Open access • Proceedings Article • DOI:10.1364/ACPC.2013.AF4C.6

# Application of Beyond Bound Decoding for High Speed Optical Communications



Bomin Li, Knud J. Larsen, Juan Jose Vegas Olmos, Darko Zibar ...+1 more authors

**Institutions:** Technical University of Denmark

**Published on:** 12 Nov 2013

**Topics:** Optical communications repeater, Decoding methods, Coding gain, Bit error rate and Optical communication



# Application of Beyond Bound Decoding for High Speed Optical Communications

**Li, Bomin; Larsen, Knud J.; Vegas Olmos, Juan José; Zibar, Darko; Tafur Monroy, Idelfonso**

*Published in:*
ACP/IPOC 2013

*Publication date:*
2013

*Document Version*
Publisher's PDF, also known as Version of record

# Application of Beyond Bound Decoding for High Speed Optical Communications

**Bomin Li, Knud J. Larsen, Juan Jose Vegas Olmos, Darko Zibar and Idelfonso Tafur Monroy**
*Department of Photonics Engineering, Technical University of Denmark, Anker Engelunds Vej 1, 2800 Kgs. Lyngby, Denmark*
*boml@fotonik.dtu.dk*

**Abstract:** This paper studies the application of beyond bound decoding method for high speed optical communications. This hard-decision decoding method outperforms traditional minimum distance decoding method, with a total net coding gain of 10.36 dB.
**OCIS codes:** (060.2330) Fiber Optics Communications; (060.4510) Optical Communications

## 1. Introduction

Optical links are currently experiencing a migration towards higher bitrates. For example, Optical access networks are now operating at 40G regimes [1], while short range systems are being developed for 400G regimes [2]. This incremental improvement in the capacity of the links is changing the paradigm of supported bit error rate (BER), which is moving from a $10^{-12}$ to a $10^{-15}$ levels [2]. This increment in the benchmark of supported BER requires improving the system performance in high speed optical fibers, from the class optics, the fiber itself and the digital signal processing algorithms. Forward Error Correction (FEC) plays a key role in this quest since it can provide extra gain to alleviate the technology pressure on the optical side of the system while supporting the signal quality levels. During the past several years, FEC has been integrated in various systems to obtain the desired performance. Foresight examples are studies of WDM Performance and Multiple-Path Interference Tolerance [3], and High-Speed 1550 nm VCSEL [4]. Overall, it becomes utmost important to come up with FEC codes of higher gain and faster processing capability.

In this paper we study beyond bound decoding method, employing the same code as proposed in [5]. Without changing the encoding structure, 0.36 dB more net coding gain (NCG) is achieved, with a total NCG of 10.36 dB. The pre-FEC BER of this 20% overhead hard decision code is $1.4 \cdot 10^{-2}$ and it is above the theoretical threshold obtained by applying minimum distance decoding method. The solution proposed in this paper meets the technical requirements in terms of latency, complexity and gain for next generation optical networks.

## 2. Principle of Beyond Bound Decoding

Product code with shortened BCH components for high speed optical communications was first studied in [5]. The component code is a shortened BCH code of length $n \leq 2 \cdot m - 1$ as shown in Figure 1. The number of information bits in a row/column is denoted $k$. By inserting $2 \cdot m - 1 - n$ leading zeros, each row/column becomes a BCH codeword.

At the decoding part, iterative decoding is employed for the product code. One iteration consists of decoding a product code horizontally once and vertically once. Each row/column is decoded by a component decoding method. In [5], the component decoding method is minimum distance decoding. In this paper, the component decoding method is beyond bound decoding. All returned error bits are checked to be within $n$ bits of a received word or not.

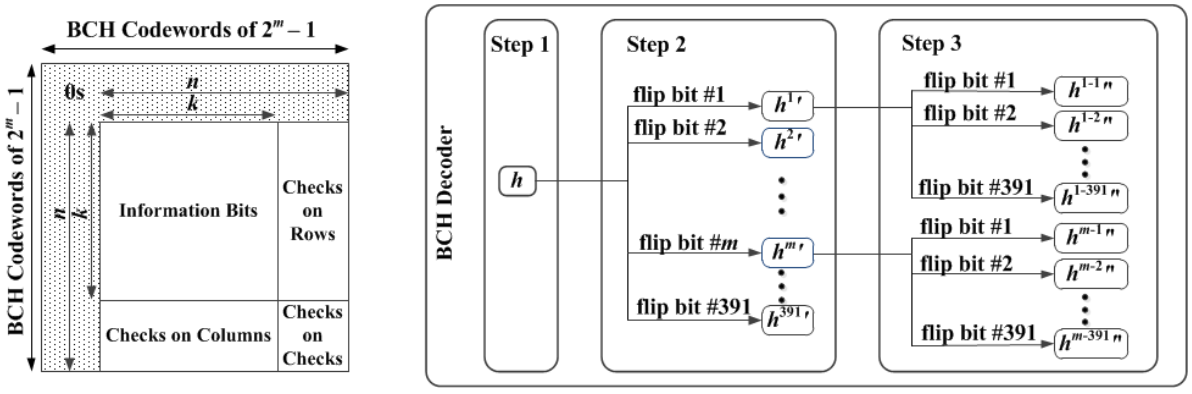


**Fig. 1:** Code Construction and Circuit Implementation Investigation

**Tab. 1:** Component Decoding Error by Beyond Bound Decoding

| Error Weight ($t$) | $h_0$ | $h_1$ | $h_2$ | $h_3$ | Decoding Success (%) | Decoding Error (%) | Comments on Decoding Error |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 | 100 | 0 | - |
| 1 | 1 | 0 | 1 | 1 | 100 | 0 | - |
| 2 | 0 | 0 | 0 | 1 | 100 | 0 | - |
| 3 | 1 | x | 0 | 0 | 100 | 0 | - |
| 4 | 0 | x | x | 0 | 88.8 | - | 11.2% not being decoded due to multiple candidate four-error patterns |
| 5,7,9,… | 1 | x | 0 | 0 | 0 | < 0.12 | - |
| | | others | | | 0 | - | 100% not being decoded |
| 6,8,10,… | 0 | 0 | 0 | 1 | 0 | < 0.12 | two-error pattern |
| | | x | x | 0 | 0 | 10.1 | unique four-error pattern |
| | | others | | | 0 | - | 100% not being decoded |

If not, decoding error is reported. It is called error position check later in this paper. To keep the encoding part untouched, the code of the same parameters are used in this paper as in [5], that is, $n = 391$ and $m = 11$.

The component has minimum distance of eight. With minimum distance decoding, it only corrects up to three errors. Beyond bound decoding, which is studied in this paper, aims to correct up to four errors. The method is explained in [6]. The basic idea of decoding is to change the value of each bit and check whether the weight of errors is reduced. If so, the value of this bit should be changed, otherwise the original value is kept. A vector $h = [h_0\ h_1\ h_2\ h_3]$ is built to help with error weight change detection. The value $h_0$ is an even parity check bit while $h_1$, $h_2$ and $h_3$ are the opposite values of corresponding determinants of syndrome matrices. The construction of syndrome matrices follows the rule for BCH code. Table 1 shows the values of the vector $h$ for different weight error patterns.

Component decoding error is a practical issue in iterative decoding of a product code as introduced errors increase decoding iterations. These introduced errors may also possibly build a non-decodable error pattern together with existing errors. Therefore it is important to reduce component decoding error probability. An overview of component decoding error probability is shown in Table 1, where x means do-not-care.

Vectors $h$' and $h$'' are calculated to help finding the reduction of error weights. After first flipping the value of one bit, the vector $h$' is obtained in the same way $h$ is obtained. For example, flip the value of bit 1 and the calculated value of $h$' is denoted $h^1$', as shown in Figure 1. Similarly, the vector $h$'' is obtained after first flipping the values of two bits. For $\underline{t} \leq 3$, error correction is done by comparing $h$ and $h$'. If the error weight is reduced, a candidate error bit is found. Flip all bits one by one and all error bits are returned. All received words in this case are decoded successfully without decoding error. For $t = 4$, most of errors should be corrected by beyond bound decoding. If the error weight is reduced to two by comparing $h$ and $h$''. two candidate error bits are found. Update the received word and decode it to find the other two candidate bits as discussed for $t \leq 3$, which in turn builds a candidate pattern. A candidate pattern is wrong if it does not pass error position check. Notice that candidate four-error patterns are not unique. To find all candidate patterns, the combinations of any two bits should be considered in the stage of calculating $h$''. Multiple successful candidate patterns prevent a received word from being decoded. The probability of not being decoded can be theoretically calculated in this way. Flip one bit to build a five-error pattern and the decoding error probability is $0.116 \cdot 10^{-2}$ as calculated for the case of $t = 5$. As there are $n - 4$ positions that may return wrong candidate error patterns and each pattern contains four error bits, the probability of not being decoded is $0.116 \cdot 10^{-2} \cdot (n - 4)/4 = 11.2\%$. For $t \geq 5$ and $t$ is odd, the decoding error probability is explained in [5]. It can also be considered in the way of the probability that a random odd-number-error-pattern syndrome corresponds to a three-error pattern [8]. The value is $(1/(3!)) \cdot (n/(2^m-1))^3 = (1/(3!)) \cdot (391/(2^{11}-1))^3 = 0.116\%$. For $t \geq 6$ and $t$ is even, it could be wrongly decoded with either a two-error pattern or a four-error pattern. The decoding error probability of a two-error pattern can be analyzed similarly to the case of $t = 5$ and it is much smaller. The decoding error probability of a four-error pattern is the probability of finding one and only one successful four-error pattern. It is $(n/4) \cdot (0.116 \cdot 10^{-2}) \cdot (1-0.116 \cdot 10^{-2})^{(n-4)/4} = (391/4) \cdot (0.116 \cdot 10^{-2}) \cdot (1-0.116 \cdot 10^{-2})^{(391-4)/4} = 10.1\%$.

## 3. Simulation Result of the Product Code

The simulation results in Matlab are shown in Figure 2. Simulation results of 10 iterations with white Gaussian noise are shown for both beyond bound decoding method and minimum distance decoding method. The pre-FEC BER of the product code is around $1.4 \cdot 10^{-2}$ by employing beyond bound decoding method. And this translates to 10.36 dB NCG. As shown in Figure 2, there is 0.36 dB improvement in NCG compared to the value of employing minimum distance decoding method.

The theoretical threshold of applying minimum distance decoding method to this 20% overhead hard-decision FEC is $1.3 \cdot 10^{-2}$, aiming for an after-FEC BER of $10^{-15}$, as explained in [7]. The simulation shows a pre-FEC BER of around $1.1 \cdot 10^{-2}$ in [5], which falls below the threshold when minimum distance decoding method is employed. In this paper the pre-FEC BER is around $1.4 \cdot 10^{-2}$, which is above the threshold due to stronger decoding capability to of beyond bound decoding method.

The error floor of the product code in this paper is lower than the one in [5], as beyond bound decoding corrects more errors than minimum distance decoding does. The error floor is $< 10^{-15}$ for BER at $1.4 \cdot 10^{-2}$, also shown in Figure 2.

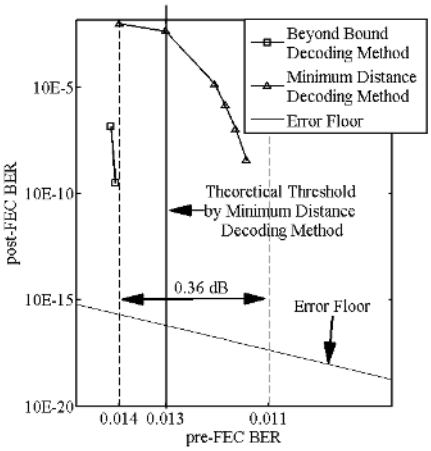


Fig. 2 Simulation Result

## 4. Circuit Implementation Investigation

The circuit implementation investigation is also shown in Figure 1, aiming for easy implementation and fast processing. Step 1 is to calculate the vector $h$. Syndrome storage costs 34 registers, so we estimate 100 registers in total and 8 clock cycles. Step 2 is to calculate $h$'. Actual values are $h^{1}$', $h^{2}$' …, $h^{391}$'. After bit $m$ ($1 \le m \le n$) is flipped, the obtained $h$' is denoted $h^{m}$'. A flag is inserted if it shows an error weight reduction. If all flags indicate $q$ ($0 < q < 3$) errors and there are $q + 1$ flags, the $q + 1$ flags associated positions reflect error bits. For example, suppose the errors are at bits $r$, $s$ and $u$. Three flags are inserted after checking $h$', at the positions of $h^{r}$', $h^{s}$' and $h^{u}$', all indicating two errors. At this step, up to three errors will be corrected. We estimate around 10 registers and 8 clock cycles for each $h^{m}$'. And 391 copies are required. Step 3 is designed to help finding four-error pattern. It calculates the value of $h$''. Actual values are $h^{1\text{-}1}$'', $h^{2\text{-}1}$'' …, $h^{391\text{-}1}$'', $h^{1\text{-}2}$'', …, $h^{391\text{-}2}$'', …, $h^{1\text{-}391}$'', … and $h^{391\text{-}391}$''. Here $h^{m\text{-}p}$'' ($1 \le m \le n$, $1 \le p \le n$) is the vector $h$'' of $h^{m}$' by flipping bit $p$ in step 3. A flag is inserted if the error weight is reduced to two after flipping bit $m$ and bit $p$. For bit $m$ in step 2, if the number of associated flags in step 3 is three, $m$ is an error bit. Otherwise bit $m$ is left unchanged. The check of flags performs error position check. For example, suppose the errors are at bits $r$, $s$, $u$ and $v$, each bit has three assoiciated flags, that is, flags $h^{r\text{-}s}$'', $h^{r\text{-}u}$'', $h^{r\text{-}v}$'' for bit $r$, $h^{s\text{-}r}$'', $h^{s\text{-}u}$'', $h^{s\text{-}v}$'' for bit $s$, $h^{u\text{-}r}$'', $h^{u\text{-}s}$'', $h^{u\text{-}v}$'' for bit $u$ and $h^{v\text{-}r}$'', $h^{v\text{-}s}$'', $h^{v\text{-}u}$'' for bit $v$. We estimate 1 register and 1 clock cycle to calculate and store the final value of error weight indication. Again 391 copies are required. In total the cost for a component decoder is $100 + (10 + 1) \cdot 391 \approx 4401$ register bits and 17 clock cycles. A product code decoder includes 391 copies of a component decoder as well as an interleaver. This requires $391 \cdot 4401 + 3912 = 1.85$ M register bits. Considering 10 iterations of a product code decoding, it takes $10 \cdot 2 \cdot (17 + 1) = 360$ clock cycles. The implementation is fast enough for FEC decoding in 100 Gb/s optical transmissions in an FPGA with a 250 MHz system clock.

## 5. Conclusion

The application of beyond bound decoding is studied in this paper. A high NCG of 10.36 dB for a product code is obtained in the simulation while the circuit implementation investigation shows the fast processing capability of the code. Meanwhile it keeps the encoding part unchanged. These features meet the current development guides in high speed optical communication links, especially in optical access systems where solutions to reduce the complexity of the optical side are preferred, and in high capacity 400G links, where FEC codes provide more gain to support stricter requirements on BER.

## 6. References

[1] Y. Luo *et al*., "Time- and wavelength-division multiplexed passive optical network (TWDM-PON) for next-generation PON stage 2 (NG-PON2)," in IEEE J. Lightwave Technol. 31, 4, pp. 587-593 (2013).

[2] D. Ofelt, "The BER objective for 400GE," IEEE 802.3 400 Gb/s Ethernet Study Group, http://www.ieee802.org/3/400GSG/public/13_05/index.shtml

[3] L.E.Nelson *et al,*, "WDM performance and multiple-path interferen ce tolerance of a real-time 120 Gbps Pol-Mux QPSK transceiver with soft decision FEC," in OFC'12, NThll.5(2012).

[4] R. Rodes *et al.*, "High-Speed 1550 nm VCSEL data transmission link employing 25 GBd 4-PAM modulation and hard decision forward error correction," in J.Lightwave Technol 31, 4, pp. 689-695(2013).

[5] B. Li *et al.*, "Over 10 dB net coding gain based on 20% overhead hard decision forward error correction in 100G optical communication systems," in ECOC'11, Tu.6.A.3(2011).

[6] S.W. Wei,"On Step-by-Step complete decoding Triple-Error-Correcting binary BCH codes," in Trans.Fundamentals. E889-A,11, pp.3360-3362(2006).

[7] J. Justesen, "Performance of product codes and related structures with iterated decoding," in IEEE Trans. Commun. 59,2, pp 407-415(2011).

[8] J. Justesen, T. Hoeholdt, "A course in error correcting codes," ISBN 3-03719-001-9.