

## ECC 内蔵メモリマクロにおける DFT 技術

櫛田 桂一、大塚 伸朗、平林 修、武山 泰久

(株) 東芝 SoC 研究開発センター 〒212-8520 川崎市幸区堀川町 580-1

E-mail: [keiichi.kushida@toshiba.co.jp](mailto:keiichi.kushida@toshiba.co.jp), [nobuaki.otsuka@toshiba.co.jp](mailto:nobuaki.otsuka@toshiba.co.jp)

あらまし ECC 回路を内蔵したメモリマクロにおいて、ECC 搭載に伴うテストコスト増加を回避する DFT 技術を開発した。新しい方式ではハミングコードのデザインを工夫することによって、標準の BIST 回路のテストで全 ECC 内蔵メモリ回路をテストすることができる。この ECC 技術は 90nm プロセス技術を用いた 512Kb SRAM マクロに搭載され、その効果が確認された。

キーワード ECC, DFT, SRAM, ハミング行列

## DFT Techniques for Memory Macro with Built-in ECC

Keiichi, Kushida, Nobuaki Otsuka, Osamu Hirabayashi, and Yasuhisa Takeyama

SoC Research & Development Center, Toshiba Corporation, Kawasaki, Japan

E-mail: [keiichi.kushida@toshiba.co.jp](mailto:keiichi.kushida@toshiba.co.jp), [nobuaki.otsuka@toshiba.co.jp](mailto:nobuaki.otsuka@toshiba.co.jp)

**Abstract** DFT techniques to implement ECC circuitry on memory macro with no additional test cost are proposed. New methodology to design a hamming code matrix is used to achieve whole ECC system testing with standard memory BIST and conventional test sequence. The proposed ECC techniques are implemented in a 512Kb SRAM macro and demonstrated by hardware characterization with 90nm technology.

**Keyword** ECC, DFT, SRAM, hamming code matrix

### 1. 背景

集積回路の微細化や低電圧化が進むにつれて、宇宙線や  $\alpha$  線に起因するソフトエラーが問題になってきている。ソフトエラーとは、メモリチップ中をイオン化された粒子が通過することによって、メモリセル内のデータが反転して情報が失われる現象のことである。この現象はチップ内の回路を破壊することではなく、チップが再び使用できることからソフトエラーと呼ばれている。

このソフトエラー問題を解決する優れた解決策の一つとして、ECC (Error Correcting Code) 技術が注目されている。図 1 にこの ECC 回路を内蔵したメモリ回路の概略図を示す。ECC 内蔵メモリのアクセスにおいて、書き込み時はデータビット用メモリに入力データが書き込まれる。それと同時に ECC 回路では入力データを元にコード情報 (チェックビット) が生成され、これがチェックビット用メモリに書き込まれる。読み出し時はデータビットとチェックビットがそれぞれのメモリから読み出され、ECC 回路内においてシンドロームと呼ばれるエラー検出データが生成される。ここ

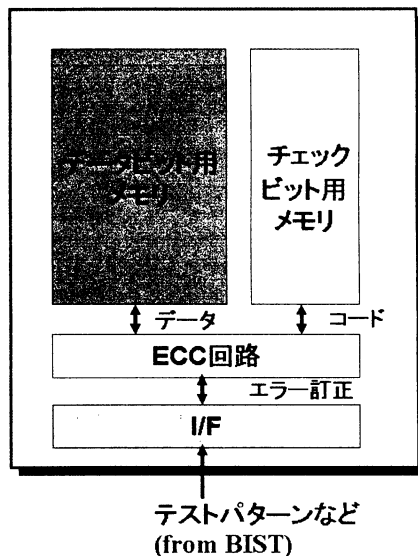


図 1 ECC内蔵メモリの概略図

で読み出されたデータにエラーがあった場合、シンドローム信号はデータのどのビットにエラーがあるかを検知し、エラービットを訂正する。

このように ECC はソフトエラーに対して有効だが、ECC を搭載する際にはコスト増加が問題となる。チェックビット用メモリと ECC 回路を追加するとチップ面積が増加するだけでなく、それらのテストコストも追加される。さらにこれらのテスト時間の増加のみでなく、チェックビット用メモリに対しては固有のメモリ BIST 制御も必要になる。

過去にこれらのテスト時間の増加を回避するため、ECC デザイン技術に関する研究が報告されている [1,2]。彼らは ECC 回路のコード化規則の元となるハミング行列に注目している。データビット用メモリのテスト時に ECC を介してチェックビット用メモリのテストパターンを生成するようなハミング行列を用いることで、これらのメモリを同時にテストすることを可能にしている。ただしそれらの研究ではハミング行列が ECC の 2 ビットエラー検知機能に対応していない。ECC 機能として必要不可欠な単ビットエラー訂正 (SEC: Single Error Correction) 機能に加えて、2 ビットエラー検知 (DED: Double Error Detect) 機能はシステムに対してエラーが発生したことを警告するため非常に重要である。さらに ECC をソフトエラーだけでなくハードエラーを救済するリダンダンシ機能として使用する場合にも有用である [3,4]。

そこで我々は SEC-DED 機能についてもサポートしたハミング行列のデザイン方法を提案する。それにより SEC-DED 機能をもつ ECC 内蔵メモリに標準 BIST を用いた通常のテストを行うだけで、チェックビ

ット用メモリを同時にテストすることが可能になる。この技術を適用すれば ECC を搭載しても新たなテスト時間や固有の BIST 回路が必要とされないため、テストコストの増加を回避することができる。次の第 2 章ではハミング行列のデザイン制約について議論し、続く第 3 章では新しいハミング行列を適用した ECC を内蔵する 512Kb SRAM マクロについて説明する。さらにこの 512Kb SRAM マクロを製造、評価した結果を第 4 章で紹介する。

## 2. ハミング行列の制約

この章では本提案のハミング行列について、SEC-DED 機能に必要な制約と新たに採用する制約について説明する。図 2 にこれらの一覧を示す。

- (0) すでに述べたとおり、SEC-DED コードのサポートは ECC 機能にとって最も重要な特徴の一つである。この制約は奇数重行 (odd-weighted column) としてよく知られている [5]。

$$h_{j1} \oplus h_{j2} \oplus \dots \oplus h_{jm} = 1 \quad (j=1,2,\dots,n) \quad (式 1)$$

ここで、 $h_{jk}$  はハミング行列  $\mathbf{H}$  の (j,k) 因子である。

次にこれから述べる 3 つの制約は、データビット用メモリとチェックビット用メモリの同時テストを実現するために新たに提案したものである。これらの制約を満たしてハミング行列をデザインすることで、通常のメモリ BIST を用いた通常のテストによってチェックビット用メモリがテストされる。

- (1) チェックビット用メモリ中において、どのビットも  $0 \rightarrow 1$ 、 $1 \rightarrow 0$  遷移が行われる。

### SEC-DED 制約

(0): どの列の和も奇数となる (odd-weighted column)

### 本提案の制約

(1): 全てのビットで  $0 \rightarrow 1$ 、 $1 \rightarrow 0$  遷移がテストされる

(2): どのビットの組み合わせでも 00/01/10/11 パターンがテストされる

(3): 全てのアドレスで 111...1 and 000...00 パターンがテストされる  
もし  $H=(\text{奇数}, \text{偶数})$  または  $(\text{偶数}, \text{奇数})$  のときは  
111...11  $\rightarrow$  011...11 パターンとする

図 2 ハミング行列デザインの制約

この制約はメモリの書き込み/読み出し動作のテストの基本として不可欠である。

次に紹介する制約は、同時メモリテストを行うにあたり不良ビットのバス抜けを防ぐために採用された。ECCをハードエラー救済にも適用する場合、スクリーニングテスト時は訂正可能なシングルビットのハードエラーはバスさせる一方で、DED機能により2ビット以上のハードエラーをスクリーニングする必要がある。この場合、図3の例に示すような2ビット不良のすり抜けに注意しなければならない。ここで1ワード中に0と1の2つの固定エラーがあるケースを考える。この場合、a) から d) までのどのテストパターンでもこのワードは2ビットの固定不良があるにもかかわらず1ビットの不良しか見えない。そのため、このワードはSEC機能により1ビットエラーの救済可能なワードとして扱われてしまう。このような2ビット不良のすり抜けを避けるため下記の制約を設ける。

- (2) データビットとチェックビット内のどの2ビットペアにおいてもテストパターン中に00/01/10/11 データパターンがテストされる。

次に説明する制約は、チェックビット用メモリに入力されるデータパターンに関するものである。データビット用メモリに、全てのビットが0のパターンを書き込む場合、ECC回路の特性上チェックビット用メモリにも全てのビットに0が書き込まれる。その一方で、データビット用メモリに書き込むパターンが全て1のときは、ECC回路によりコード化されたチェックビットのパターンは通常全てが1にはなることはない。そのため、データビットのパターンが全て1のときにチェックビットのパターンも全て1となるように以下の制約を設ける。

- (3) データビットとチェックビットのメモリセルのパターンが全て0と全て1の状態にセットされる。

$$h_{1k} \oplus h_{2k} \oplus \dots \oplus h_{nk} = 1 \quad (k=1,2,\dots,m)$$

(式2)

ここでこの制約を奇数行列(odd-weighted row)と呼ぶ。もしハミング行列のサイズ  $H_d(n, m)$  が(奇数, 偶数)もしくは(偶数, 奇数)であるときは、制約(3)とSEC-DED制約(0)は互いに矛盾する。この場合、データビットに全て1が書かれるときだけテストパターンの最上位ビットを1から0(つまり0111...1)に代える。それにより式(2)は次のように修正される。

$$h_{1k} \oplus h_{2k} \oplus \dots \oplus h_{n-1k} = 1 \quad (k=1,2,\dots,m)$$

(式2.1)

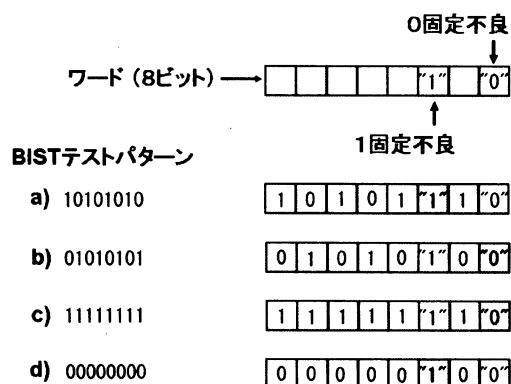


図3 2ビットエラーすり抜け問題の一例

この制約(3)は必ずしも必要不可欠ではないが、通常チェックビットのデータパターンは簡単に外部から確認することができないため、チェックビットのデータを全て1にできるようにしておく事は非常に有用である。

以上の制約について、ハミング行列とテストパターンの具体例を用いて紹介する(図4)。簡単のため、ここではハミング行列のサイズを(8+5)x5とする。ただし実際にはこれよりずっと広いデータ幅が用いられる。テストパターンについては8ビットのデータビットに対し、5ビットのチェックビットを使用する。また、第4章で詳しく述べるが、チェックビット用メモリによる面積増加を抑制するため、コード共有化技術を想定している。そこで内部のデータビット幅が8ビットであるのに対し、外部のアクセスビット幅を2ビットとしている。このためテストパターンが2ビットずつ変化しているように見えるが、実際は外部のテストパターンとして通常のマーチパターンが使用されている。以下では各制約がどのように機能しているかを説明する。

- (1) 提案例では全てのチェックビットが0→1と0→1の遷移を経験しているが、従来例ではチェックビットの最下位がこれを満たしていない。
- (2) 暗色で示される2ビットのテスト列に注目する。提案例では00/01/10/11の全パターンが含まれているのに対し、従来例では01/10のパターンが欠けている。提案例であれば制約(2)により図3で紹介した2ビットのハードエラーは正しくスクリーニングされる。
- (3) 従来例ではテストパターン中にコードデータが全て1になることはないが、提案例ではテストパターンが011...1のときにチェックビットが

(i) 従来のハミング行列

$$H_c(8 \times 5) \quad I_m(5 \times 5)$$

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

Odd-weighted column

データビット: w	チェックビット: c
0000000000	00000000
0000000011	11110000
0000011111	10100000
0001111111	01100000
0111111111	10100000
1111111100	01010000
1111100000	00000000
1100000000	11000000
0000000000	00000000
0101010101	11110011
0101010110	00001111
0101010101	01000111
0110101010	10000111
1010101010	01000111
1010100011	10111111
1010010101	11110011
1001010101	00110111
0101010101	11110011

(2) 00/11のみテスト

(ii) 本提案のハミング行列

$$H_c(8 \times 5) \quad I_m(5 \times 5)$$

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

Odd weight-  
ed row

Odd-weighted column

データビット: w	チェックビット: c
0000000000	00000000
0000000011	01111111
0000011111	10010000
0001111111	01100100
0111111111	10100000
0111111100	01010000
0111100000	11000000
0100000000	00000000
0000000000	11000000
0101010101	11110011
0101010110	00001111
0101010101	01000111
0110101010	10000111
1010101010	01000111
1010100011	10111111
1010010101	11110011
1001010101	00110111
0101010101	11110011

(2) 全00/10/11/11テスト

図4 ハミング行列(SEC-DED方式)とテストパターン例

全て1となる。(3)の補足制約により、ここではテストパターンの最上位ビットを0としている。これらの制約を用いることで、従来のテストパターンとテストシーケンスによるメモリテストによってチェックビット用メモリのテストも同時に、適切に行われる。

### 3. SRAM 設計

本提案のECCを搭載するにあたり、チップサイズと性能劣化を抑制するために、このほかにいくつかの技術を盛り込んだ。ここでは32ビット幅のI/Oを持つSRAMマクロについて考える。

まずデザイン課題の一つが面積抑制である。今回チェックビット用メモリによる面積増加を抑制するため、コード共有化技術を採用した。この技術について以下に説明する。SEC-DEDのECCでは、32ビットのデータに対して7ビットのチェックビットが必要となるが、128ビットのデータに対しては9ビットのチェックビットしか必要としない。このとき、それぞれのチェックビットによる面積ペナルティは2.2%と7%である。このようにECCでは処理するデータ幅が大きいほど面積ペナルティは小さくなる。そこで今回は外部の32ビットデータ幅に対して128ビットの内部データ幅を採用し、データはインターフェースのマルチプレクサを介してアクセスする方式を採用した。これによりビット幅の小さいメモリにECCを採用する場合でも面積増加を最小限に抑えることができる。

もう一つのデザイン課題はスピード性能である。こ

ではスピード性能への対策として、チェックビット用メモリに2ポートメモリを採用している。図5はデータビット用メモリとチェックビット用メモリのアクセス時のタイミングチャートである。書き込み動作[A]では、新しいデータの書き込み前に128ビットのデータビットと9ビットのチェックビットが全て読み出される。もし読み出されたデータに単ビットエラーが検知された場合、このエラーは新しいコードビットを生成する前に訂正されなければならない。この動作はコード共有技術とビットごとのライトマスク機能を搭載するためには必要不可欠である。そして新しい32ビットのデータは同じサイクルでデータビット用メモリに格納される。一方で、新しく書き込むデータを加えた128ビットデータから新たなチェックビットが生成される。サイクルタイムへの影響を最小にするため、コード生成とチェックビット用メモリへの書き込みは次のサイクル[B]と[C]でそれぞれ行われる。もし別の書き込みがサイクル[C]で発生した場合、チェックビット用メモリは異なるアドレスに同時に読み出しと書き込みが行われることになる。このことから、2ポートSRAMをチェックビット用メモリとして採用した。これにより、1ポートSRAMを採用した場合にはこの作業に2サイクルが必要となるのに対し、2ポートメモリでは1サイクルで処理することが可能になる。

さらにもう一つのデザイン課題はECC回路のスピード性能テストである。通常はECCでエラーを訂正する動作を含むサイクルがクリティカルパスとなるため、このことに十分注意してテストを行う必要がある。そ

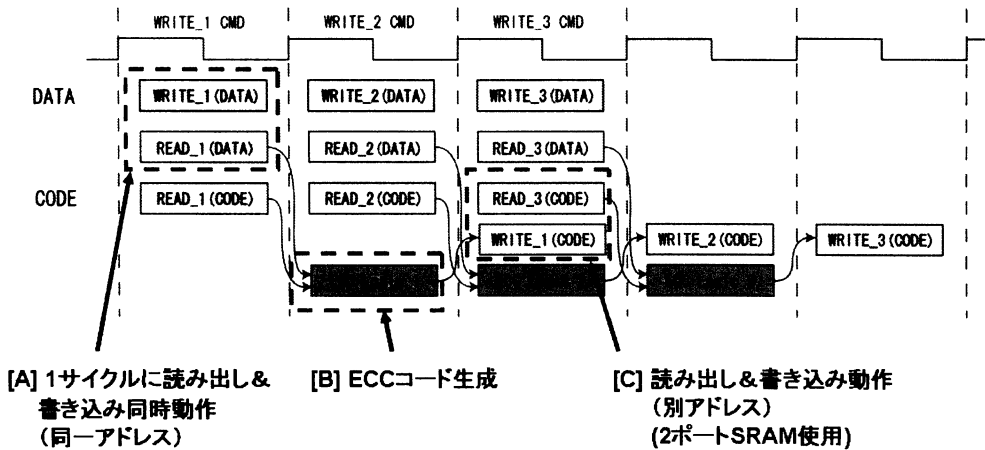


図5 本提案SRAMにおける書き込み動作時のタイミングチャート

のため従来のスキャンテストは ECC 回路のスピードスクリーニングテストとして完全ではない。そこで新しく擬似ソフトエラー生成回路を採用した。このテスト回路はすべてのビットに対してエラーを挿入することができ、ECC のクリティカルパスを活性化することが可能である。したがってこの回路を使用すれば全てのエラー訂正パスにアクセスしてスピードチェックを行うことができる。

#### 4. 性能と評価結果

今回提案した ECC 技術を検証するため、32ビット I/O の 512Kb ECC 内蔵 SRAM マクロを設計した。これは 90nmCMOS プロセスで製造された。図6はそのチップ写真である。512Kb (4K ワード、128ビット) の1ポート SRAMをデータビット用メモリとして採用し、また 36Kb(4K ワード、9ビット)の2ポート SRAMをチェックビットメモリとして採用した。従来の ECC

SRAM マクロ (32I/O、512Kb、ライトマスク有り) と今回提案したマクロの特徴を表1に示した。

データビットのテストと一緒にチェックビット用メモリのスクリーニングを行う検証は、十分な低電圧条件でハードエラーを再現したチップを使用して行われた。ここでは ECC 回路内でエラーの発生を報告する SEC 信号と DED 信号の機能を使用した。検証の結果、チェックビット内に単ビットのエラーが存在するときは ECC 回路が SEC 信号を出力し、メモリ BISTは“pass”を報告した。このことから、チェックビット内の単ビットエラーは ECC により正しく訂正されたことがわかる。また、2ビット以上のエラーが存在するときは ECC 回路が DED 信号を出力し、メモリ BISTは“fail”を報告した。このことから ECC で訂正できないエラーが存在したときは DED 信号により速やかにシステムに対して警告が発せられ、さらにチェックビットに含まれるエラーが BIST のテスト結果に反映されている

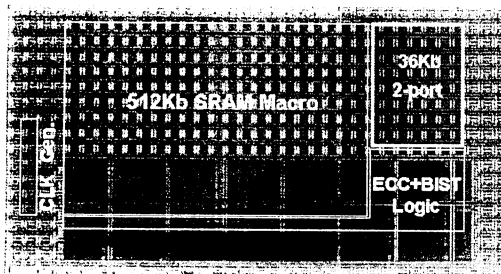


図6 チップ写真

x32 I/O, 512kb ECC SRAM

	Conventional	Proposed
Frequency (@typ.)	200MHz	333MHz
BIST type	Custom	Standard
Test time	BIST + Scan	BIST only
ECC test	Scan test	BIST (at a time)
Size penalty	+29% +redundancy +Custom BIST	+20%

表1 性能比較表

ことがわかる。また、ECC 回路自体の検証として、SEC 信号の検証は擬似ソフトエラー生成回路を用いて行われ、DED 信号についても同様に擬似ソフトエラー生成回路でさまざまな2ビットのエラーパターン適用することで検証された。

図7はSRAMマクロについて、ECCを有効または無効にしてスピード性能を評価した結果である。興味深いことに、ECC機能を有効にした方がSRAMを高速動作させることが可能であった。これは高速動作時においてECC回路がスピードの遅いセルを救済するからであると考えられる。図8(1)は512KbのECC内蔵SRAMチップを十分低電圧動作させたとき、ECCで訂正可能であった最大ビット数の分布を示している。これによりECCのエラー訂正能力を検証した。その結果平均で約200ビットのエラーを訂正できることが判明した。図8(2)はその検証を示すフェイルビットマップである。この結果から、ECCはソフトエラーだけでなく、ハードエラーの救済にも非常に効果的であることがわかる。

### 5. 結論

ECC内蔵メモリマクロの新しいDFT技術が提案された。SEC-DED機能を持つECCのハミング行列に本提案の制約を追加することで、標準メモリBISTと従来のテスト制御でチェックビット用メモリを同時にテストすることが可能になった。この技術によりテスト時間の増加と固有のメモリBISTの追加が不要となった。またECCを搭載することによる面積増加と性能劣化を抑制するための回路技術を搭載した512KbSRAMマクロを設計した。これらの技術は90nmCMOSプロセス

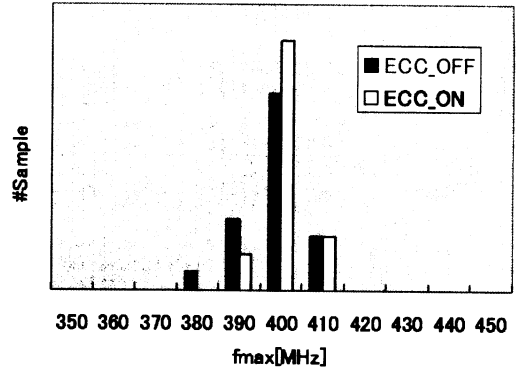
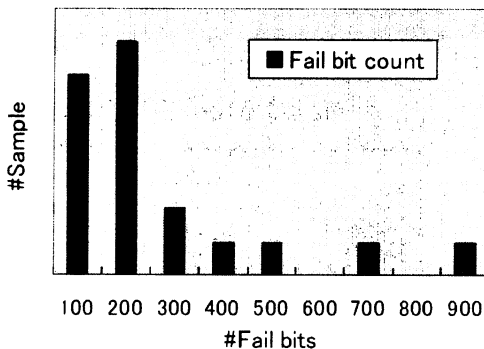


図7 スピード性能比較

で製造された512KbSRAMマクロによって検証され、これらの効果が有効であることが確認された。

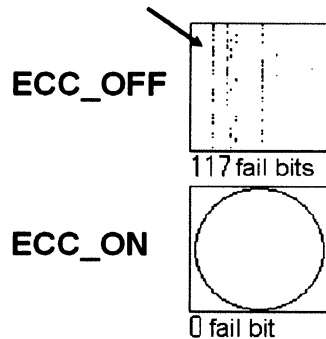
### 文 献

- [1] P. Ramanathan, K.K. Saluja, M. Franklin, "Zero cost testing of check-bits in RAMs with on-chip ECC," VLSI Test Symposium, pages 292-297, Apr.1992.
- [2] P. Ramanathan, K.K. Saluja, M. Franklin, "Testing check-bits at no cost in RAMs with on-chip ECC," Computers and Digital Techniques, IEE Proceedings E, Vol.140 Issue 6, Nov.1993.
- [3] E.Q. Garcia, S.M. Paranjape, "Soft Error Correction Technique And System For Odd Weighted Row Error Correction Codes," IBM, U. S. patent No. 5644583
- [4] G. Nagano, M. Takahashi, "Error-Correcting System," Fujitsu, U.S. patent 4394763
- [5] L. Penzo et al., "VLSI design of systematic odd-weight-column byte error detecting SEC-DED codes," in Proc. 8th Int. Conf. on VLSI Design, Jan.1995.



(1) ECC救済ビット数

Hard errors near VDDmin condition



(2) 本提案SRAMのフェイルビットマップ

図8 ECCによるエラー訂正