EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

An adaptive low-power LDPC decoder using SNR estimation

Joo-Yul Park and Ki-Seok Chung*

Abstract

Owing to advancement in 4 G mobile communication and mobile TV, the throughput requirement in digital communication has been increasing rapidly. Thus, the need for efficient error-correcting codes is increasing. Furthermore, since most mobile devices operate with limited battery power, low-power communication techniques are attracting considerable attention lately. In this article, we propose a novel low-power, low-density parity check (LDPC) decoder. The LDPC code is one of the most common error-correcting codes. In mobile TV, SNR estimation is required for the adaptive coding and modulation technique. We apply the SNR estimation result to the proposed LDPC decoding to minimize power consumption due to unnecessary operations. The SNR estimation value is used for predicting the iteration count until the completion of the successful LDPC decoding. When the SNR value is low, we omit computing the parity check and the tentative decision. We implemented the proposed decoder which is capable of adaptively skipping unnecessary operations based on the SNR estimation. The power consumption was measured to show the efficiency of our approach. We verified that, by using our proposed method, power consumption is reduced by 10% for the SNR range of 1.5-2.5 dB.

Keywords: LDPC, decoder, SNR estimation, low power

1. Introduction

Recent advances in 4 G mobile communication systems require reliable high transmission rates. The bandwidth requirement for high speed 4 G information transfer is 100 Mbps, while the requirement for low-speed or stationary-state transfer is 155 Mbps-1 Gbps. Thus, the use of powerful error-correcting codes is crucial for the next generation mobile communication system [1]. Low-density parity check (LDPC) decoding has become especially relevant because of its excellent error correcting capability.

LDPC codes are linear block codes that were originally introduced by Gallager [2] in 1962. During that time, they attracted little attention, since hardware implementation of such decoding was impractical in the 1960s, and has been neglected since. However, the value of LDPC codes was rediscovered by Mackay and Neal in 1995, and many subsequent studies have shown that LDPC probabilistic decoding is very effective [3,4]. Recently, Chung et al. [5] showed that LDPC codes can come within 0.0045 dB of the Shannon limit. Turbo code was regarded as the best channel-coding technique before the rediscovery of LDPC, but LDPC codes have a smaller minimum distance than Turbo codes. LDPC codes exhibit very good BER curves, because they suffer from minimal error floor issues. Furthermore, iterative LDPC decoding schemes based on the Sum-Product algorithm [6] can be fully parallelized, leading to high-speed decoding [7]. For these reasons, LDPC codes are very attractive for high-speed 4 G wireless communication. Currently, DVB-S2, which is an European high-quality digital satellite broadcasting standard, features the concatenation of LDPC codes with BCH codes as their channel-coding scheme [8].

Dynamic power consumption of a module is proportional to the amount of switching activities. With a low SNR, the received signal may not be successfully decoded before the maximum number of iterations is reached, and the corresponding decoding may consume a great amount of power. On the contrary, at a high SNR, the decoding may succeed with a fewer number of iterations before the pre-defined maximum number of iterations is tried. Therefore, it is important to accurately estimate SNR values to achieve high-speed, low-power decoding. For excellent BER performance, both the size of the block and

* Correspondence: kchung@hanyang.ac.kr
Department of Electronics, Computer and Communication Engineering, Hanyang University, Seoul, Korea

the maximum number of decoding iterations should be large. Decoding large blocks requires significant amounts of computation and memory. In addition, decoding latency as well as power consumption will increase significantly, in turn, creating a significant decrease in communication bandwidth [4,9]. Recent studies have focused on lowering power consumption by adjusting either the maximum number of iterations or the quantization level according to estimated SNR values [10-12]. In this article, we propose a novel adaptive architecture that selectively carries out the tentative decision and parity-check operations depending on estimated SNR values to reduce power consumption.

The remainder of this article is organized as follows. In Section 2, a typical decoding algorithm and principles of LDPC decoding with adaptive coding and modulation (ACM) are presented. A novel, low-power LDPC decoding algorithm is presented in Section 3. In Sections 4 and 5, we present a performance evaluation of our novel design. Section 6 concludes this article.

2. Background

2.1. LDPC decoding algorithms

LDPC codes are linear block codes based on a parity-check matrix called an H-matrix, in which rows and columns represent parity-check codes and symbols, respectively. The H-matrix can be equivalently represented by a Tanner graph, a bipartite graph in which one partite has check nodes and the other has bit nodes. The check nodes correspond to the rows of the H-matrix, while the bit nodes correspond to the columns of the H-matrix. An H-matrix and the equivalent Tanner graph for an illustrative (10, 5) code are shown in Figures 1 and 2, respectively [6].

A conventional LDPC decoding algorithm is shown in Algorithm 1. Decoding is carried out iteratively in such a way that adjacent nodes in the Tanner graph exchange probabilities for the received codeword as shown in Figure 2. Decoded code words are checked against the H-matrix as shown in Step 9 of Algorithm 1. If the parity-check equation is satisfied ($H\ C^T = 0$), the decoding ends successfully even before the pre-defined maximum number of iterations is reached [13].

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Figure 1 Parity check matrix (length 10, dimension 5).

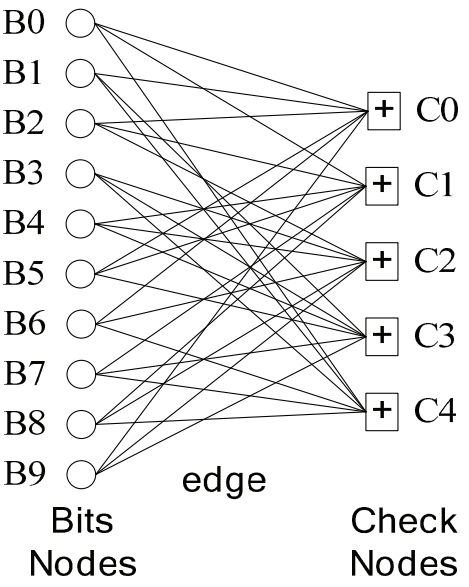

Figure 2 H (10, 5) Tanner graph.

2.2. Adaptive coding and modulation

In the ACM architecture, the data from a base station are transmitted after channel coding, interleaving, and modulation are processed. The receiver first estimates the channel state information with the received signal and then sends the estimation result back to the sender. The channel state estimation is typically performed according to the SNR value. The sender determines the modulation and coding scheme (MCS) [8] level based on this information and adaptively applies channel coding, interleaving, and modulation methods according to the channel state for the upcoming transmission. ACM techniques typically result in better transmission rates with smaller error rates than typical coding and modulation techniques, since the proper MCS level is determined based on the estimated channel state. Currently, standards for mobile multimedia services such as DVB-S2 and DVB-T2 employ ACM techniques. In ACM, accurate channel state estimators are crucial [14]. We used experimental results to identify the best channel estimator, and propose a new adaptive decoding algorithm that utilizes information provided by this accurate channel state estimator (Figure 3).

3. Proposed LDPC decoder

The overall performance of LDPC decoding depends significantly on the number of decoding iterations. Large numbers of iterations may result in unacceptably long delays that may, in turn, lead to failures in real-time processing. Therefore, programmers typically set a limit on the maximum number of iterations allowed by LDPC. If the parity-check equation is satisfied, decoding

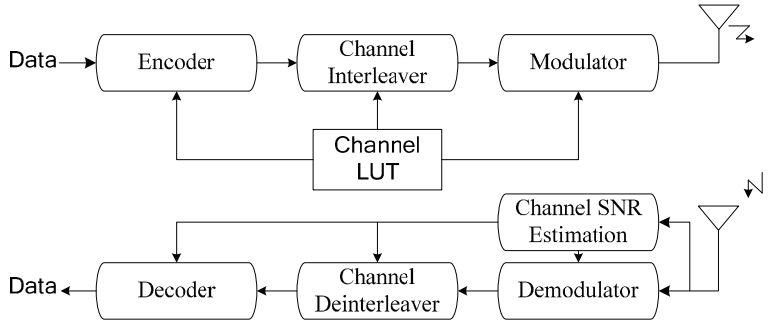

Figure 3 Adaptive coding and modulation architecture.

is completed even before the maximum number of iteration is reached.

In our novel decoder, we carry out the parity-check operation only if SNR estimates fall within a certain range. In this way, we reduce power consumption and decoding latency. As discussed in the previous section, our proposed scheme is very efficient because it does not require any additional hardware.

3.1. Architecture of the proposed LDPC decoder

$$\hat{\rho}_{\mathrm{ML}} = \frac{\left|\frac{1}{N}\sum_{m=0}^{N-1} r_m c_m^*\right|^2}{\frac{1}{N}\sum_{m=0}^{N-1}\left|r_m c_m^*\right|^2 - \left|\frac{1}{N}\sum_{m=0}^{N-1} r_m c_m^*\right|^2} \tag{8}$$

$$\hat{\rho}_{\mathrm{SNV}} = \frac{\left[\frac{1}{N}\sum_{m=0}^{N-1} r_m c_m^*\right]^2}{\frac{1}{N}\sum_{m=0}^{N-1} r_m{}^2 - \left[\frac{1}{N}\sum_{m=0}^{N-1} r_m c_m^*\right]^2} \tag{9}$$

The architecture of the proposed LDPC decoder is shown in Figure 4. The sender inserts the start of frame (SOF) into the encoded signal, and the receiver estimates the SNR. The estimated SNR information is provided to the LDPC decoder to determine whether the parity check and the tentative decision will be computed. An accurate estimation of SNR is crucial for the proposed adaptive parity-check scheme to succeed. To identify the best estimation algorithm, we assessed the accuracies of the two existing algorithms which are known to be reliable. First, the accuracy of the maximum likelihood (ML) algorithm (Equation 8), which is adequate for a short SOF as in the case of the DVB-S2 standard, was measured. Second, the accuracy of the signal-to-noise variance (SNV) (Equation 9) algorithm [15], which is a special case of the ML algorithm, was measured. In this experiment, we assumed a SOF of 26 symbols, as used in DVB-S2. In Equations 8 and 9, c is the value of pilot symbols defined in SOF, and the receiver already knows the value. By using the correlation between c value and the received value r, SNR values are estimated.

Figure 5 shows the results of SNR estimation. We decided to use the SNV algorithm as our SNR estimation algorithm since it outperforms the ML algorithm with respect to mean square error (MSE). As mentioned earlier, an SNR estimator is required for ACM. It should be noted that this estimator is not necessary for our proposed adaptive parity check, but it is only necessary for ACM, which requires SNR estimation. Therefore, we believe that our scheme can improve decoding performance without incurring any overhead in terms of hardware.

3.2. Adaptive parity check by SNR estimation

We assessed the number of decoding iterations associated with various SNR values to see how the number of iterations changes according to the SNR value. Table 1 summarizes the result after we carried out a simulation of 1,000,000 frame data for a rate-1/2 LDPC code with a block length of 9216 and a dimension of 4608 (CMMB code rate = 1/2 code) [16]. For each SNR value, we assessed the average number of iterations and the minimum number of iterations.

When SNR values were low, high iteration counts were necessary, whereas when SNR values were high, low iteration counts were sufficient. Hence, the tentative decision and the parity-check equation need not be

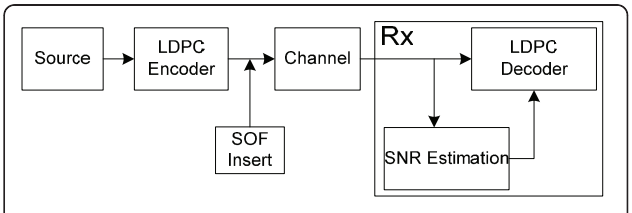

Figure 4 Proposed LDPC decoding flow.

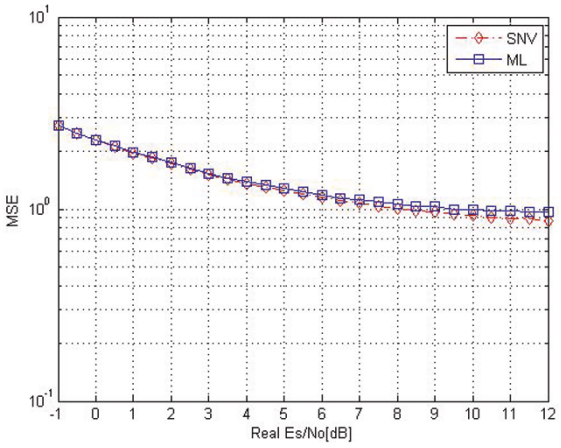

Figure 5 SNR vs. SNR estimation (MSE).

computed until we obtain a reasonable SNR value after a certain number of iterations.

A conventional modified UMP-BP algorithm computes the tentative decision and the parity-check equation after every iteration (Figure 6). According to Table 1 when the SNR is 2 dB, the iteration should be repeated at least six times. Therefore, it is not meaningful to compute the tentative decision and the parity check equation until this minimum number of iterations is achieved. Computing unnecessary values will increase both the decoding delay and the amount of power consumed. In our proposed scheme, based on the SNR, we store the minimum number of iterations in a look-up table to selectively carry out the parity check step. By this method, we may postpone computing the tentative decision and the parity check equation until the predetermined minimum number of

Table 1 The number of iterations versus SNR value (CMMB R = 1/2)

SNR	Iteration #			SNR	Iteration #		
	Min	Max	Avg		Min	Max	Avg
-1	50	50	50	6	2	4	2.16
-0.5	50	50	50	6.5	1	3	2.02
0	50	50	50	7	1	3	1.86
0.5	50	50	50	7.5	1	3	1.48
1	42	50	49.88	8	1	2	1.16
1.5	10	50	21.51	8.5	1	2	1.04
2	6	19	10.94	9	1	2	1.01
2.5	4	11	7.75	9.5	1	2	1
3	4	9	6.03	10	1	2	1
3.5	4	7	4.91	10.5	1	2	1
4	3	6	4.1	11	1	1	1
4.5	2	5	3.45	11.5	1	1	1
5	2	4	3.02	12	1	1	1
5.5	2	4	2.6				

[9216, 4608], Max iteration: 50, Frame: 1,000,000

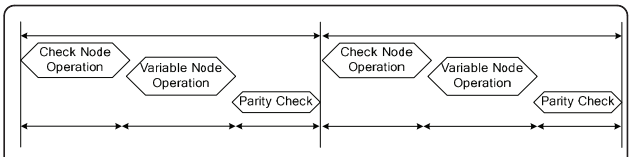

Figure 6 Conventional LDPC decoding flow.

iterations is reached. Our proposed scheme is summarized in Algorithm 2.

Conventionally, a parity-check operation is conducted at every decoding iteration step. However, in Algorithm 2, the parity-check operation is carried out only at the predetermined iteration count based on the predicted SNR as shown in Figure 7.

The low-power technique proposed in the article requires an accurate SNR estimation. As Table 1 shows, the valid SNR range where the LDPC decoding can be carried out is greater than or equal to 1.5 dB. Experimental results in Figure 5 show that when the SNR is greater than or equal to 1.5 dB, the error range is 0-2 dB. The results show that the range of the minimal iteration count according to the SNR will be from 1 through 10, and the difference in the iteration counts for each SNR value is on average less than 3. Therefore, the proposed low-power method cannot only work correctly, but also reduce the power consumption even if the error occurs in SNR estimation.

4. Simulation results

To evaluate the effectiveness of the proposed algorithm, simulations were conducted for mobile communication standards such as CMMB [16] and DVB-S2 [17]. The simulation results for CMMB are summarized in Table 1. The experimental results show that power consumption and the amount of computation required are effectively reduced when the SNR is less than 6. Also, we verified that there was no BER performance degradation when we applied the proposed algorithm.

DVB-S2 supports various code rates, and decoding is carried out by selecting an appropriate code rate according to the SNR value. Figure 8 shows BER versus SNR curves when the H-matrix of DVB-S2 is short in length. From these figures, we observe that as SNR values increase, and performance degradation becomes minimal even if we increase the code rates if the ACM

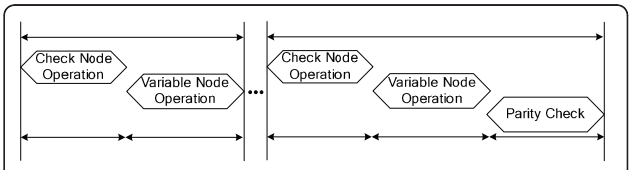

Figure 7 Proposed LDPC decoding flow.

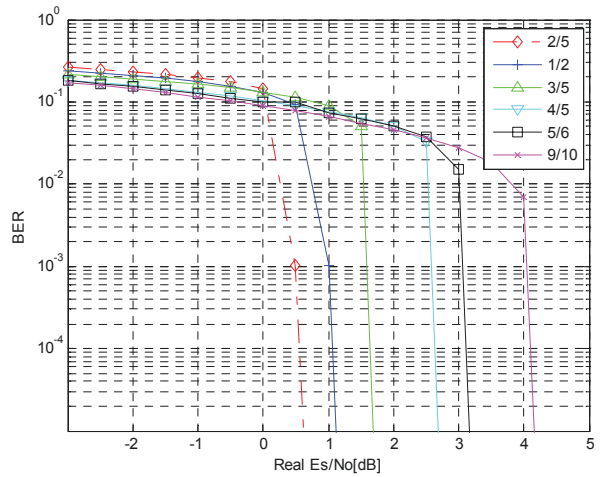

Figure 8 BER performance DVB-S2 BPSK short.

technique was applied. Therefore, in DVB-S2, decoding is processed by selecting an appropriate code rate depending on the SNR value.

Figure 9 shows the average iteration counts for various code rates versus SNR values. It is clearly observed that the iteration counts change as code rate and SNR values. We apply the proposed adaptive parity check algorithm by selecting the best code rate for each SNR value. The best code rates for each SNR value were drawn from results of simulations and previously published reports [18].

Table 2 summarizes the iteration counts when the block length is short in DVB-S2 standard. Table entries in bold and italic fonts are the selected best code rates for specific SNR values. For example, when SNR is a value between -0.5 and -1, the best code rate is 2/5. When SNR falls between 1.5 and 2, the best code rate is 3/5.

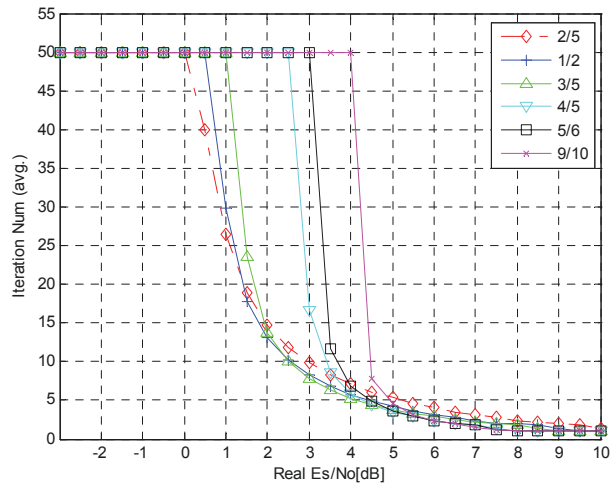

Figure 9 Iteration count DVB-S2 BPSK short.

Table 2 The number of iterations vs. SNR value (DVB-S2 short)

	2/5		1/2		3/5		4/5	
	Min	Avg	Min	Avg	Min	Avg	Min	Avg
0.5	***37***	***39.93***	50	50	50	50	50	50
1	20	26.5	***23***	***29.88***	50	50	50	50
1.5	14	18.94	15	17.85	***19***	***23.62***	50	50
2	11	14.74	11	13.09	11	13.71	50	50
2.5	9	11.89	8	10.22	8	10.01	50	50
3	7	9.92	7	8.24	6	7.84	***13***	***16.61***
3.5	6	8.3	5	6.79	5	6.36	7	8.61
4	5	7.12	4	5.75	4	5.27	5	5.95
4.5	4	6.09	3	4.96	4	4.42	3	4.47
5	4	5.32	3	4.24	3	3.84	3	3.5
5.5	3	4.59	2	3.66	3	3.25	2	2.98
6	3	4.05	2	3.19	2	2.96	2	2.31

When we use our novel proposed algorithm, we can select a code rate of 3/5 when the SNR is 2.7, and the average iteration count will be 23.62. However, since the minimum iteration count will be 19 in this case, we may skip the parity check and tentative checking operations up to the 19th iteration without affecting performance.

By running simulations, we verified that skipping the parity check and tentative check operations up to the minimum number of iterations does not affect performance. In addition, by skipping these operations, the total simulation time was reduced significantly. Table 3 shows the reduced simulation time after applying the proposed novel algorithm.

In the next section, we present experimental results on power consumption when we implement the proposed algorithm in hardware.

Table 3 Simulation time comparison (DVB-S2 short, 2500 frame)

	2/5		3/5		2/3		4/5	
	Before	After	B	A	B	A	B	A
1	2313	2249						
1.5	1603	1567	1515	1477	2778	2711		
2	1210	1183	1081	1054	1534	1497		
2.5	972	950	834	814	1105	1078		
3	801	784	669	652	857	837	1160	1133
3.5	667	558	586	573	693	676	606	591
4	571	480	468	458	577	564	425	414
4.5	491	420	397	390	493	480	326	319
5	429	369	346	338	423	413	266	259
5.5	377	325	299	294	379	368	217	213
6	332	324	267	262	326	319	192	187

5. Implementation results

To evaluate the performance of our algorithm, we implemented a decoder for (3,6) regular LDPC codes, the length of which is 9216 (CMMB code rate = 1/2 code). As shown in Figure 10, we implemented the proposed decoder, which has a partially parallel architecture [19].

To implement the proposed adaptive parity-check architecture, look-up tables were used both to indicate whether we should carry out parity check and to adjust the coefficients of the Modified Min-Sum algorithm.

The proposed LDPC decoder architecture was synthesized by Synopsys's Design Compiler using the Chartered 0.18 μm CMOS cell library. The size of the implemented decoder is 256 K (in NAND2) (Table 4).

We measured the power consumption of the synthesized design using Synopsys's Power Compiler. Figure 11 shows the amount of power consumption for each operation when a decoding iteration is carried out. A tentative operation simply stores the decision value in a buffer using the results from bit node operation. Thus, the amount of power consumption is not significant. However, the parity-check operation first reads values from the buffer using addresses generated by AGU, and then carries out parity checking, where the amount of power consumption is significant. The amount of power consumption overhead due to the addition of the comparison unit between SNR values and LUTs to implement the proposed algorithm is negligible (less than 0.1% of the total power consumption). The amount of power consumed by the SNR estimator is not measured, since the SNR estimation unit is included in every ACM-based DVB-S2 and CMMB decoding. Therefore, it should not be regarded as an additional overhead in the proposed approach.

As shown in Figure 12, the smaller the SNR is, the greater number of iterations the decoding requires. It is obvious that repeated computation of the parity-check equation and the tentative decision will lead to high latency and power consumption. If we compute these functions only if the SNR estimation falls within a certain range, then we can avoid excessive power consumption due to unnecessary parity checks and tentative decisions.

Table 4 Synthesis results

Technology	Chartered 0.18 μm CMOS
LDPC total (in NAND2)	256 K
AGU (in NAND2, with memory)	82 K
Parity check (in NAND2)	1 K
Tentative (in NAND2)	1 K
SNR estimator (in NAND2)	31 K
Operating freq. (MHz)	188
Voltage (V)	1.8

For example, according to Table 1 the average number of iterations is 7.75 when the SNR is 2.5 dB. Up to the fourth iteration, we may omit parity-check and tentative operations since we have no performance degradation. When we perform parity-check and tentative operations at every iteration, the amount of power consumption is 256.63 mW. When we skip parity-check and tentative operations until the fourth iteration, the amount of power consumption becomes 235.43 mW. Therefore, the amount of power consumption is reduced by 21.10 mW. We also observe that there is a significant reduction in power consumption for the SNR range of 2-6 dB. Especially, approximately 10% of power consumption is reduced for the SNR range of 1.5-2.5 dB.

Next, we discuss the effectiveness of the proposed algorithm for reducing the power consumption for DVB-S2. We implemented DVB-S2 in hardware as described in [20,21]. Figure 13 shows the reduction of power consumption after the novel algorithm is applied. As shown in Figure 14, the effectiveness of power

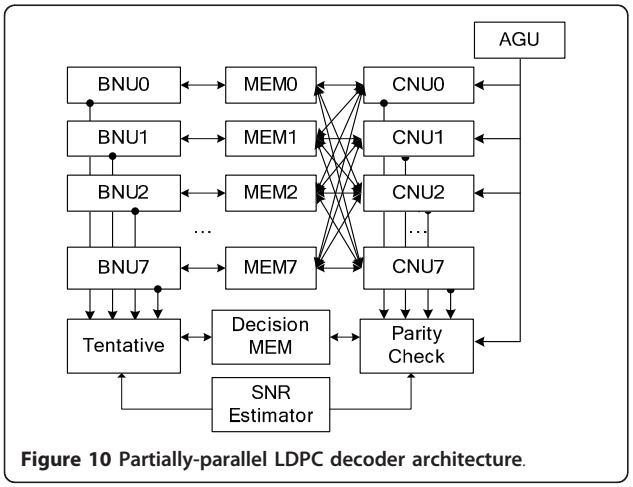

Figure 10 Partially-parallel LDPC decoder architecture.

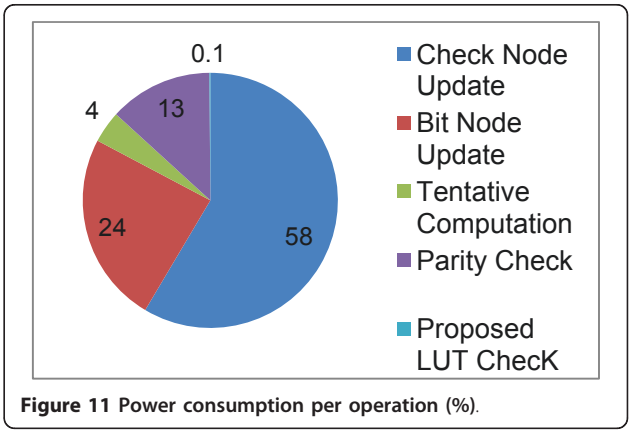

Figure 11 Power consumption per operation (%).

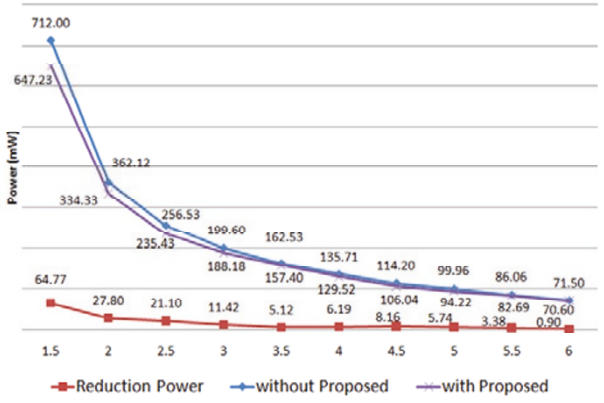

Figure 12 Partially-parallel LDPC decoder architecture SNR vs. power consumption.

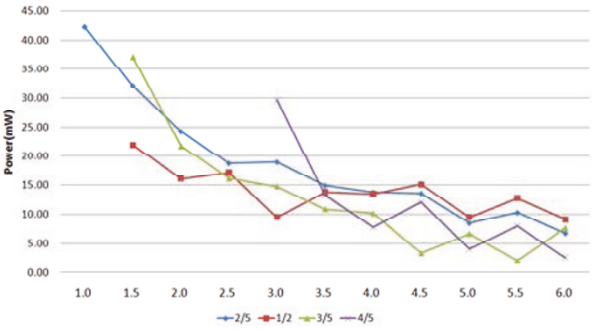

Figure 14 DVB-S2 short length LDPC decoder architecture SNR vs. reduction of power consumption.

reduction for DVB-S2 at various data rates is very good, even at high SNR values.

6. Conclusion

In this article, we propose a novel adaptive parity-check decoding scheme based on SNR estimation. Our proposed scheme does not require any additional hardware. We observe that the iteration count until the completion of LDPC decoding can be predicted by the SNR value. Therefore, we may omit computation of the parity check and the tentative decision if the SNR value is too low to lead to successful decoding, which, in turn, reduces power consumption. Experimental results show that significant amounts of power reduction may be realized when SNR values are low. We expect that by applying this algorithm to the design of mobile devices with digital broadcasting chips, we can increase their battery life considerably.

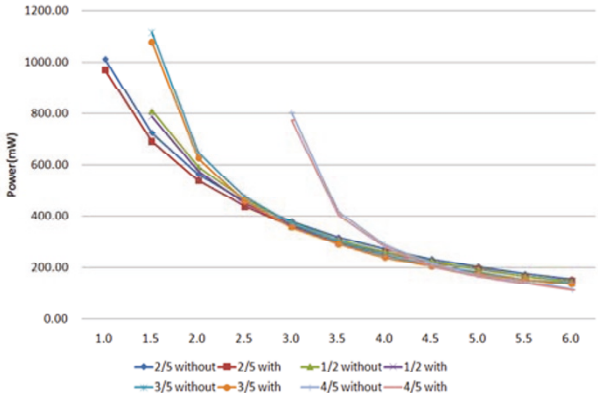

Figure 13 DVB-S2 short length LDPC decoder architecture SNR vs. power consumption.

Algorithm 1. LLR decoding algorithm

1: {Initialization :}

Set iteration number.i = 0., and F_n (LLR) for bit nodes (n = 1,2,...,N)

and for each (m, n) if H_{mn} = 1 set $Z_{mn} = F_n$ (1)

2: while $i \le i_{\max}$

3: for all (the check node) do

where each set (m,n) if H_{mn} = 1

$$T_{mn} = \prod_{n' \in N(m) \setminus n} \frac{1 - \exp(z_{mn'})}{1 + \exp(z_{mn'})} \tag{2}$$

$$L_{mn} = \ln \frac{1 - T_{mn}}{1 + T_{mn}} \tag{3}$$

4: end for

5: for all (the bit node) do

where each set (m,n) if H_{nm} = 1 Update

$$z_{mn} = F_n + \sum_{m' \in M(n) \setminus m} L_{m'n} \tag{4}$$

$$z_n = F_n + \sum_{m \in M(n)} L_{mn} \tag{5}$$

6: end for

7: for all $\hat{c}_n$ for (n = 1,2,...,N) Compute all the tentatives do

$$\hat{c} = [\hat{c}_n] \ , \ \begin{cases} \hat{c}_n = 1 & \text{if } z_n > 0 \\ \hat{c}_n = 0 & \text{if } z_n < 0 \end{cases} \tag{6}$$

8: end for

9: for all $\hat{c}_n$ for (n = 1,2,...,N) Parity Check do

$$H \cdot \left[\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_N\right]^T = \begin{cases} 0 & \text{return sucess} \\ 1 & \text{continue } i+ \end{cases} \quad (7)$$

10: end for
11: end while

Algorithm 2. Using adaptive parity check for modified UMP-BP

1: {Initialization :}
Set iteration number i = 0, and F_n (LLR) for bit nodes (n = 1,2,...,N)
and for each (m,n) if H_{mn} = 1 set $Z_{mn} = F_n$ (10)
2: while $i \le i_{\max}$
3: for all (the check node) do
where each set (m, n) if H_{mn} = 1

$$L_{mn} = \left(\prod_{n' \in N(m)\backslash n} \operatorname{sign}(Z_{mn'})\right) \cdot \min_{n' \in N(m)\backslash n} |Z_{mn'}| \cdot \alpha \quad (11)$$

4: end for
5: for all (the bit node) do
where each set (m, n) if H_{mn} = 1 Update

$$z_{mn} = F_n + \sum_{m' \in M(n)\backslash m} L_{m'n} \quad (12)$$

$$z_n = F_n + \sum_{m \in M(n)} L_{mn} \quad (13)$$

6: end for
7: if (i > SNR Estimation_Table)
8: for all $\hat{c}_n$ for (n = 1,2,...,N) Compute all the tentative do

$$\hat{c} = \left[\hat{c}_n\right] \ , \ \begin{cases} \hat{c}_n = 1 & \text{if } z_n > 0 \\ \hat{c}_n = 0 & \text{if } z_n < 0 \end{cases} \quad (14)$$

9: end for
10: for all $\hat{c}_n$ for (n = 1,2,...,N) Parity Check do

$$H \cdot \left[\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_N\right]^T = \begin{cases} 0 & \text{return sucess} \\ 1 & \text{continue } i+ \end{cases} \quad (15)$$

11: end for
12: end if
13: else if

continue i+

14: end if
15: end while

Abbreviations

ACM: adaptive coding and modulation; LDPC: low density parity check; MCS: modulation and coding scheme; ML: maximum likelihood; MSE: mean square error; SNV: signal-to-noise variance; SOF: start of frame.

Acknowledgements

"This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)" (NIPA-2011-C1090-1100-0010)

Competing interests

The authors declare that they have no competing interests.

Received: 11 November 2010 Accepted: 31 July 2011
Published: 31 July 2011

References

1. Standardization Roadmap for IT839 StrategyVer (2007)
2. R Gallager, Low-density parity-check codes. IRE Trans Inf Theory **IT-8**, 21–28 (1962)
3. D MacKay, R Neal, Near Shannon limit performance of low density parity check codes IEE Electron Lett. **32**(18), 1645–1646 (1996). doi:10.1049/el:19961141
4. D MacKay, Good error-correcting codes based on very sparse matrices. IEEE Trans Inf Theory **45**(2), 399–431 (1999). doi:10.1109/18.748992
5. S-Y Chung, GD Forney Jr, TJ Richardson, R Urbanke, On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. IEEE Commun Lett. **5**, 58–60 (2001)
6. J Chen, A Dholakia, E Eleftheriou, MPC Fossorier, X Hu, Reduced-complexity decoding of LDPC codes. IEEE Trans Commun. **53**, 1288–1299 (2005). doi:10.1109/TCOMM.2005.852852
7. A Blankby, C Howland, A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder. IEEE J Solid-State Circuits **37**(3), 404–412 (2002). doi:10.1109/4.987093
8. ETSI, Digital Video Broadcasting (DVB), Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications. ETIS EN 302 307 v1.1.2, june 2006
9. V Sorokine, F Kschischang, S Pasupathy, Gallager codes for CDMA applications—part II: implementations, complexity, and system capacity. IEEE Trans Commun. **48**(11), 1818–1828 (2000). doi:10.1109/26.886472
10. S Hong, W Stark, Design and implementation of a low complexity VLSI turbo-code decoder architecture for low energy mobile wireless communications. J VLSI Signal Process Syst. **24**(1), 43–57 (2000). doi:10.1023/A:1008114510995
11. S Kim, GE Sobelman, H Lee, R Nicole, Adaptive quantization in min-sum based irregular LDPC decoder, in *Circuits and Systems, ISCAS*, 536–539 (2008)
12. A Darabiha, AC Carusone, FR Kshichang, Power reduction techniques for LDPC decoders. IEEE Trans VLSI Syst. **43**, 1835–1845 (2008)
13. S Lin, DJ Costello Jr, *Error Control Coding*, 2nd edn. (Prentice-Hall, Englewood Cliffs, 2004)
14. G Albertazzi, S Cioni, GE Corazza, M Neri, R Pedone, P Salmi, A Vanelli-Coralli, M Villanti, On the adaptive DVB-S2 physical layer: design and performance Wireless Commun IEEE. **12**(6), 62–68 (2005). doi:10.1109/MWC.2005.1561946
15. DR Pauluzzi, NC Beaulieu, A comparison of SNR estimation techniques for the AWGN channel. IEEE Trans Commun. **48**(10), 1681–1691 (2000)
16. http://www.cmmb.org.cn/
17. ETSI EN 302 307, Digital Video Broadcasting (DVB), Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and otherbroadband satellite applications
18. M Smolnikar, M Mohorcic, T Javornik, Utilisation of LDPC decoder parameters in DVB-S2 ACM procedures, in *IWSSC '07*, 194–198 (2007)
19. S-J Lee, J-Y Park, K-S Chung, Memory efficient multi-rate regular LDPC decoder for CMMB. IEEE Trans Consumer Electron. **55**(4), 1866–1874 (2009)
20. P Urard, E Yeo, L Paumier, P Georgelin, T Michel, V Lebars, E Lantreibecq, B Gupta, A 135 Mb/s DVB-S2 compliant codecbased on 64800b LDPC and BCH codes, in *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC 2005 IEEE International*, 1, 446–609 (2005)

21. P Urard, L Paumier, V Heinrich, N Raina, N Chawla, A 360 mW 105 Mb/s DVB-S2 compliant codec based on 64800b LDPC and BCH codes enabling satellite- transmission portable devices, in *Solid-State Circuits Conference, 2008. Digest of Technical Papers ISSCC 2008 IEEE International,* 310–311 (2005)

doi:10.1186/1687-1499-2011-48
Cite this article as: Park and Chung: **An adaptive low-power LDPC decoder using SNR estimation.** *EURASIP Journal on Wireless Communications and Networking* 2011 **2011**:48.