FPGA implementation of distribution matching and dematching

FPGA IMPLEMENTATION OF DISTRIBUTION MATCHING AND DEMATCHING

Tsuyoshi Yoshida[1,2], Masashi Binkai[1], Shota Koshikawa[1], Shun Chikamori[1]*
Keisuke Matsuda[1], Naoki Suzuki[1], Magnus Karlsson[3], and Erik Agrell[3]

[1]*Information Technology R&D Centre, Mitsubishi Electric Corporation, Kamakura, Japan*
[2]*Graduate School of Engineering, Osaka University, Suita, Japan*
[3]*Fibre-Optic Communications Research Centre, Chalmers University of Technology, Gothenburg, Sweden*
**Yoshida.Tsuyoshi@ah.MitsubishiElectric.co.jp*

Keywords: Digital modulation, logic circuit, modulation coding, optical fibre communication.

Abstract

We implemented distribution matching and dematching (DM/invDM) on an FPGA in a back-to-back configuration, for the first time demonstrating very low BERs. At a post-FEC BER of 1.5×10^{-15}, a BER after invDM of 1.6×10^{-14} was measured.

1 Introduction

Probabilistic shaping (PS) is now a widely examined scheme in the optical fibre communication community for pushing the capacity closer to the Shannon limit [1,2]. A central approach has been reverse concatenation of distribution matching (DM) and forward error correction (FEC) coding. The reverse concatenation simplifies the digital signal processing (DSP) implementation of the inverse DM (invDM) a lot because its input interface becomes binary bits instead of high-resolution real values. Constant composition DM (CCDM) [3] proposed together with [1] shows excellent performance at quasi-infinite DM word lengths, but it is highly complex for hardware implementation due to the use of arithmetic coding, regardless of the simplification from reverse concatenation. There have been many relevant works on DM algorithms, for example enumerative sphere shaping [4], multiset-partition DM [5] and DM with shell mapping [6]. Look-up table (LUT) based DMs have been shown in fixed-length hierarchical DM [7], fixed-length optimum bit-level DM with binomial coefficients [8], and variable-length prefix-free code DM [9]. In case of limited source entropy live traffic, we have shown joint source–channel coding for simultaneous data compression and shaping to reduce the average symbol energy. This is based on sorting of the LUT contents in the DM with minimal added complexity [10]. This compression feature is somewhat similar to burst signalling in time-domain multiple access [11].

In reverse concatenation PS, we must consider the end-to-end performance. Post-invDM is therefore a more relevant place to quantify the performance than post-FEC. Typically, DM is a highly nonlinear operation, so that a small amount of errors at post-FEC may lead to a catastrophic error burst after the invDM. Our previously proposed hierarchical DM is partially decodable even in the presence of errors before invDM [7]. This is a unique feature among the DMs [4–9]. We predicted its performance at very low bit error rates (BERs) analytically and by simulations in [7]. However, very low BERs have never been measured for any DM scheme, while FEC was evaluated at BER at less than 10^{-10} via a field programmable gate array (FPGA) implementation [12].

In this work, we implemented the hierarchical DM on a FPGA, and demonstrate a very low BER close to 10^{-15} for the first time to the best of our knowledge. We exclude the FEC and soft-demapping function for simpler design and evaluation of the FPGA circuitry. Instead, DM to invDM back-to-back error insertion test was performed to quantify the system output performance at a given post-FEC BER. We also implemented a function to generate various source statistics and joint source–channel coding for data compression.

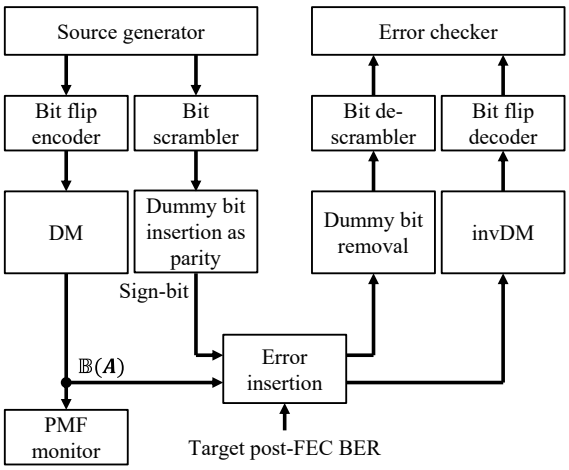

Fig. 1 Schematic of implemented functions on FPGA.

2 FPGA implementation

Figure 1 shows the block diagram of our implemented key functions on the FPGA (Xilinx® Virtex® Ultrascale+™ VCU118 XCVU9P). At the transmitter side (left hand side in Fig. 1), source bits having a given mark ratio $P_S(1)$ are generated in the circuit shown in Fig. 2 (explained below), and all incoming non-sign bits per block are flipped or not so that the so that the desired distribution between '0' (spaces) and '1'

(marks) is obtained. One bit per block is allocated for the recognition of the flipped ('1') or not ('0'), which will be used on the receiver side to recover the original bits. The DM consists of many hierarchical LUTs, and the address-content relations in each LUT are sorted as explained in [7,10]. The DM output bits $\mathbb{B}(\boldsymbol{A})$ corresponds to the QAM in-phase/quadrature amplitudes $\boldsymbol{A} \in \mathcal{R}^2$, where $\mathbb{B}(\cdot)$ and $\mathcal{R}$ denote symbol-to-bits conversion and the set of nonnegative real numbers, resp. The histogram of $\boldsymbol{A}$ is stored for monitoring the probability mass function (PMF) $P_{\boldsymbol{A}}$. The sign bits for the QAM symbol are bit scrambled, and then combined with dummy bits that emulate FEC parity bits, where the assumed FEC code rate is around 5/6. Both DM output bits and sign bits are multiplexed and input to the error insertion block. The schematic of the source generator is illustrated in Fig. 2. Each source bit is selected from one of three possible candidates; 0) bit logic '0', 1) a pseudo-random binary sequence (PRBS) of length 2^{31}–1 bits and 2) bit logic '1'. The mask signal, used for the selection and shown as an orange line, is generated from the target source mark ratio $P_S(1)$, which is given by the user. If $P_S(1) \leq 0.5$, the mask signal oscillates between 0 and 1 such that the average fraction of logic '1's in S is $P_S(1)$, and if $P_S(1) > 0.5$, the mask similarly oscillates between 1 and 2. For example, when the target $P_S(1)$ is 0.3, the mask value will be 0 and 1 during 40 and 60% of the time, resp., as shown in the right inset in Fig. 2. The mask value transient timing is gradually moved to avoid short repeated patterns.

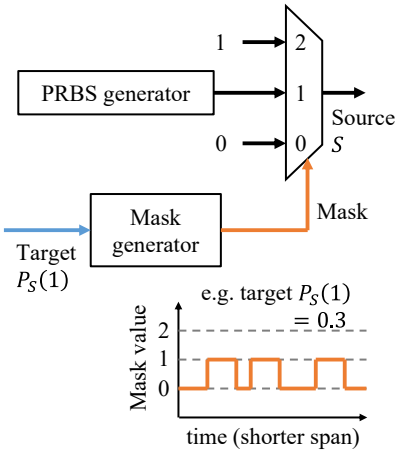

Fig. 2 Schematic of source generator.

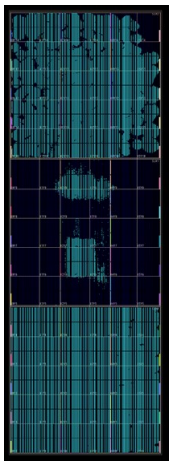

Fig. 3 Utilised area of the FPGA.

At the error insertion function in Fig. 1, exclusive ORs are performed between input bits and an error vector consisting of zeros and ones. The error vector is configured based on the target post-FEC BER (= pre-invDM BER) from the external setting. On the receiver side (right-hand side in Fig. 1), the dummy bits are separated from the sign bits. The error insertion and dummy bit removal emulates the channel, demapping, and FEC decoding. The remaining sign bits are descrambled. As for non-sign bits, invDM is performed and the bit-flipping is terminated. Then, the transmitted source bits and the recovered bits are compared and the number of erroneous bits is used to quantify the system output performance.

We performed FPGA fitting (synthesis) at a clock frequency of 90 MHz. Then the maximum system throughput was around 60 Gb/s. Figure 3 shows the graphical description of the utilised area of the FPGA. There are three layers called super-logic region (SLR) in the FPGA, which correspond to the top, middle and bottom square areas. Each SLR consists of 30 logic regions. As an initial implementation, the logical synthesis was performed without keeping the hierarchy of the register transfer level codes. The clock speed is limited by the critical path, which in our case is the line between SLRs. Table 1 shows the resource utilisation. A quarter of the available system logic cells were mainly used for the initial value setting function to the LUT in the hierarchical DM. The block random access memory (RAM) utilisation was 4.65%, which was mainly used for LUTs in the hierarchical DM. As shown in Tab. 2, the total power consumption was 601 mW and the main functions consumed 340 mW of it. The external functions include source generator, PMF monitor, error insertion and error checker. When we increase the system throughput, we must have multiple instances of the main functions, except for the control-bus interface and the initial value setting of the LUTs. Then, utilisation of the memory will be proportionally increased but that of the system logic cells will not. We need six instances to accommodate a system throughput of 400 Gb/s, and then the main function would consume, e.g., around 2 W. This is an implementation example for FPGA, so the clock frequency will be higher and power consumption will be reduced significantly when implementing it in an application-specific integrated circuit. The implemented circuitry is adoptable to formats from PS-8-QAM to PS-4096-QAM by setting the LUT contents to suitable values. Here we focused on PS-16-QAM and PS-64-QAM with DM word lengths of 200 and 100 complex symbols, resp., as example cases.

Table 1 Resource utilisation

Category	*Component*	*Available*	*Utilisation*
System logic cell	LUT as logic	1182k	24.17%
	Register	2364k	22.49%
Memory	Block RAM	75.9 Mb	4.65%
	Ultra RAM	270.0 Mb	Not used
DSP slice		6840 slice	Not used

Table 2 Power consumption

Category	*Function*	*Power consumption*
Environment		101 mW
External functions		159 mW
Main functions	Bit flip enc./dec.	13 mW
	DM/invDM	314 mW
	Bus interface and initial value setting	13 mW
	Total	340 mW
Total		601 mW

3 Experiments

By using the implemented FPGA and its evaluation kit, we experimentally verified the processing of DM/invDM and other functions such as source generation, source compression and monitoring. The histogram of the complex amplitude $\boldsymbol{A}$ for PS-16-QAM and PS-64-QAM was measured over more than 10^{10} complex symbols, and then converted into a PMF, after which the entropy $H(\boldsymbol{A})$ was computed. Figure 4 shows the measured DM output entropy vs. target source mark ratio $P_S(1)$. Because of the bit-flipping function in the joint source–channel coding, $H(\boldsymbol{A})$ is symmetric around $P_S(1) = 0.5$, as expected. When the $P_S(1)$ deviates from 0.5, $H(\boldsymbol{A})$ and the average symbol energy are reduced, while that for CCDM [3] are constant. Then the required signal-to-noise ratio (SNR) at a given performance is also reduced. Note that the DM rate losses for PS-16-QAM and PS-64-QAM were 0.034 and 0.064 bit per QAM symbols, resp. In the figure, we also show the binary entropy function $H(S)$ normalised to $H_{\max} = H(\boldsymbol{A})$ at $P_S(1) = 0.5$ (black, dashed) for comparison. The gap between $H(\boldsymbol{A})$ and $H(S)H_{\max}$ shows the rate loss in the data compression. The rate loss is ideally zero but took larger values at smaller $H(\boldsymbol{A})$ cases due to very simple processing of the joint source–channel coding. Regardless of the rate loss increase, $H(\boldsymbol{A})$ itself is reduced significantly to help reduce the required SNR substantially.

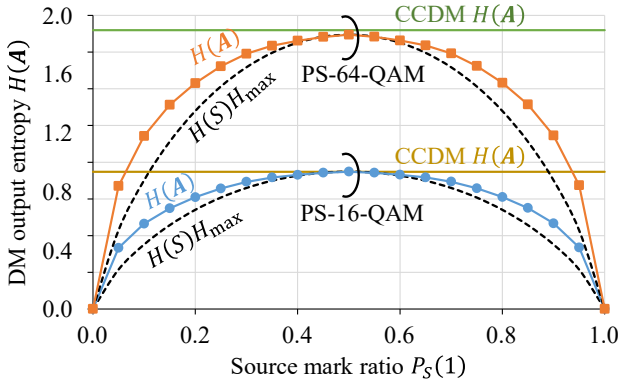

Fig. 4 Measured DM output entropy $H(\boldsymbol{A})$ as a function of target source mark ratio $P_S(1)$ based on $>10^{10}$ QAM symbols.

Next, we set an assumed post-FEC BER to the random error insertion function and measured the number of erroneous bits and the number of transmitted bits, i.e., the error enhancement of the invDM. The targeted source mark ratio was set to $P_S(1) = 0.3$, 0.5 or 0.7. Figure 5 shows measured system output BER as a function of an assumed post-FEC BER. The BER increase by the invDM was only around 10 times due to the partially decodable feature of the hierarchical DM. As predicted in [7], this ratio is significantly smaller than the other DMs such as CCDM. For example, a system output BER after invDM of 1.5×10^{-14} was measured at a post-FEC BER of 1.6×10^{-15} for PS-64-QAM. In this case, we observed more than 250 bit errors at the system end over 1.7×10^{16} transmitted source bits in a 70-hour measurement, which examined various source bits and error conditions only for the case $P_S(1) = 0.5$ due to the time constraint. These results ensure that the DM/invDM and joint source–channel coding will not cause error floors or excessive error enhancements, so the required post-FEC BER is kept to around 10^{-16}. Though we could not implement CCDM in the FPGA due to its high complexity, the invDM BER increase was simulated [7], and it was similar to the number of DM output bits per DM word, i.e. 400 or 200 for PS-16-QAM or PS-64-QAM in this case when we assume random bit errors before the invDM as the worst case.

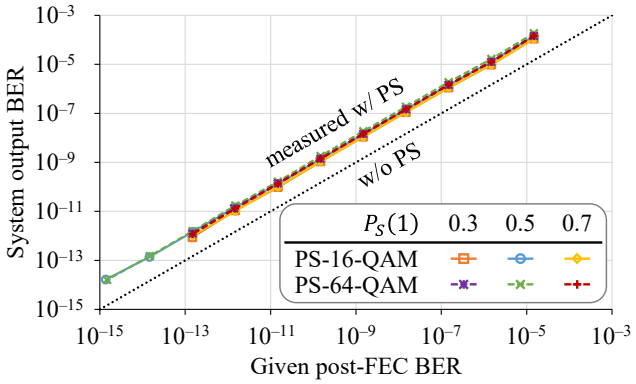

Fig. 5 Measured system output BER as a function of assumed post-FEC BER at back-to-back error insertion test.

4 Conclusions

As the key functions in reverse concatenation PS, DM and invDM were implemented and evaluated on an FPGA for the first time. It showed a reasonably small BER increase ratio, i.e. only around 10 times, despite the highly nonlinear logical operation in the invDM. Thus, we can reach to a very low BER around 10^{-14} also after the invDM at a post-FEC BER of about 10^{-15}. Here we assumed the errors before the invDM as randomly located, which is the worst case in terms of system output BER. No similar low-BER relations have been demonstrated for any other DM schemes.

In the FPGA implementation, there was no use of ultra RAM or DSP slices and it consumed ~25% of the system logic cells and 4.7% block RAM for 60 Gb/s throughput. Even with this FPGA, it would be possible to realise a system throughput of 100 Gb/s or more by implementing multiple parallel instances of the coding circuitry (in this case the incremental hardware resource is the block RAM only). We also implemented joint source–channel coding for source bits compression and quantified the smaller DM output entropy.

5 Acknowledgements

This work was partly supported by "Massively Parallel and Sliced Optical Network," the Commissioned Research of National Institute of Information and Communications Technology (NICT), and "The research and development of innovative optical network technology as a new social infrastructure" of the Ministry of Internal Affairs and Communications, Japan. We thank Assoc. Prof. Koji Igarashi of Osaka University for fruitful discussions.

6 References

[1] Böcherer, G., Steiner, F., Schulte, P.: 'Bandwidth efficient and rate-matched low-density parity-check coded modulation', IEEE Trans. Commun., 2015, 33, (20), pp. 4338–4352

[2] Buchali, F., Steiner, F., Böcherer, G., et al.: 'Rate adaptation and reach increase by probabilistically shaped 64-QAM: an experimental demonstration', J. Lightw. Technol., 2016, 34, (7), pp. 1599–1609

[3] Schulte, P., Böcherer, G.: 'Constant composition distribution matching', IEEE Trans. Inf. Theory, 2016, 62, (1), pp. 430–434

[4] Gültekin, Y. C., Willems, F. M. J., Hotums, W. J., et al.: 'Approximate enumerative sphere shaping', Proc. IEEE Int. Symp. Inf. Theory, USA, June 2018, pp. 676–680

[5] Fehenberger, T., Millar, D. S., Koike-Akino, T., et al.: 'Multiset-partition distribution matching', IEEE Trans. Commun., 2019, 67, (3), pp. 1885–1893

[6] Schulte, P., Steiner, F.: 'Divergence-optimal fixed-to-fixed length distribution matching with shell mapping', http://www.arxiv.org/pdf/1803.03614v2

[7] Yoshida, T., Karlsson, M., Agrell, E.: 'Hierarchical distribution matching for probabilistically shaped coded modulation', J. Lightw. Technol., 2019, 37, (6), pp. 1579–1589

[8] Koganei, Y., Sugitani, K., Nakashima, H., et al.: 'Optimum bit-level distribution matching with at most $O(N^3)$ implementation complexity', Proc. Opt. Fib. Commun. Conf. (OFC), USA, March 2019, p. M4B.4

[9] Cho, J., Winzer, P. J.: 'Multi-rate prefix-free code distribution matching', Proc. Opt. Fib. Commun. Conf. (OFC), USA, March 2019, p. M4B.7

[10] Yoshida, T., Karlsson, M., Agrell, E.: 'Joint source–channel coding via compressed distribution matching in fiber-optic communications', Proc. Opt. Fib. Commun. Conf. (OFC), USA, March 2019, p. M4B.6

[11] Vacondio, F., Bertran-Pardo, O., Pointurier, Y., et al.: 'Flexible TDMA access optical networks enabled by burst-mode software defined coherent transponders', Proc. Eur. Conf. Opt. Comm. (ECOC), UK, Sep. 2013, p. We.1.F.2

[12] Cai, Y, Wang, W., Qian, W., et al.: 'FPGA investigation on error-floor performance of a concatenated staircase and Hamming code for 400G-ZR forward error correction', Proc. Opt. Fib. Commun. Conf. (OFC), USA, March 2019, p. Th4C.2