



A high throughput architecture for a low complexity soft-output demapping algorithm

I. Ali, U. Wasenmüller, and N. Wehn

Microelectronic Systems Design Research Group, University of Kaiserslautern, 67663 Kaiserslautern, Germany

Correspondence to: I. Ali (imran@eit.uni-kl.de)

Received: 5 January 2015 – Revised: 15 May 2015 – Accepted: 11 August 2015 – Published: 3 November 2015

Abstract. Iterative channel decoders such as Turbo-Code and LDPC decoders show exceptional performance and therefore they are a part of many wireless communication receivers nowadays. These decoders require a soft input, i.e., the logarithmic likelihood ratio (LLR) of the received bits with a typical quantization of 4 to 6 bits. For computing the LLR values from a received complex symbol, a soft demapper is employed in the receiver.

The implementation cost of traditional soft-output demapping methods is relatively large in high order modulation systems, and therefore low complexity demapping algorithms are indispensable in low power receivers. In the presence of multiple wireless communication standards where each standard defines multiple modulation schemes, there is a need to have an efficient demapper architecture covering all the flexibility requirements of these standards. Another challenge associated with hardware implementation of the demapper is to achieve a very high throughput in double iterative systems, for instance, MIMO and Code-Aided Synchronization.

In this paper, we present a comprehensive communication and hardware performance evaluation of low complexity soft-output demapping algorithms to select the best algorithm for implementation. The main goal of this work is to design a high throughput, flexible, and area efficient architecture. We describe architectures to execute the investigated algorithms. We implement these architectures on a FPGA device to evaluate their hardware performance. The work has resulted in a hardware architecture based on the figured out best low complexity algorithm delivering a high throughput of 166 Msymbols/second for Gray mapped 16-QAM modulation on Virtex-5. This efficient architecture occupies only 127 slice registers, 248 slice LUTs and 2 DSP48Es.

1 Introduction

In transmitter, a constellation mapper takes groups of bits and maps them to particular constellation points. A specific magnitude and phase represents a certain combination of bits in the transmitted symbol. Due to distortion by the wireless channel, an error occurs in the position of each transmitted constellation point. In the receiver, the phase and magnitude of each received symbol is extracted, and a decision is made about what combination of bits the transmitter sent.

In the receiver, bit level demapping can be performed such that the output of demapper is “hard”, i.e., either a logical value 1 or 0. Alternatively, the demapper output can be “soft”; a soft-value indicating the probability, that the modulated bit associated with a given demapper output is to be of logical value 1 or 0. The soft-output (LLR) of the k th bit c_k in noisy received symbol sequence r is

$$\text{LLR}(c_k|r) = \ln \frac{p(c_k = 1|r)}{p(c_k = 0|r)} \quad (1)$$

where p denotes probability.

If modulating bits are uncoded or algebraic coded such as RS-codes or BCH codes, the demapper output is typically hard. If modulating bits are coded with a convolutional, LDPC, or Turbo-Code encoder the demapper output must be soft in order to yield superior performance. Consequently, soft-output demappers are an integral part of many modern communication receivers.

Optimal soft-output demapping algorithms involve computationally complex functions such as logarithmic and exponential functions, and thus are not well suited for hardware implementation. On the other hand, suboptimal methods significantly reduce the computational complexity by adopting simplified functions. However, they still require to calculate distances between the received symbol and all constellation

points Li et al. (2009); Su et al. (2011); Lin et al. (2010); Ryoo et al. (2003); Li and Shi (2014) and Lee et al. (2011). The computational complexity of suboptimal methods is further reduced in so-called low complexity soft-output demapping algorithms. A large number of works in this domain have focused on theoretical/simulation aspects of the algorithms aiming to attain superior frame error rate (FER) performance. Little attention has been paid to the actual implementation of such algorithms that look to deliver more than 100 Msymbols/second.

Usually the demapping function is executed only once on each burst in the receiver. However, the double iterative systems such as MIMO and Code-Aided Synchronization engage the demapper in their outer iteration. Consequently, the demapping function is executed multiple times on each burst in the receiver. Accordingly, the double iterative systems require very high throughput demapper. In this regard, consider the needs of gateway of Second Generation Digital Video Broadcasting Interactive Satellite System which is typically 20 Msymbols/second DVB (2014). In case of utilizing code-aided synchronization, typically 8 outer iterations are performed (see Fig. 1) to achieve the desired communications performance as reported in Ali et al. (2014). In such system, the demapper must deliver a throughput of 8-times 20 Msymbols/second. Therefore, we decided to set a minimum 160 Msymbols/second throughput specifications for this work.

In the presence of multiple wireless communication standards where each standard defines multiple modulation schemes, there is a need to have single demapper architecture covering all the flexibility requirements of these standards. We focus on popular Gray mapping M -PSK and M -QAM modulation schemes in this work which are specified in many wireless communication standards. The architectures reported in Altera (2007); Park et al. (2008); and Jafri et al. (2010) support multiple modulation schemes. However, they do not satisfy the throughput requirement outlined above.

Based on the results of a thorough literature search and deep analysis, we find the following algorithms having remarkable reduced complexity without compromising on quality of communications performance: (1) the algorithm reported in Lin et al. (2010) identifies the two required constellation points to compute one LLR in a very simple way and (2) The algorithms reported in Tosato and Bisaglia (2002); Ryoo et al. (2003); Kim et al. (2006); Arar et al. (2007); and Sun and Zeng (2011) are quite similar to each other and provide a very simple approach to compute LLR. We call this approach as decision threshold algorithm in this sequel.

The computational complexity of the aforementioned algorithms have been examined in the literature by counting number of operations required which is not a sufficient measure to derive realistic complexity for hardware implementation. Instead, hardware complexity metrics are: throughput,

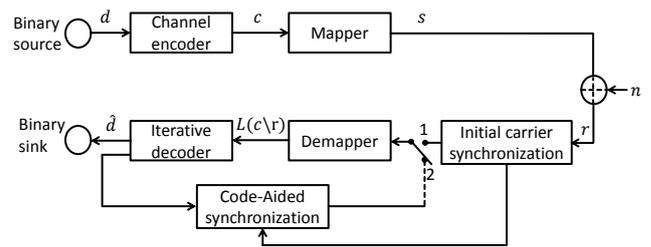


Figure 1. Baseband model of iterative channel decoding based system.

latency, resource utilization, and power. We investigate the hardware performance of the considered algorithms by realizing FPGA/ASIC implementations under the constraint of above specified throughput. At the conclusion of our work, we identify a demapping algorithm having the lowest implementation complexity.

This paper is structured as follows. In Sect. 2, we describe the system model as well as optimal and traditional suboptimal algorithms while Sect. 3 explains low complexity suboptimal algorithms. The communications performance of the algorithms is shown in Sect. 4. The hardware architectures and their implementation complexity are compared in Sect. 5, and Sect. 6 concludes this work.

2 System model

The system model comprising channel encoder, iterative channel decoder, mapper, demapper, initial carrier synchronization, and Code-Aided synchronization is shown in Fig. 1. The “Channel encoder” processes binary signal d and produces the encoded signal c . Then, the “Mapper” block maps M coded bits $c_0, c_1, \dots, c_{M-1} \in \{0, 1\}$ to a complex symbol s using the mapping function $s = \text{map}(c_0, c_1, \dots, c_{M-1})$. The Additive White Gaussian Noise (AWGN) channel adds noise n in the signal. Discrete-time baseband signal at the receiver can be represented as

$$r(k) = s(k) \cdot e^{j(2\pi k f_0 T + \Phi)} + n(k) \quad k = 0, 1, \dots, K-1 \quad (2)$$

where $s(k)$ is the transmitted signal, K is the length of the received signal, T is the symbol duration, f_0 is frequency offset, Φ is phase offset, and $n(k)$ is a sequence of complex white Gaussian noise samples with variance σ^2 .

In the receiver after performing automatic gain control, frame detection, timing synchronization, and initial carrier synchronization (phase/frequency) the resulting data sequence is transferred to “Demapper”. The demapping module demodulates the complex channel symbols and extracts M soft-outputs for a received symbol using a log likelihood ratio calculation. The “Iterative decoder” estimates the transmitted bits using soft-input from the “Demapper”. The soft-output of “Iterative decoder” is used by “Code-Aided synchronization” to further compensate the frequency and phase

offset from the received burst. Afterwards, the newly corrected burst is passed to “Demapper” and subsequently the next iteration of “Iterative decoder” is performed. Hence, in this double iterative system demapping function is performed after each iteration of the decoder. After presenting the system model, we discuss about the demapping algorithms. The soft-output demapping methods are classified into two major categories optimal and suboptimal which are explained in the subsequent sections.

2.1 Optimal soft demapping

For M -ary modulation scheme, the demapper needs to calculate log-likelihood ratios on the coded bits c_0, c_1, \dots, c_{M-1} for each incoming received symbol. The channel information of the coded bit c_k conditioned on the received symbol r can be calculated as follows.

$$\text{LLR}(c_k|r) = \ln \frac{\sum_{i=0}^{2^{M-1}-1} \exp \left[-\frac{1}{2\sigma^2} (r - s_{k(1,i)})^2 \right]}{\sum_{i=0}^{2^{M-1}-1} \exp \left[-\frac{1}{2\sigma^2} (r - s_{k(0,i)})^2 \right]} \quad (3)$$

where σ^2 is variance of AWGN channel, $s_{k(1,i)}$ and $s_{k(0,i)}$ represent the constellation points whose k th bits are one and zero respectively and M represents the number of bits in one modulated symbol. In 16-QAM modulation, four bits constitute a symbol so in this case $M = 4$.

It can be clearly seen in Eq. (3) that the optimal demapping method involves logarithmic and exponential functions to compute LLR. Because of these computationally complex mathematical operations, the optimal demapping method is not suitable for hardware implementation. This computational complexity is reduced in suboptimal demapping methods which is described in the following section.

2.2 Suboptimal soft demapping

In order to eliminate the logarithmic and exponential functions in Eq. (3), the suboptimal algorithms adopt an approximation. Since the sum term in Eq. (3) is dominated by the largest term, it can be simplified as reported in Robertson et al. (1995). The simplification can be formally expressed as

$$\ln \sum_j \exp(-a_j) \approx \max(-a_j) = \min(a_j) \quad (4)$$

where $a_j \geq 0$. With this approximation, LLR can be computed as follows.

$$\text{LLR}(c_k|r) = \frac{1}{2\sigma^2} \cdot \left[\min_i (r - s_{k(0,i)})^2 - \min_i (r - s_{k(1,i)})^2 \right] \quad (5)$$

where $i = 0, 1, \dots, 2^{M-1} - 1$.

It is evident from Eq. (5) that the suboptimal demapping algorithm significantly reduces the computational complexity by avoiding logarithmic and exponential functions as opposed to the optimal algorithm. Despite this simplification,

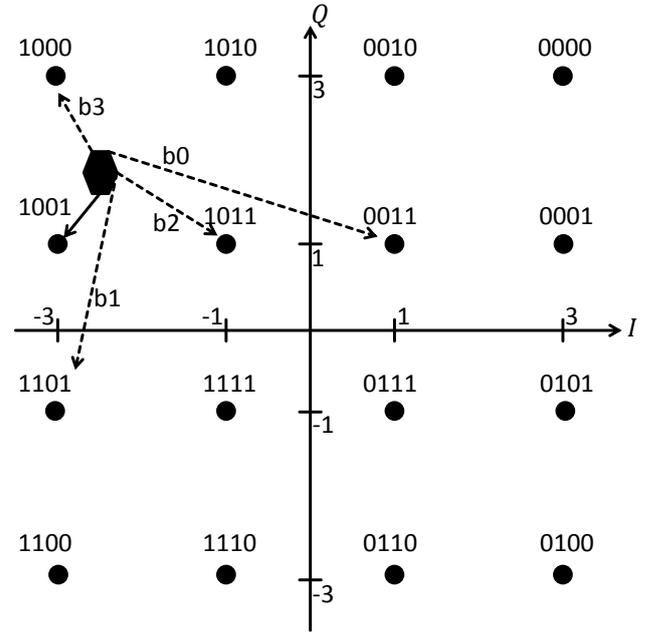


Figure 2. Lin algorithm illustration for Gray mapped 16-QAM modulation. The black color circles denote constellation points whereas the black color hexagon represents the received symbol

the suboptimal demapping algorithm involves computation of all possible Euclidean distances and then an exhaustive search to determine the two nearest constellation points. This complexity is considerably prominent in high order modulation schemes. This computational complexity can be further reduced by adopting some simple techniques as explained in the subsequent section.

3 Low complexity suboptimal demapping algorithms

This section explains two low complexity suboptimal demapping algorithms which are applicable to popular Gray mapped modulation schemes: (1) Lin algorithm and (2) decision threshold algorithm.

3.1 Lin algorithm

The algorithm described in Lin et al. (2010) does not compute all possible Euclidean distances as opposed to the traditional sub-optimal demapper. Instead, it identifies two constellation points $s_{k(0,i)}$ and $s_{k(1,i)}$ of Eq. (5) that are at minimum distance from the received symbol followed by computation of only two squared distances. This identification is carried out using very simple mathematical operations. This technique is explained with an example in Fig. 2.

The magnitude of the received symbol in the complex plane is $(-2.5, 1.5)$ in the considered example. At first, the Cartesian coordinates of the received symbol are rounded to the coordinates of its nearest constellation point (NCP). The magnitude of real part of the received signal is -2.5 which is closer to -3 as compared to -1 . Similarly, the imaginary part of the received symbol is 1.5 which is closer to 1 instead of 3 . Therefore, the NCP of the received symbol is $(-3, 1)$ in the constellation diagram. The corresponding bit mapping of the NCP is “1001”. This NCP is used to compute the first squared distance from the received symbol. Afterwards, four nearest constellation points are identified with respect to this computed CP where k th bit of the formers is flipped corresponding to the k th bit of the latter. We call the former constellation points as Flipped Constellation Points (FCPs).

For the considered case, the first bit b_0 (MSB) in the NCP (1001) is “1”. Now we want to compute the FCP whose first bit is “0”, i.e., flipped with respect to the first bit of NCP. This can be accomplished by using a simple transformation $x'x1x$. In this transformation term x' means flip the corresponding bit, x represent no change in the corresponding bit, and 1 means replace the corresponding bit by 1. Using this transformation, the computation of FCP for MSB of NCP (1001) results in “0011” which is highlighted in Fig. 2 for bit b_0 . The second squared distance is computed between the received symbol and this FCP. Similarly, the remaining three FCPs are computed corresponding to the second b_1 , third b_2 and fourth b_3 bits using the transformations $xx'x1$, $xxx'x$ and $xxxx'$ respectively.

Finally, LLR of one bit is computed by using the two calculated squared distances, i.e., squared distance between the received symbol and the NCP, and squared distance between the received symbol and its corresponding FCP. Remark, the first computed squared distance can be utilized for LLR calculation of the remaining three bits of the received symbol. In short, for a 16-QAM modulated symbol five squared distances are computed to calculate LLR of four bits, whereas for the same case the traditional sub-optimal demapper needs to compute 16 squared distances.

It is very important to mention that the abovementioned mathematical transformations to compute FCP are specific to the mapping scheme shown in Fig. 2. If the mapping scheme is changed, these mathematical transformations to compute FCP also need to be modified accordingly. Furthermore, this technique is applicable to only Gray mapping modulation schemes, including PAM, PSK, and square QAM. Under these constraints, the algorithm computes the distances which are exactly needed in Eq. (5) as claimed in Lin et al. (2010).

3.2 Decision Threshold Algorithm

In order to reduce the computational complexity, this algorithm adopts a simple decision threshold comparison mech-

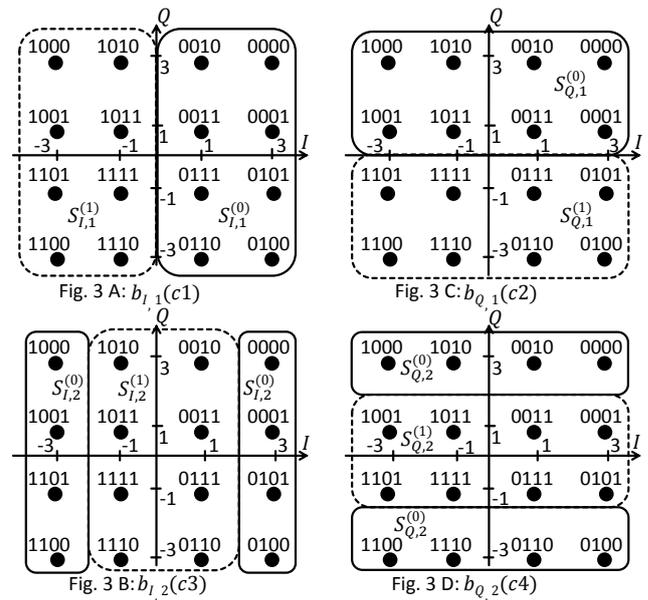


Figure 3. Partitions of the 16-QAM constellation used in decision threshold algorithm

anism, and thus a single distance computation accomplishes the soft demapping.

In the case of 16-QAM constellation, the partitions $(S_{I,k}^0, S_{I,k}^1)$ are shown for the generic in-phase components of the complex signal $b_{I,k}$ (c_1, c_2) in Fig. 3a and b. The partitions $(S_{Q,k}^0, S_{Q,k}^1)$ for the quadrature component of the complex signal $b_{Q,k}$ (c_3, c_4) are shown in Fig. 3c and d. The MSB c_1 is always 1 in the left half section and 0 in the right half section (see Fig. 3a). The second bit c_2 is always 1 in the lower half section and 0 in the upper half section (see Fig. 3c). The third bit c_3 is always 1 in the middle section and 0 in the outer section (see Fig. 3b). The LSB bit c_4 is always 1 in the middle section and 0 in the outer section (see Fig. 3d). The decision threshold algorithm exploits this property of Gray mapping and provides very simple expression to calculate the LLR.

As discussed that components of the complex signal are delimited by either horizontal or vertical boundaries. Therefore, the two symbols within the two subsets, nearest to the received signal, always lie in the same row if the partition boundaries are vertical (bits $b_{I,1}$ and $b_{I,2}$ in Fig. 3a and b) or in the same column if the boundaries are horizontal (bits $b_{Q,1}$ and $b_{Q,2}$ in Fig. 3c and d). The same observation holds true for 8-PSK and 64-QAM constellations. As a consequence, the LLR of the constituting bits in 16-QAM modulation can be derived as follows.

$$\text{LLR}(c_1|r) = \frac{2}{\sigma^2} \cdot [x] \quad (6)$$

$$\text{LLR}(c_2|r) = \frac{2}{\sigma^2} \cdot [y] \quad (7)$$

$$\text{LLR}(c_3|r) = \frac{2}{\sigma^2} \cdot \left[|x| - \frac{C+D}{2} \right] \quad (8)$$

$$\text{LLR}(c_4|r) = \frac{2}{\sigma^2} \cdot \left[|y| - \frac{C+D}{2} \right] \quad (9)$$

where two positive constants C and D represent the magnitudes of I and Q components of 16-QAM symbol which are 1 and 3 in this example. The terms x and y represent the distances of real and imaginary parts of the received symbol from the origin respectively in the complex plane. Regarding term $\frac{2}{\sigma^2}$, the detailed derivation can be found in Ryoo et al. (2003). The above described equations show that computing $\text{LLR}(c_1|r)$ and $\text{LLR}(c_2|r)$ require no distance calculation, whereas the computation of $\text{LLR}(c_3|r)$ and $\text{LLR}(c_4|r)$ require calculating two absolute values and two simple subtractions. It is worth mentioning that the expressions Eqs. (13) to (18) are specific to the mapping scheme shown in Fig. 3. If the mapping scheme is changed, these mathematical expressions also need to be modified accordingly. Furthermore, this technique is applicable to only Gray mapping modulation schemes.

For a given constellation diagram of Gray coded 8-PSK modulation, LLR of the constituting bits can be computed as follows.

$$\text{LLR}(c_1|r) = \frac{2}{\sigma^2} \cdot [x] \quad (10)$$

$$\text{LLR}(c_2|r) = \frac{2}{\sigma^2} \cdot [y] \quad (11)$$

$$\text{LLR}(c_3|r) = \frac{2}{\sigma^2} \cdot [|x| - |y|] \quad (12)$$

For a given constellation diagram of Gray coded 64-QAM modulation, LLR of the constituting bits can be computed as follows.

$$\text{LLR}(c_1|r) = \frac{2}{\sigma^2} \cdot [x] \quad (13)$$

$$\text{LLR}(c_2|r) = \frac{2}{\sigma^2} \cdot [-|x| + 4] \quad (14)$$

$$\text{LLR}(c_3|r) = \frac{2}{\sigma^2} \cdot [-| |x| - 4 | + 2] \quad (15)$$

$$\text{LLR}(c_4|r) = \frac{2}{\sigma^2} \cdot [y] \quad (16)$$

$$\text{LLR}(c_5|r) = \frac{2}{\sigma^2} \cdot [-|y| + 4] \quad (17)$$

$$\text{LLR}(c_6|r) = \frac{2}{\sigma^2} \cdot [-| |y| - 4 | + 2] \quad (18)$$

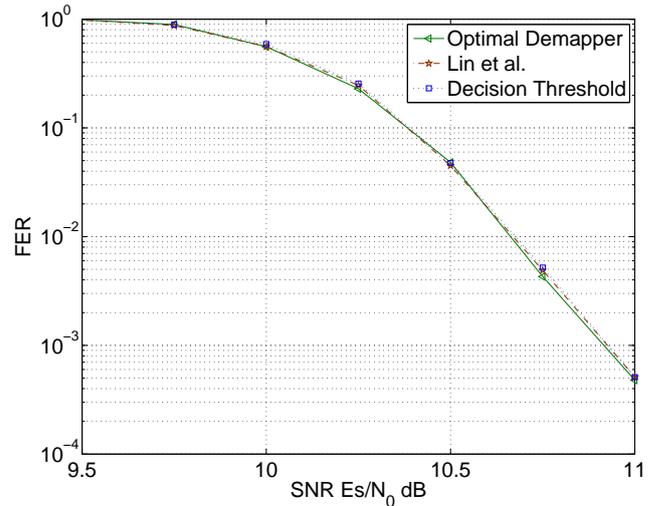


Figure 4. FER performance of Turbo-Code decoder after 8 iterations. The length of 16-QAM modulated burst is 536 symbols and code rate is 3/4. Three different demapping algorithms are applied: 1- optimal, 2- Lin algorithm, and 3- decision threshold algorithm

4 Communications performance

We compare the communications performance of optimal algorithm, Lin algorithm and decision threshold algorithm in Fig. 4. The simulations were carried out with bit true models of the hardware units to take into account quantization losses. We used 9 bit quantization each for input real and imaginary component, 6 bit for $\frac{2}{\sigma^2}$, and 6 bit for output LLR. We used a 16-state duo-binary Turbo-Code decoder in our simulations having Max-Log-Map with 0.75 extrinsic scaling factor, 8 iterations, and 7 bit for the extrinsic LLR. Both initial carrier synchronization and Code-Aided synchronization are performed to compensate the phase and frequency offsets. The FER graph clearly shows that the performance of all investigated algorithms is nearly identical. In short, by setting appropriate value of $\frac{2}{\sigma^2}$ the investigated suboptimal algorithms show similar communications performance to that of optimal algorithm. Remark that the simplified mathematical expressions Eqs. (6) to (9) adopted in decision threshold algorithm are equivalent to Eq. (5) adopted in Lin algorithm.

5 Hardware performance

In this section, we describe the architectures for abovementioned low complexity suboptimal demapping algorithms and compare their implementation performance. We used synthesizable VHDL to model the architectures.

5.1 Architecture for Lin algorithm

We present the architecture for Lin algorithm in Fig. 5. This architecture is described to support only 16-QAM modula-

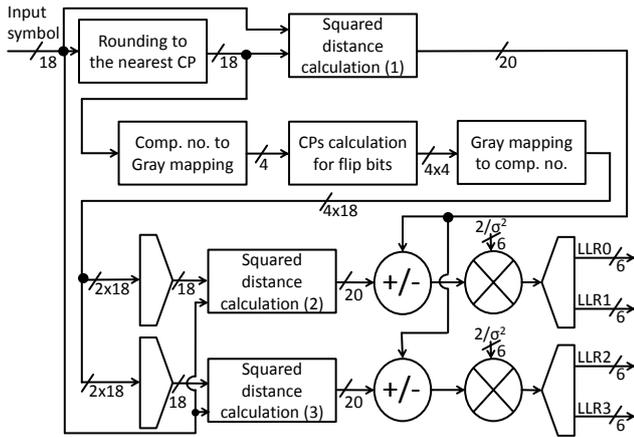


Figure 5. Proposed architecture for Lin algorithm.

tion scheme for evaluation purpose but in reality the Lin algorithm can be applied to all Gray mapped M -PSK and M -QAM modulation schemes.

In the architecture, the rounding of the input received symbol towards the NCP is carried out in “Rounding to the nearest CP” block according to the procedure explained in Sect. 3.1. Then first squared distance is computed between the NCP and the received input symbols in “Squared distance calculation (1)” block to implement one term of Eq. (5). This result is used either first or second term of this equation depending upon the value of the corresponding bit. The NCP which is a complex number, is mapped to a predefined Gray code in “Comp. no. to Gray mapping” block. The resulting Gray code of the NCP is used to compute the nearest FCP. With respect to 16-QAM modulation scheme, four FCPs are computed. This operation is performed in “CPs calculation for flip bits”. The resulting Gray codes of four FCPs are converted into complex numbers in “Gray mapping to comp. no.”. The squared distances are computed between the resulting complex numbers of four FCPs and the received symbol in “Squared distance calculation (2)” and “Squared distance calculation (3)”. To save and reuse the hardware units (multiplier and adder), we compute only two squared distances at a time and therefore we use a multiplexer and a demultiplexer at the input and output of the multipliers.

The results of “Squared distance calculation (1)”, and either “Squared distance calculation (2)” or “Squared distance calculation (3)” are used to compute LLR of each bit. All in all, only two squared distances are used to compute LLR of each bit. Finally, $\frac{2}{\sigma^2}$ is multiplied to compute LLR. The quantization bitwidths adopted in this work are mentioned in the figure. We compute two LLRs per clock cycle to achieve the aforementioned throughput.

5.2 Architecture for decision threshold algorithm

We present the architecture for decision threshold algorithm in Fig. 6. The proposed architecture provides flexibility to

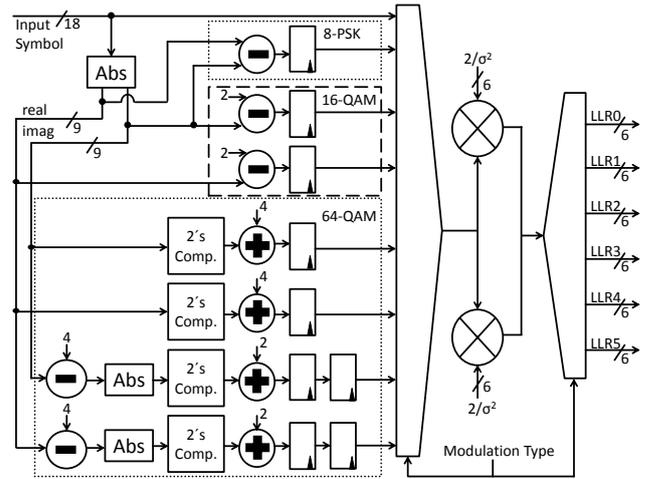


Figure 6. Proposed architecture for decision threshold algorithm

Table 1. FPGA (Virtex-5) Place-and-Route results comparison.

Parameter	Jafri et al. (2009)	Proposed 1	Proposed 2
Algorithm	conventional suboptimal	Lin algorithm	decision threshold algorithm
Flexibility Modulation Support	BPSK to 256-QAM	16-QAM	BPSK, QPSK, 8-PSK, 16-QAM, 64-QAM
Mapping Frequency (MHz)	Gray, non-Gray 156	Gray 312	Gray 333
Throughput (Msymbols/sec) 16-QAM	26	156	166
Resources			
Slice Regs	1596	288	127
Slice LUTs	2627	279	248
DSP48Es	6	8	2
BRAM	8	0	0

support following modulation schemes: BPSK, QPSK, 8-PSK, 16-QAM, and 64-QAM. In the case of BPSK and QPSK the input received symbol is directly used for LLR calculation. For the other mentioned modulation schemes, LLRs of first two bits are computed directly from the input received symbol because they do not involve any arithmetic operations. The computation of LLRs of the other bits involve simple arithmetic operations like absolute value, addition, subtraction and 2’s complement. We calculate two LLRs per clock cycle to achieve the desired throughput.

5.3 FPGA implementation results

We compare the implementation results of our proposed architectures with a state-of-the-art demapper Jafri et al. (2010) in Table 1. As the latter design is implemented on a Xilinx Virtex-5 FPGA, so we used the same FPGA device (xc5vlx330-2ff1760) for implementation of our architectures to make a fair comparison.

The implementation results show that our proposed architectures achieve a much higher clock frequency and consequently deliver almost 6 times higher throughput besides occupying almost 10 times less resources as compared with state-of-the-art implementation. These results also show that the architecture based on decision threshold algorithm has less implementation complexity than that of Lin algorithm. The former saves 56 % slice registers, 12 % slice LUTs and 75 % DSP48Es than the latter. In summary, decision threshold algorithm has the lowest implementation complexity among the investigated architectures.

5.4 ASIC implementation results

Because the architecture described for decision threshold algorithm shows the lowest implementation complexity on FPGA, we selected this architecture for ASIC implementation. We implemented it on a 65 nm low power CMOS library. We used Synopses tools to perform Synthesis and P&R. This efficient design occupies only 0.006 mm² area (2886 gates) after P&R and with worst case process parameters (1.1V, 125 °C). The design achieves a high clock frequency of 645 MHz, and therefore it delivers a very high throughput of 322 Msymbols/second with 16-QAM modulation. The design consumes only 3.85 mW power at nominal case.

6 Conclusions

Our investigation reveals that the decision threshold algorithm is a clear winner among the investigated demapping algorithms from the point of view of communications and implementation performance. The communications performance achieved by this algorithm costs only a tiny fraction of the computational effort required to achieve the same communications performance using the optimal and traditional sub-optimal algorithms. We have presented a very high throughput, area efficient, low power, and flexible architecture based on this algorithm. Our proposed architecture delivers almost 6 times higher throughput and requires about 10 times less resources on a FPGA as compared with state-of-the-art implementation.

Edited by: J. Anders

Reviewed by: two anonymous referees

References

- Digital Video Broadcasting (DVB): Second Generation DVB, Interactive Satellite System (DVB-RCS2), Part 2: Lower Layers for Satellite standard, ETSI EN 301 545-2 V1.2.1, 2014.
- Ali, I., Wasenmüller, U., and Wehn, N.: Hardware Implementation Issues of Carrier Synchronization for Pilot-Symbol Assisted Bursts: A Case Study for DVB-RCS2, in: 8th IEEE International Conference on Signal Processing and Communication Systems, Gold Coast, Australia, doi:10.1109/SPC.2013.6735092, 2014.
- Altera: Constellation Mapper and Demapper for WiMAX, Tech. rep., <http://www.altera.com/literature/an/an439.pdf>, May 2007.
- Arar, M., Amours, C. D., and Yongacoglu, A.: Simplified LLRs for the Decoding of Single Parity Check Turbo Product Codes Transmitted Using 16QAM, *Research Letters in Communications*, p. 4, doi:10.1155/2007/53517, 2007.
- Jafri, A. R., Baghdadi, A., and Jezequel, M.: Rapid design and prototyping of universal soft demapper, in: IEEE International Symposium on Circuits and Systems (ISCAS), 3769–3772, 2010.
- Kim, K. S., Hyun, K., Yu, C. W., Park, Y. O., Yoon, D., and Park, S. K.: General Log-Likelihood Ratio Expression and its Implementation Algorithm for Gray-Coded QAM Signals, *ETRI Journal*, 28, 291–300, 2006.
- Lee, J. H., Sunwoo, M. H., Kim, P. S., and Chang, D.-I.: Low complexity soft-decision demapper for DVB-S2 using phase selection method, in: Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication, p. 45, ACM, 2011.
- Li, J. and Shi, Y.: Simplified Soft-output Demapper Based on a Linear Transformation Technique for M-ary PSK., *Sensors & Transducers*, 181, 1726–5479, 2014.
- Li, M., Nour, C., Jogo, C., and Douillard, C.: Design of rotated QAM mapper/demapper for the DVB-T2 standard, in: IEEE Workshop on Signal Processing Systems, SiPS 2009, 018–023, doi:10.1109/SIPS.2009.5336265, 2009.
- Lin, D., Xiao, Y., and Li, S.: Low Complexity Soft Decision Technique for Gray Mapping Modulation, *Wireless Personal Communications*, 52, 383–392, 2010.
- Park, J. W., Sunwoo, M. H., Kim, P. S., and Chang, D.-I.: Low complexity soft-decision demapper for high order modulation of DVB-S2 system, in: International SoC Design Conference (ISOC '08), vol. 02, II–37–II–40, 2008.
- Robertson, P., Villebrun, E., and Hoeher, P.: A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain, in: Communications, 1995. ICC '95 Seattle, IEEE International Conference on “Gateway to Globalization”, vol. 2, 1009–1013, doi:10.1109/ICC.1995.524253, 1995.
- Ryoo, S., Kim, S., and Lee, S. P.: Efficient soft demapping method for high order modulation schemes, in: CDMA International Conference (CIC), Seoul, Korea, 2003.
- Su, J., Lu, Z., Yu, X., and Hu, C.: A novel low complexity soft-decision demapper for QPSK 8PSK demodulation of DVB-S2 systems, in: International Conference of Electron Devices and Solid-State Circuits (EDSSC), 1–2, doi:10.1109/EDSSC.2011.6117701, 2011.
- Sun, X. and Zeng, Z.: A Novel Simplified Log-Likelihood Ratio for Soft-Output Demapping of CMMB, in: Advances in Computer Science and Education Applications, vol. 202, 350–356, Springer, Berlin, Heidelberg, 2011.

Tosato, F. and Bisaglia, P.: Simplified soft-output demapper for binary interleaved COFDM with application to HIPERLAN/2, in: IEEE International Conference on Communications (ICC), vol. 2, 664–668, 2002.