변수노드의 차수가 3인 정칙 LDPC 부호의 Absorbing 집합에 관하여

정회원 이 기 준*, 종신회원 정 하 봉**, 정회원 장 환 석**

On the Absorbing Sets of the Regular LDPC Codes with Variable Node Degree Three

Kijun Lee* *Regular Member*, Habong Chung** *Lifelong Member*,
Hwanseok Jang** *Regular Member*

요 약

본 논문에서는 변수노드의 차수가 3인 정칙 LDPC 부호의 absorbing 집합(AS)에 대한 여러 가지 성질을 알아본다. Girth가 인 경우, 가장 작은 absorbing 집합은 AS임을 보였고, 인 경우 이 AS가 가장 작은 fully absorbing 집합임을 증명하였다. 또한 (3,5) Tanner LDPC 부호에서 오류 마루의 주원인으로 꼽히는 몇몇 absorbing 집합의 critical number를 유도하였다.

Key Words : Absorbing Set, Tanner (3,5) LDPC Codes, Fully Absorbing Set, Critical Number

ABSTRACT

In this paper, we investigate into the existence and various properties of absorbing sets in regular LDPC codes with variable node degree three. Also, we figure out the critical number of some absorbing sets that are believed to be the major cause of error floor in (3,5) Tanner LDPC codes.

Ⅰ. Introduction

The typical performance curve of the code is referred as so-called waterfall curve. Occasionally the performance curve of the codes using iterative decoding such as LDPC codes and Turbo codes shows the rapid slope decline in high SNR region called error floor. In the case of Turbo codes, the existence of the low-weight codewords is known as the main cause of the error floor [1]. But in the case of LDPC codes, it is hard to explain this error floor just with the low-weight codewords of an LDPC code [2].

In many previous studies, this undesirable behavior of LDPC codes in iterative decoding have been attributed to the combinatorial structures of the codes. They are referred as the stopping sets in BEC [3] and the trapping sets in AWGN channel [2,4,5]. Also the concept of pseudo-codewords [6,7] due to the property of the locally operated iterative decoder has been attributed to this phenomenon.

※ "This work is supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD)" (KRF-2006-311-D00656)
* 삼성전자 Device Solution 부문 (kijuni.lee@samsung.com),
** 홍익대학교 전자공학과 (habchung@hongik.ac.kr, hsjang@mail.hongik.ac.kr)
논문번호 : KICS2009-05-225, 접수일자 : 2009년 5월 30일, 최종논문접수일자 : 2009년 10월 19일

It has been shown that the special form of trapping sets called absorbing sets can be the main reason of the error floor in some structured LDPC codes. In [8] and [9], those problematic absorbing sets were revealed through computer simulations.

Since it is not easy to analytically identify the absorbing sets of a given code, most of the previous studies use simulation to show the existence of small absorbing sets. Upto now, the only analytic approach was done by Dolecec et al. [10]. They proved the existence of small absorbing sets in the array-type LDPC codes with the variable node degree of 2, 3 and 4.

In this paper, we investigate into the existence and various properties of absorbing sets in regular LDPC codes with variable node degree three. Also, we figure out the critical number of absorbing sets in (3,5) Tanner LDPC codes.

The paper is organized as follows. In section 2, we review some useful definitions and notations. The small absorbing sets in regular LDPC codes with variable node degree three are revealed in section 3. The critical number of an absorbing set is obtained in section 4 and the concluding remarks are in section 5.

Ⅱ. Preliminaries

Let H be the parity-check matrix of a code, V be the set of variable nodes, C be the set of check nodes, E be the set of edges $\{(i,j) \mid h_{ij} \neq 0\} \in V \times C$ and $G(H) = (V, C; E)$ be the Tanner graph of H. Let $G(D)$ be the induced subgraph corresponding to a subset $D \subset V$. The degree of a node is defined as the number of neighborhood nodes connected to it. Let $\epsilon(D)$ and $o(D)$ be the set of even- and odd-degree check nodes in $G(D)$, respectively. In the graph depictions in this paper, a circle represents a variable node and a square represents a check node. A filled circle means an erroneous variable node while an unfilled circle means a correct variable node. Similarly, a filled square means an unsatisfied check node(USCN) having a check-sum one and an unfilled square means a satisfied check node (SCN) having a check-sum zero.

Richardson [2] suggested that the main reason of the error floor is the "near-codeword".

Definition 1 (Near-codeword) [2]

A (w,v) near-codeword is defined as a binary vector $\underline{x}$ whose weight is w and the weight of its syndrome is v. □

When $\underline{x}$ is considered as an error vector, the Hamming weight v of the syndrome of $\underline{x}$ is the number of unsatisfied check-sums. Also in high SNR regime, most of the errors are likely to have small w. If we think the role of unsatisfied check-sums is to change the bit value in the iterative decoding, small v is not enough to correct the errors. Thus in high SNR regime, near-codewords with small w and v are thought to be as the main reason of the error floor [2].

Definition 2 (Trapping set) [2]

An (a,b) trapping set, abbreviated as (a,b) TS is defined as the set T of a variable nodes such that the order of the set $o(T)$ of odd-degree check nodes in the induced subgraph $G(T)$ is b. □

Let $\underline{x}$ be a vector of weight a such that ones are in the positions of the variable nodes of (a,b) TS and zeros in the rests. If the weight of the syndrome of $\underline{x}$ is b, then the vector $\underline{x}$ is regarded as an (a,b) near-codeword.

Definition 3 (Critical number) [8]

The critical number of trapping set T is the minimum number of the erroneous variable nodes that leads to a decoding failure, provided that the erroneous variable nodes are confined inside of T. If the critical number of T is $|T|$, T is called the minimal trapping set. □

Now, let us consider the role of check nodes in bit flipping algorithm. Assume that all-zero codeword is sent. If the number of erroneous variable nodes connected to a check node is odd, the check node is unsatisfied and will deliver the message that each variable node connected to it is wrong. On the other hand, if the number of erroneous variable nodes connected to a check node is even, the check is satisfied and will deliver the message that the variable nodes connected to it are right. Therefore during iterative decoding, the SCN cannot contribute to the change of the values of variable nodes. The combinatorial structure, called the "absorbing set", is relevant to this explanation. Let D be the subset of V. For $v \in D$, let $e_v(o_v)$ be the number of even (odd) degree connected to v in the induced subgraph $G(D)$, respectively.

Definition 4 (Absorbing set) [9]

An (a,b) absorbing set (AS) D is an (a,b) trapping set such that $e_v > o_v$, all $v \in D$. □

The tentative decision of the variable nodes in the bit flipping algorithm is the "majority vote" of the values from check nodes connected to it. In case that all variable nodes in absorbing set are in error, the erroneous values of the variable nodes in absorbing set remain unaltered. Therefore the absorbing set over BSC behaves like the stopping set over BEC in the sense that the errors will be not corrected any more.

Definition 5 (Fully absorbing set) [10]

An (a,b) fully absorbing set D is an (a,b) AS such that for all $v' \in V \backslash D$, the number of edges connected to $C \backslash o(D)$ from v' is larger than the number of edges connected to $o(D)$ from v'. □

The fully absorbing set comes from the notion that not only the interior but also the exterior variable nodes of an absorbing set should be taken into consideration in the iteration process. That is, while a non-fully absorbing set may propagate errors to the exterior variable nodes, the fully absorbing set doesn't propagate errors. In this sense, the fully absorbing set is a stable configuration. Certainly, every codeword is a fully absorbing set.

Ⅲ. The Smallest Absorbing Set

In this section, we find the smallest absorbing sets in a regular LDPC code with $d_v = 3$.

Lemma 6

In an (a,b) absorbing set D of (d_v,d_c) regular LDPC codes, we have $a-b \equiv 0$ (mod 2) if d_v is odd, and b is even if d_v is even.

Proof: The number of edges in the induced graph $G(D)$ is $d_v \cdot a$, since the degree of variable nodes is d_v. At the same time, the number of edges can be enumerated from check node perspective, and we have

$$d_v \cdot a = (\text{\# of edges from even degree check nodes}) + (\text{\# of edges from odd degree check nodes}).$$

Reducing the above equation modulo 2, we have

$$d_v \cdot a - b \equiv 0 \pmod{2}.$$

Therefore $a-b \equiv 0$ (mod 2) if d_v is odd and b is even otherwise. □

Lemma 7

In an (a,b) absorbing set D of a regular LDPC code with $d_v = 3$, $a \geq b$.

Proof: From the definition, we have $e_v > o_v$, for all $v \in D$. Thus in case of $d_v = 3$, o_v is at most one. Therefore $a \geq b$. □

The next theorem shows the general form of the absorbing sets in regular LDPC codes with $d_v = 3$.

Theorem 8

Let D be an (a,b) absorbing set of a regular LDPC code with $d_v=3$ and $a>2$. If all the variable nodes participate in a cycle in the induced graph, then $b=a$. (Resulting absorbing set is depicted in Fig. 1)

Proof: It is manifest that the maximum length of a cycle in a bipartite graph containing a variable nodes cannot exceed $2a$. Since $a \geq b$ from Lemma 7, and we have an example of (a,a) absorbing set with $2a$-cycle in Fig. 1, it is enough to show that there must exist a cycle of length less than $2a$ in the induced graph $G(D)$ if $b<a$.

Since $d_v=3$ and all the variable nodes participate in a cycle, the degree of any odd degree check node in $G(D)$ must be one. Thus, $b<a$ implies that there is at least a variable node v with $e_v=3$ and $o_v=0$. Let c_1, c_2, and c_3 be 3 neighboring check nodes of degree 2. Since v must participate in the cycle of length $2a$, this cycle of length $2a$ contains 2 of the 3 neighboring check nodes, namely c_2 and c_3. Let v' be the other variable node connected to the remaining one check node c_1. Since v' also participates in the cycle of length $2a$, there is a segment from v to v' of length less than or equal to a in the cycle. Thus, adjoining this segment with path v-c_1-v' makes another cycle of length less than or equal to $a+2$. Finally, the fact that $a+2<2a$ proves the theorem. □

The following two corollaries are the immediate consequences of Theorem 8.

Corollary 9

For a regular LDPC code with $d_v=3$ and with girth g, the smallest absorbing set is a $(g/2,g/2)$ absorbing set and there is a cycle with length g in the induced graph corresponding to this absorbing set.

Corollary 10

In a regular LDPC code with $d_v=3$ and with girth g, there is no (a,b) absorbing set such that $a \leq g/2,\ b<g/2$.

The next theorem tells us about the conditions for an $(g/2,g/2)$ AS to be a fully absorbing set.

Theorem 11

In a regular LDPC code with $d_v=3$ and girth g, the smallest absorbing set $(g/2,g/2)$ AS is a fully absorbing set if $g>8$.

Proof: Let D be the $(g/2,g/2)$ absorbing set. Assume that D is not a fully absorbing set. Then, there exist at least one variable node v' $\in V \setminus D$ which is connected to two or more odd degree (degree one) check nodes in $o(D)$. Assume that these two odd degree check nodes are connected to the variable nodes, v_i and v_j in D, respectively. (See Fig. 2.) Because v_i and v_j participate in the cycle of length g, there must

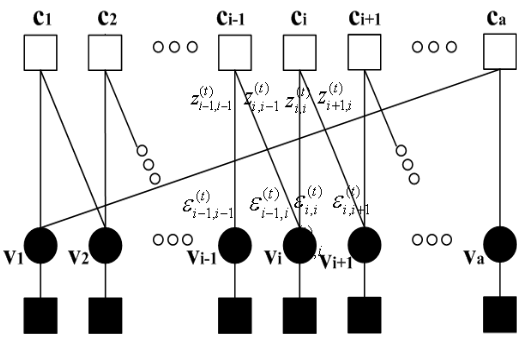

Fig. 1. (a,a) AS with $2a$-cycle

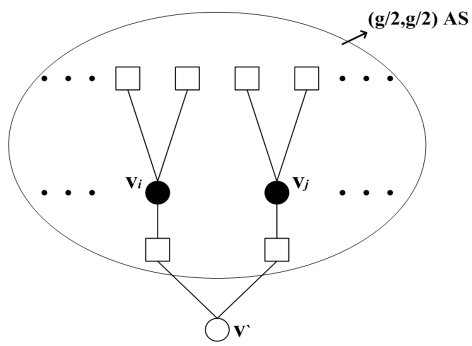

Fig. 2. For the proof of Theorem 11.

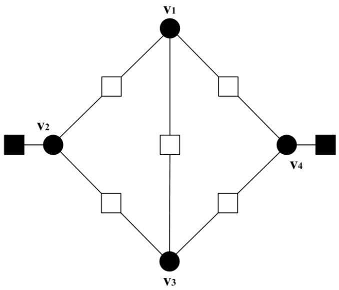

Fig. 3. A (4,2) TS (AS)

exist a segment from v_i to v_j whose length is less than or equal to $g/2$. Adjoining this segment to the path v_i-v'-v_j forms a cycle of length less than or equal to $g/2+4$. Since the girth is g, it leads to a contradiction if $g/2+4<g$, which in turn becomes $g>8$. □

In most of the LDPC codes in practice, the girth g is 6 or 8. In this case, the $(g/2\,,g/2)$ AS is the smallest, but may not be the smallest fully absorbing set. In fact, in an array-type LDPC with $d_v=3$, the $(3,3)$ AS is not the smallest fully absorbing set. The smallest fully absorbing set in this case is a $(4,2)$ AS shown in Fig. 3 which can be considered as the merged form of two $(3,3)$ AS's. In a non-fully absorbing set, the error bits inside the absorbing set may propagate to the variable nodes outside the absorbing set during bit flipping decoding.

Theorem 12

In (3,5) Tanner LDPC codes,

① A $(4,4)$ AS is the smallest when $p=31$.

② A $(5,5)$ AS is the smallest and a fully absorbing set when $p=61$ and 151.

③ A $(6,6)$ AS is the smallest and a fully absorbing set when $p\geqq 181$.

Proof: In [11] the girth of a (3,5) Tanner LDPC code is derived and it is as follows.

$$g=\begin{cases}8, & \text{if } p=31\\ 10, & \text{if } p=61,151\\ 12, & \text{if } p\geqq 181\end{cases}$$

Applying Corollary 9 and Theorem 11, we can prove the theorem. □

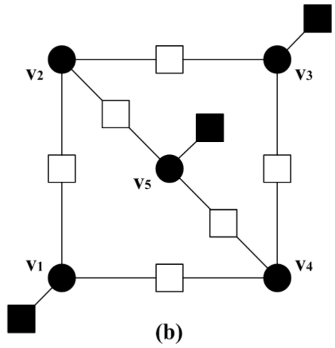

Fig. 4. (5,3) AS: the smallest fully absorbing set in (3,5) Tanner LDPC codes with $p=31$.

In a regular LDPC code with $d_v=3$, a g-cycle always corresponds to a $(g/2,g/2)$ AS. Thus, from Theorems 8 and 12, the $(g/2,g/2)$ AS is the smallest fully absorbing set for (3,5) Tanner LDPC codes with $p\geq 61$. As one can see in the next section, the critical number is $g/2$, i.e., the minimum number of uncorrectable errors in this code is $g/2$. Thus, it is fairly reasonable to say that minimum length cycles are the major reason of the error floor. For the case of $p=31$, the $(4,4)$ AS is not a fully absorbing set. Analysing the 8-cycles given in [11], we can verify that the smallest fully absorbing set is the $(5,3)$ AS shown in Fig. 4, which can be made of three 8-cycles sharing three variable nodes.

Ⅳ. Critical Number of an Absorbing Set

Now, consider the case when some (not all) variable nodes in the absorbing set are in error. For example, if the variable nodes, v_1, v_2, and v_3 of a $(4,2)$ AS in Fig. 3 are in error, then decoding fails. The variable nodes v_1, v_2, and v_3 make a $(3,3)$ AS. Thus the critical number of the $(4,2)$ AS is three. In an effort to lower the error floor by breaking small absorbing sets [12], knowing the critical number of an absorbing set is important. In this section, we derive the critical number of an (a,a) AS and a $(5,3)$ AS in (3,5) Tanner LDPC codes.

Theorem 13

In a regular LDPC code with $d_v = 3$, the critical number of an (a,a) absorbing set is a.

Proof: Let $v_0, v_1, v_2, \cdots, v_{a-1}$ be the variable nodes consisting of an (a,a) absorbing set D and c_k, $0 \le k \le a-1$ be the degree-2 check nodes in the absorbing set connected to the variable nodes v_k and v_{k+1}. Assume without loss of generality that the configuration of the (a,a) absorbing set is given Fig. 1. Assume that all-zero codeword is sent and errors in the received vector are confined only in the absorbing set.

Let $v_i^{(t)}, 0 \le i \le a-1$, be the tentative decoded value of the variable node v_i after the t-th iteration. Let $z_{i,i}^{(t)}$ and $z_{i,i-1}^{(t)}$ be the message from v_i to c_i and c_{i-1}, respectively during the t-th iteration. Let $\epsilon_{i,i}^{(t)}$ and $\epsilon_{i,i+1}^{(t)}$ be the message from c_i to v_i and v_{i+1}, respectively during the t-th iteration. Note that $v_i^{(0)} \left(= z_{i,i}^{(1)}\right)$ is the initial value of v_i received from the channel. Here, an iteration is defined to be a variable node update followed by a check node update. From the fact that all c_i's are of degree 2, we have $\epsilon_{i-1,i}^{(t)} = z_{i-1,i-1}^{(t)}$ and $\epsilon_{i,i}^{(t)} = z_{i+1,i}^{(t)}$. Since the tentative decoding is a majority vote, $v_i^{(t)}$ can be expressed as

$$\begin{aligned} v_i^{(t)} &= maj\left\{0, \epsilon_{i-1,i}^{(t)}, \epsilon_{i,i}^{(t)}\right\} \\ &= maj\left\{0, z_{i-1,i-1}^{(t)}, z_{i+1,i}^{(t)}\right\} \end{aligned} \tag{1}$$

where the subscripts are of arithmetic modulo a. In Gallager B algorithm, variable node updates can be expressed as

$$\begin{aligned} z_{i-1,i-1}^{(t)} &= maj\left\{0, v_{i-1}^{(0)}, \epsilon_{i-2,i-1}^{(t-1)}\right\} \\ &= maj\left\{0, v_{i-1}^{(0)}, z_{i-2,i-2}^{(t-1)}\right\} \end{aligned} \tag{2}$$

and

$$\begin{aligned} z_{i+1,i}^{(t)} &= maj\left\{0, v_{i+1}^{(0)}, \epsilon_{i+1,i+1}^{(t-1)}\right\} \\ &= maj\left\{0, v_{i+1}^{(0)}, z_{i+2,i+1}^{(t-1)}\right\} \end{aligned} \tag{3}$$

From the truth table of a majority vote, we have $maj\{0, a, b\} = a\,b$. Thus, by plugging (2) and (3) into (1), (1) can be rewritten as

$$v_i^{(t)} = z_{i-1,i-1}^{(t)}\, z_{i+1,i}^{(t)} = \left(v_{i-1}^{(0)}\, z_{i-2,i-2}^{(t-1)}\right)\left(v_{i+1}^{(0)}\, z_{i+2,i+1}^{(t-1)}\right)$$

Applying (2) and (3) recursively, we have the following expression.

$$v_i^{(t)} = \left(\prod_{k=1}^{m} v_{i-k}^{(0)}\, v_{i+k}^{(0)}\right) z_{i-m-1,i-m-1}^{(t-m)}\, z_{i+m+1,i+m}^{(t-m)}$$

Especially for $t = a$ and $m = a-1$, the above expression can be rewritten as

$$v_i^{(a)} = \left(\prod_{k=1}^{a-1} v_{i+k}^{(0)}\right) z_{i,i}^{(1)} = \prod_{k=0}^{a-1} v_k^{(0)}$$

Thus, unless all the initial received values are in error, decoding is completed after at least a-th iteration. Therefore, the critical number of an absorbing set is a. □

As mentioned before, the smallest fully absorbing set in a (3,5) Tanner LDPC code with $p = 31$ is the (5,3) AS shown in Fig. 4. The next theorem tells us that its critical number is 3.

Theorem 14

The critical number of a $(5,3)$ AS in a (3,5) Tanner LDPC code with $p = 31$ is 3.

Proof: Let the notations for variable nodes and check nodes and their update messages in a $(5,3)$ AS be shown in Fig. 5. Assuming that no errors occur outside of the $(5,3)$ AS, the check node updates and variable node updates at iteration t can be expressed as follows:

$$\begin{aligned} &\epsilon_{12}^{(t)} = z_{11}^{(t)}, \epsilon_{11}^{(t)} = z_{21}^{(t)}, \epsilon_{22}^{(t)} = z_{32}^{(t)}, \epsilon_{23}^{(t)} = z_{22}^{(t)}, \epsilon_{33}^{(t)} = z_{43}^{(t)}, \\ &\epsilon_{34}^{(t)} = z_{33}^{(t)}, \epsilon_{41}^{(t)} = z_{44}^{(t)}, \epsilon_{44}^{(t)} = z_{14}^{(t)}, \epsilon_{52}^{(t)} = z_{55}^{(t)}, \epsilon_{55}^{(t)} = z_{25}^{(t)}, \\ &\epsilon_{64}^{(t)} = z_{56}^{(t)}, \epsilon_{65}^{(t)} = z_{46}^{(t)} \end{aligned}$$

$z_{ij}^{(1)} = v_i^{(0)}$ for all i and j. and for $t \ge 2$,

$$\begin{aligned} z_{21}^{(t)} &= maj\{V_2^{(0)}, \epsilon_{22}^{(t-1)}, \epsilon_{52}^{(t-1)}\} = maj\{V_2^{(0)}, z_{32}^{(t-1)}, z_{55}^{(t-1)}\} \\ z_{44}^{(t)} &= maj\{V_4^{(0)}, \epsilon_{64}^{(t-1)}, \epsilon_{34}^{(t-1)}\} = maj\{V_4^{(0)}, z_{56}^{(t-1)}, z_{33}^{(t-1)}\} \\ z_{11}^{(t)} &= maj\{0, V_1^{(0)}, \epsilon_{41}^{(t-1)}\} = maj\{0, V_1^{(0)}, z_{44}^{(t-1)}\} \end{aligned}$$

$$
\begin{aligned}
z_{32}^{(t)} &= maj\{0, V_3^{(0)}, \epsilon_{33}^{(t-1)}\} = maj\{0, V_3^{(0)}, z_{43}^{(t-1)}\} \\
z_{55}^{(t)} &= maj\{0, V_5^{(0)}, \epsilon_{65}^{(t-1)}\} = maj\{0, V_5^{(0)}, z_{46}^{(t-1)}\} \\
z_{22}^{(t)} &= maj\{V_2^{(0)}, \epsilon_{12}^{(t-1)}, \epsilon_{52}^{(t-1)}\} = maj\{V_2^{(0)}, z_{11}^{(t-1)}, z_{55}^{(t-1)}\} \\
z_{43}^{(t)} &= maj\{V_4^{(0)}, \epsilon_{44}^{(t-1)}, \epsilon_{64}^{(t-1)}\} = maj\{V_4^{(0)}, z_{14}^{(t-1)}, z_{56}^{(t-1)}\} \\
z_{33}^{(t)} &= maj\{0, V_3^{(0)}, \epsilon_{23}^{(t-1)}\} = maj\{0, V_3^{(0)}, z_{22}^{(t-1)}\} \\
z_{14}^{(t)} &= maj\{0, V_1^{(0)}, \epsilon_{11}^{(t-1)}\} = maj\{0, V_1^{(0)}, z_{21}^{(t-1)}\} \\
z_{56}^{(t)} &= maj\{0, V_5^{(0)}, \epsilon_{55}^{(t-1)}\} = maj\{0, V_5^{(0)}, z_{25}^{(t-1)}\} \\
z_{25}^{(t)} &= maj\{V_2^{(0)}, \epsilon_{12}^{(t-1)}, \epsilon_{22}^{(t-1)}\} = maj\{V_2^{(0)}, z_{11}^{(t-1)}, z_{32}^{(t-1)}\} \\
z_{46}^{(t)} &= maj\{V_4^{(0)}, \epsilon_{44}^{(t-1)}, \epsilon_{34}^{(t-1)}\} = maj\{V_4^{(0)}, z_{14}^{(t-1)}, z_{33}^{(t-1)}\}
\end{aligned}
$$

Similarly, the value at tentative decision of each variable nodes are as follows:

$$
\begin{aligned}
V_1^{(t)} &= maj\{0, \epsilon_{11}^{(t)}, \epsilon_{41}^{(t)}\} = maj\{0, z_{21}^{(t)}, z_{44}^{(t)}\} = z_{21}^{(t)} z_{44}^{(t)} \\
V_2^{(t)} &= maj\{\epsilon_{12}^{(t)}, \epsilon_{52}^{(t)}, \epsilon_{22}^{(t)}\} = maj\{z_{11}^{(t)}, z_{55}^{(t)}, z_{32}^{(t)}\} \\
V_3^{(t)} &= maj\{0, \epsilon_{23}^{(t)}, \epsilon_{33}^{(t)}\} = maj\{0, z_{22}^{(t)}, z_{43}^{(t)}\} = z_{22}^{(t)} z_{43}^{(t)} \\
V_4^{(t)} &= maj\{\epsilon_{34}^{(t)}, \epsilon_{44}^{(t)}, \epsilon_{64}^{(t)}\} = maj\{z_{33}^{(t)}, z_{14}^{(t)}, z_{56}^{(t)}\} \\
V_5^{(t)} &= maj\{0, \epsilon_{55}^{(t)}, \epsilon_{65}^{(t)}\} = maj\{0, z_{25}^{(t)}, z_{46}^{(t)}\} = z_{25}^{(t)} z_{46}^{(t)}
\end{aligned}
$$

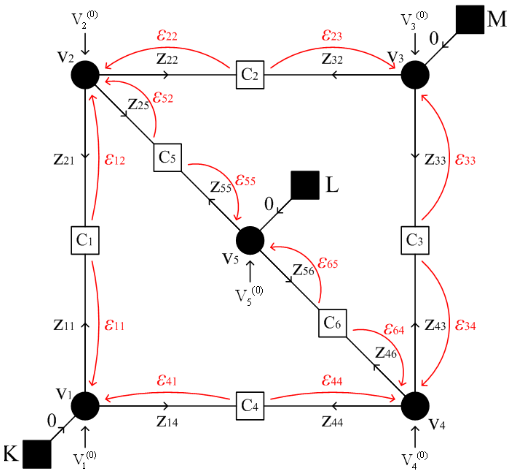

Fig. 5. Updates in a $(5,3)$ AS

Combining these expressions, one can obtain the value of each variable node at tentative decision in terms of the initially received value. For example, the values for V_i at iteration $t=2$ are;

$$
\begin{aligned}
V_1^{(2)} &= z_{21}^{(2)} z_{44}^{(2)} \\
&= maj\{V_2^{(0)}, V_3^{(0)}, V_5^{(0)}\} maj\{V_4^{(0)}, V_3^{(0)}, V_5^{(0)}\} \\
V_2^{(2)} &= maj\{z_{11}^{(2)}, z_{55}^{(2)}, z_{32}^{(2)}\} \\
&= maj\{(V_1^{(0)} V_4^{(0)}), (V_5^{(0)} V_4^{(0)}), (V_3^{(0)} V_4^{(0)})\}
\end{aligned}
$$

$$
\begin{aligned}
V_3^{(2)} &= z_{22}^{(2)} z_{43}^{(2)} \\
&= maj\{V_2^{(0)}, V_1^{(0)}, V_5^{(0)}\} maj\{V_4^{(0)}, V_1^{(0)}, V_5^{(0)}\} \\
V_4^{(2)} &= maj\{z_{33}^{(2)}, z_{14}^{(2)}, z_{56}^{(2)}\} \\
&= maj\{(V_3^{(0)} V_2^{(0)}), (V_1^{(0)} V_2^{(0)}), (V_5^{(0)} V_2^{(0)})\} \\
V_5^{(2)} &= z_{25}^{(2)} z_{46}^{(2)} \\
&= maj\{V_2^{(0)}, V_1^{(0)}, V_3^{(0)}\} maj\{V_4^{(0)}, V_1^{(0)}, V_3^{(0)}\}
\end{aligned}
$$

Note that $V_i^{(2)} = V_i^{(0)}$, $1 \le i \le 5$, when the initial received values are $V_1^{(0)} = V_3^{(0)} = V_5^{(0)} = 1$ and $V_2^{(0)} = V_4^{(0)} = 0$.

In fact, under the same initial received values of $V_1^{(0)} = V_3^{(0)} = V_5^{(0)} = 1$ and $V_2^{(0)} = V_4^{(0)} = 0$, we have

$$
\begin{aligned}
V_1^{(t)} &= z_{21}^{(t)} z_{44}^{(t)} = z_{32}^{(t-1)} z_{55}^{(t-1)} z_{56}^{(t-1)} z_{33}^{(t-1)} \\
&= z_{43}^{(t-2)} z_{46}^{(t-2)} z_{25}^{(t-2)} z_{22}^{(t-2)} = V_3^{(t-2)} V_5^{(t-2)} \\
V_3^{(t)} &= z_{22}^{(t)} z_{43}^{(t)} = z_{11}^{(t-1)} z_{55}^{(t-1)} z_{56}^{(t-1)} z_{14}^{(t-1)} \\
&= z_{44}^{(t-2)} z_{46}^{(t-2)} z_{25}^{(t-2)} z_{21}^{(t-2)} = V_1^{(t-2)} V_5^{(t-2)} \\
V_5^{(t)} &= z_{25}^{(t)} z_{46}^{(t)} = z_{11}^{(t-1)} z_{32}^{(t-1)} z_{14}^{(t-1)} z_{33}^{(t-1)} \\
&= z_{44}^{(t-2)} z_{43}^{(t-2)} z_{21}^{(t-2)} z_{22}^{(t-2)} = V_1^{(t-2)} V_3^{(t-2)}
\end{aligned}
$$

and

$$
\begin{aligned}
V_2^{(t)} &= z_{11}^{(t)} z_{55}^{(t)} + z_{11}^{(t)} z_{32}^{(t)} + z_{55}^{(t)} z_{32}^{(t)} \\
&= z_{44}^{(t-1)} z_{46}^{(t-1)} + z_{44}^{(t-1)} z_{43}^{(t-1)} + z_{46}^{(t-1)} z_{43}^{(t-1)} \\
&= z_{56}^{(t-2)} z_{33}^{(t-2)} z_{14}^{(t-2)} = V_4^{(t-2)} \\
V_4^{(t)} &= z_{33}^{(t)} z_{14}^{(t)} + z_{33}^{(t)} z_{56}^{(t)} + z_{14}^{(t)} z_{56}^{(t)} \\
&= z_{22}^{(t-1)} z_{21}^{(t-1)} + z_{22}^{(t-1)} z_{25}^{(t-1)} + z_{21}^{(t-1)} z_{25}^{(t-1)} \\
&= z_{11}^{(t-2)} z_{55}^{(t-2)} z_{32}^{(t-2)} = V_2^{(t-2)}
\end{aligned}
$$

The final equalities of the last two equations are a little tricky, since $V_4^{(t-2)}$ is not $z_{56}^{(t-2)} z_{33}^{(t-2)} z_{14}^{(t-2)}$ but $maj\{z_{56}^{(t-2)} z_{33}^{(t-2)} z_{14}^{(t-2)}\}$. But with the given initial conditions, $z_{56}^{(t)}$, $z_{33}^{(t)}$, and $z_{14}^{(t)}$ always take value 1 so that $maj\{z_{56}^{(t-2)} z_{33}^{(t-2)} z_{14}^{(t-2)}\}$ is same as $z_{56}^{(t-2)} z_{33}^{(t-2)} z_{14}^{(t-2)}$.

The above equations tell us that the tentative decoded values of variable nodes remain same at every other iteration. Therefore, with three initial errors at V_1, V_3, and V_5, the errors are trapped.

Now, the remaining part for proving the critical number is three is to show that any error patterns of weight less than or equal to 2 can be corrected during the iterative decoding, and this can be easily shown since the $(5,3)$ AS is of the merged form of two $(4,4)$ AS's whose critical number is 4. □

Theorem 14 tells us that the major cause of the error floor in (3,5) Tanner LDPC codes can be the errors in some three variable nodes in a (5,3) AS. The error trapping process in a $(5,3)$ AS with three initial errors at V_1, V_3, and V_5 is depicted in Fig. 6.

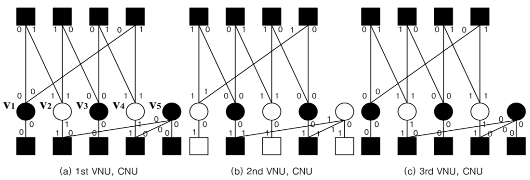

Fig. 6. Decoding process of (5,3) AS with errors at variable node v_1, v_3 and v_5

V. Conclusions

In this paper, we find out the smallest absorbing sets and the smallest fully absorbing sets in regular LDPC codes with $d_v = 3$. Also, we figure out the critical number of some absorbing sets that are believed to be the major cause of error floor in (3,5) Tanner LDPC codes.

References

〔1〕 S. Benedetto and G. Montorsi, "Unveiling turbo codes: Some results on parallel concatenated coding schemes," IEEE Trans. on Inform. Theory, vol. 42, no. 2, pp. 409-428, Mar. 1996.

〔2〕 T. Richardson, "Error floors of LDPC codes", in Proc. 41st Annual Conference on Communication, Control and Computing, pp.1426-1435, Sep. 2003.

〔3〕 C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," IEEE Trans. on Inform. Theory, Vol. 48, No. 6, pp. 1570-1579, June 2002.

〔4〕 D. Mackay and M. Postol, "Weaknesses of Margulis and Ramanujan-Margulis low-density parity-check codes," Electronic Notes in Theoretical Computer Science, Vol. 74, 2003.

〔5〕 S. Laendner and O. Milenkovic, "Algorithmic and combinatorial analysis of trapping sets in structured LDPC codes", In Proc. of Wireless Networks, Communication and Mobile Computing, Vol.1, pp. 630-635, Hawaii, USA, 2005.

〔6〕 P. Vontobel and R. Koetter, "Graph-cover decoding and finite-length analysis of message -passing iterative decoding of LDPC codes," submitted IEEE Trans. Inform. Theory, 2005.

〔7〕 C. Kelley and D. Sridhara, "Pseudocodewords of Tanner graphs," IEEE Trans. on Inform. Theory, Vol. 53, No. 11, pp. 4013-4038, Nov. 2007.

〔8〕 S. K. Chilappagari, S. Sankaranarayanan, and B. Vasic "Error floors of LDPC codes on the binary symmetric channel," in Proc. of IEEE ICC 2006, Vol.3, pp. 1089-1094, June 2006.

〔9〕 Z. Zhang, L. Dolec, B. Nikolic, V. Anantharam, and M. Wainwright, "Investigation of error floors of structured low-density parity-check codes by hardware emulation," in Proc. of GLOBECOM 2006, pp. 1-6, San Francisco, CA, Nov. 2006.

〔10〕 L. Dolecec, Z. Zhang, V. Anantharam, M. Wainwight, B. Nikolic, "Analysis of absorbing sets for array-based LDPC codes," in Proc. of ICC 2007, pp. 6261-6268, 24-28 June 2007.

〔11〕 S. Kim, J.-S. No, H. Chung, and D.J. Shin, "On the girth of Tanner's (3,5) quasi-cyclic LDPC codes," IEEE Trans. on Inform. Theory, Vol. 52, No. 4, pp. 1739-1744, Apr. 2006.

〔12〕 K. Lee, H. Chung, and H. Jang, "Two-stage decoding in Gallager B algorithm of LDPC codes using redundant parity-check equations," Proc. JCCI 2009, III-E-4, Kwangju, Korea, April 2009.

이 기 준 (Kijun Lee) 정회원

2002년 2월 홍익대학교 전자전기공학부 공학사
2004년 2월 홍익대학교 대학원 전자공학과 공학석사
2009년 2월 홍익대학교 대학원 전자공학과 공학박사
2009년 3월~현재 삼성전자 Device Solution 부문
<관심분야> 디지털통신, 오류정정부호, LDPC 부호, 반복복호

정 하 봉 (Habong Chung) 종신회원

1981년 2월 서울대학교 전자공학과 공학사
1985년 12월 미국 University of Southern California, 전기공학과 공학석사
1988년 8월 미국 University of Southern California, 전기공학과 공학박사
1988년 9월~1991년 8월 미국 뉴욕주립대 전기공학과 조교수
1991년 9월~현재 홍익대학교 전자전기공학부 교수
<관심분야> 부호 이론, 조합수학, 시퀀스 설계

장 환 석 (Hwanseok Jang) 정회원

2008년 2월 홍익대학교 전자전기공학부 공학사
2008년 3월~현재 홍익대학교 전자공학과 석사과정
<관심분야> 디지털통신, 오류정정부호, LDPC 부호