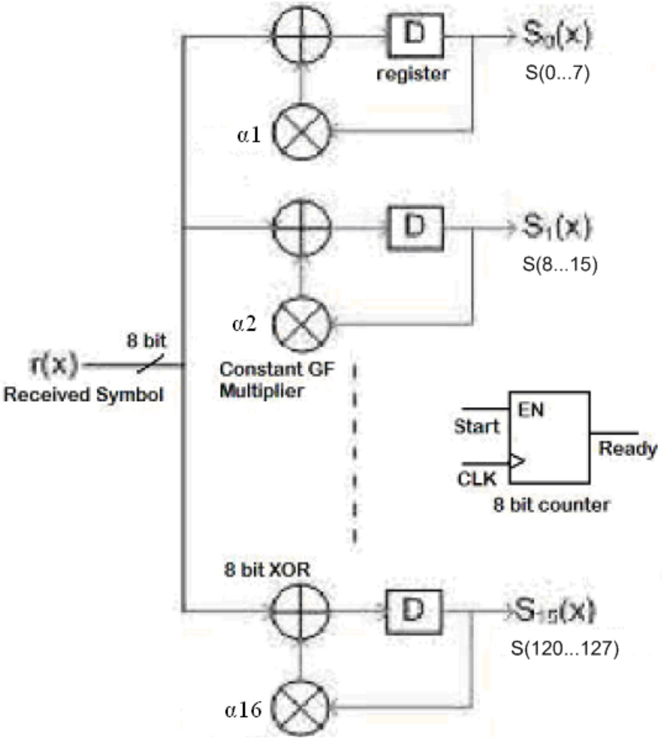


**Figure A.3** Hardware implementation of a syndrome generator.

In the syndrome calculator block shown in Figure A.4, two syndrome generators are used in parallel. This design allows receiving data at a constant rate without extra buffer

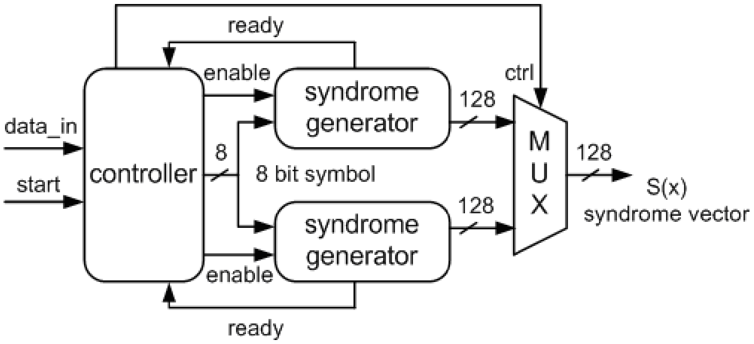


**Figure A.4** Block diagram of a syndrome calculator block with parallel syndrome generators.

delays. One generator is activated to receive data when the other one is outputting syndrome vectors and resetting registers. The control block is used to coordinate the two syndrome generators.

### *Key Equation Solver*

The challenge of a key equation solver is the BM algorithm implementation. A finite state machine shown in Figure A.5 is derived from the process flow algorithm, shown in Figure A.1. The detailed derivation is explained in [3]. In our design, two dummy states are inserted in the state machine to ensure that each data path goes through the same number of clock cycles for system synchronization.

### *The Error Locator - Chien Search*

The Chien search algorithm takes the error locator polynomial $\Lambda(X)$ calculated in the key equation solver as the input, and generates two polynomials that are used in identifying the errors in $r(x)$. The first one is generally referred to as the zero polynomial $X(X)$, which is applied to the Forney algorithm for determination of the error magnitudes. The other one is the position polynomial $P(X)$, which indicates the position of the erroneous symbols in $r(X)$.

The Chien search uses a root detection block, as shown in Figure A.6, to evaluate the following function:

$$X_i = \Lambda_0 + \sum_{i=0}^{\nu} \Lambda_j(\alpha^i)^j, \quad \text{for } i = 1 \text{ to } 255,\ \nu \text{ degree of } \Lambda(X). \tag{A.7}$$

In hardware, this is implemented with multiple weighted sum blocks, in combination with a

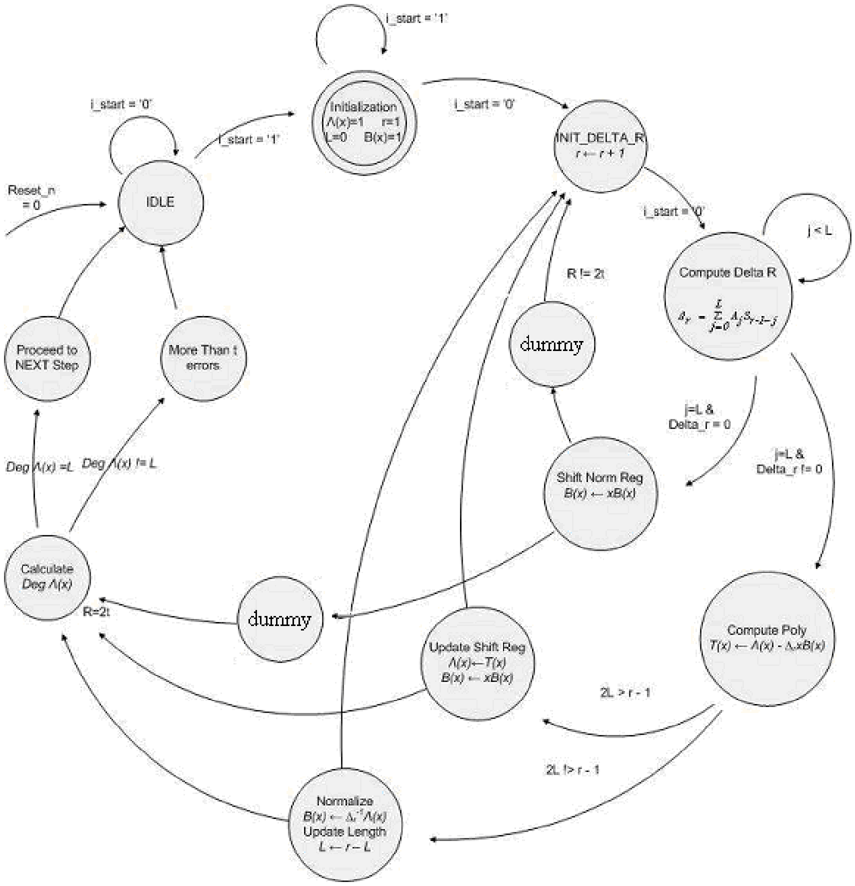


**Figure A.5** Berlekamp-Massey State Diagram [1]

GF adder and a zero detection circuitry. As shown in Figure A.6, the lowest order coefficient of $\Lambda(0)$ is forwarded directly to the adder. Each of the next higher order coefficients $\Lambda(X)$, are forwarded to a corresponding weighted sum block along with a constant $\alpha^i$. Each weight block is constructed with a MUX, an 8-bit register, and an 8-bit XOR. The MUX in each block first selects the coefficient of $\Lambda(X)$ to store in a register; the output of the next register (every weighted sum block, and $\Lambda(0)$) is passed to the GF adder for determination of the sum and thus possible detection of a root. The sum goes through a zero detection circuit, which drives the line ZRO high if the sum is equal to zero, which occurs when the current GF symbol $\alpha^i$ is a root of the zero polynomial $X(X)$. For the second and subsequent symbols in the same codeword, the current content of the register in the weight blocks are multiplied by $\alpha^i$ (integer $0 \leqslant i \leqslant t$). For example, in the third iteration, the first weighted sum block will multiply $\alpha^1$ with $\Lambda(1)\alpha^1$, and the register stores $\Lambda(1)\alpha^2$. This process iterates for all the 255 symbols in a codeword, so that each symbol is interrogated to determine whether it is a root.

Line ZRO is used to enable two sets of registers. The first set corresponds to $X(X)$, to store the values of each root $X$. An index counter keeps track of the iteration $i$, which is an input of a GF exponential circuit. This circuit is simply a LUT generating a signal MAG. It is the magnitude of the root being detected, at each corresponding iteration count, so that the root value is stored in the first available register at the detection of the root. At the same time, the other set of registers $P(X)$'s store the corresponding error positions.

***Error Magnitude Polynomial Calculator***

The error magnitude polynomial can be rearranged as the following:

$$\Omega(X) = S1+(S2+\Lambda(1)S1)X+(S3+\Lambda(1)S2+\Lambda(2)S1)X^2+\cdots+(S8+\Lambda(1)S7+\cdots+\Lambda(7))X^7 \tag{A.8}$$

The hardware to compute $\Omega(X)$ is shown in Figure A.7. The first 64 bits of the syndrome vector are loaded to eight 8-bit shift registers. The enable controller synchronizes the computation. At the first clock cycle, the value $S1$ is outputted as the first syndrome polynomial coefficient. During the second clock cycle, the first GF multiplier is enabled, so that $S1\Lambda(1)$ is evaluated and added to $S2$ to form the second coefficient. The third coefficient is calculated by first shifting the shift register to the right and enabling one more GF multiplier. The same procedure is repeated for six more iterations until all coefficients of $\Omega(X)$ are evaluated.

***Error Magnitude Evaluator - Hardware Implementation of Forney Algorithm***

Figure A.8 shows the hardware implementation of the Forney evaluator which evaluates the error values according to the equation $e_{i_k} = \frac{-X_k\Omega(X_k^{-1})}{\Lambda'(X_k^{-1})}$. The coefficients of each zero (root) stored in the zero polynomial $X(X)$ in the Chien search block are input one at a time through a pipeline structure. The upper branch of Figure A.8 evaluates the numerator of the equation, while the lower branch evaluates the denominator. Evaluation of the derivative of $\Lambda(X)$ is equivalent to retaining the coefficients $\Lambda(8)$, $\Lambda(6)$, $\Lambda(4)$, and $\Lambda(2)$ and evaluating this polynomial at the roots. Since division (inversion) is area inefficient to implement in hardware, a look up table is used to generate $\Lambda'(X_k^{-1})$, which is multiplied to the numerator in the last stage.

The RS decoder implemented here takes one 8-bit symbol at a time as the input. All 255 symbols (239 information symbols + 16 parity symbols) in one codeword need to be input sequentially to be eavaluated for error detection and correction. The deseriablized

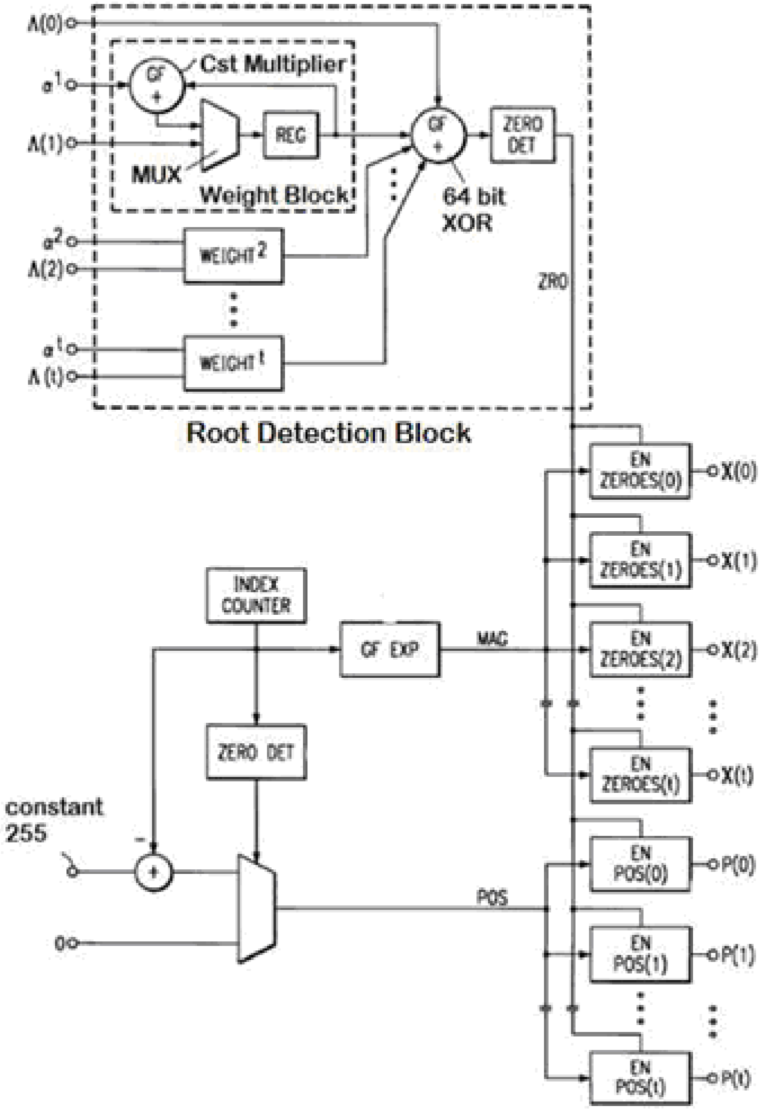


**Figure A.6** Chien search block diagram [4].

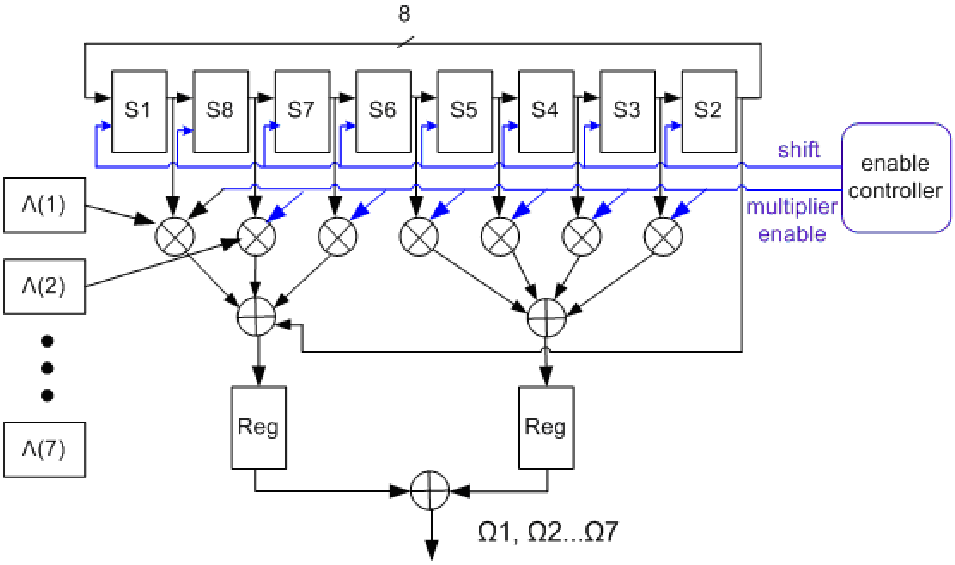


**Figure A.7** Hardware implementation of error magnitude polynomial computation.

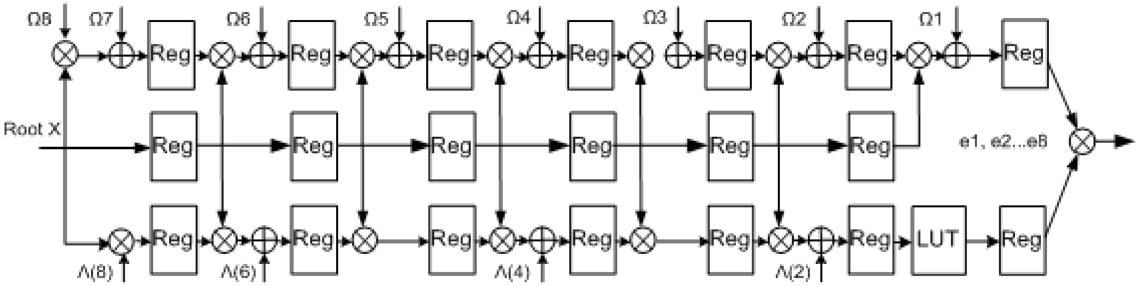


**Figure A.8** Hardware Implementation of Forney Algorithm [5].

data on the FPGA board are processed as 64 bits in parallel at 78.125 Mb/s. To serialize the data to a width of 8-bit, the data speed is inevitably increased to 78.125 Mb/s × 8 = 625 Mb/s. However, the digital clock manager on the avaliable FPGA board can not provide a clock signal at this speed and a period of $\frac{1}{625}$ Mb/s = 1.6 ns is not enough for the sequences of XOR calculations in the RS decoder. It is due to the hardware speed limitation that the RS decoder is not tested in the BM-CPA presented in this thesis.

# Bibliography

[1] K. C. C. Wai and S. J. Yang, *Field Programmable Gate Array implementation of Reed-Solomon Code, RS(255,239)*.

[2] A. Houghton, *Error Coding for Engineers*. Springer, 2001.

[3] S. S. Shah, S. Yaqub, and F. Suleman, "Self-correcting Codes Conquer Noise Part 2: Reed-Solomon Codecs," *EDN Magazine*, pp. 107–120, 2001.

[4] T. D. Wolf, "Efficient Hardware Implementation of Chien Search in Reed Solomon Decoding," *United States patent*, vol. US, no. 6,209,114 B1, 2001.

[5] R. T. Chien, "Cyclic Decoding Procedure for the Bose-Chandhuri-Hocquenghem Codes," *IEEE Transactions on Information Theory*, vol. IT-10, pp. 357–363, 1964.