Codes for Partially Stuck-at Memory Cells

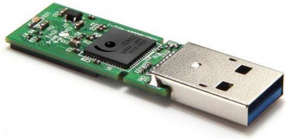
Antonia Wachter-Zeh and Eitan Yaakobi

Computer Science Department
Technion—Israel Institute of Technology

April 19, 2015

Coding Seminar at Technion

Motivation: Non-volatile memories

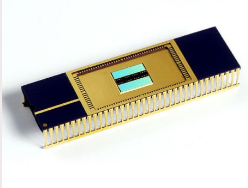


Multi-level flash memories

- electronic charge represents multiple cells
- if charge is trapped, level can only be increased
- avoid erasing of blocks by only increasing the levels in a new write

Phase change memories (PCMs)

- multiple distinct physical states (one amorphous & some partially crystalline)
- cells might reach only crystalline states



Outline

1. Definitions & (Partially) Stuck-At Cells
2. Bounds on the Redundancy
3. Our Constructions
   - Construction for $u < q$
   - Construction Based on q-ary Codes
   - Construction Based on Binary Codes
4. Codes for Unreachable Levels
5. Capacity Considerations
6. Overview & Conclusion

Outline

1. Definitions & (Partially) Stuck-At Cells
2. Bounds on the Redundancy
3. Our Constructions
   - Construction for $u < q$
   - Construction Based on q-ary Codes
   - Construction Based on Binary Codes
4. Codes for Unreachable Levels
5. Capacity Considerations
6. Overview & Conclusion

(Partially) Stuck-at Cells

Stuck-at Cells (memory with defects) [Kuznetsov-Tsybakov, 1974]

- binary cells: cell can be stuck-at level 0 or 1
- q-ary cells: cell can be stuck-at any level $s \in [q]$
- a stuck-at cell cannot change its level

Partially Stuck-at Cells

- q-ary cells: cell can be partially stuck-at any level $s \in [q]$
- a partially stuck-at cell can only store levels at least s

Examples:

- trapped charge in flash memories: only high levels
- in PCMs, a cell cannot reach the amorphous state (but all others)
- degraded reliability: cells cannot/shouldn't represent high levels

(Partially) Stuck-at Cells

Stuck-at Cells (memory with defects) [Kuznetsov-Tsybakov, 1974]

- binary cells: cell can be stuck-at level 0 or 1
- q-ary cells: cell can be stuck-at any level $s \in [q]$
- a stuck-at cell cannot change its level

Partially Stuck-at Cells

- q-ary cells: cell can be partially stuck-at any level $s \in [q]$
- a partially stuck-at cell can only store levels at least s

Examples:

- trapped charge in flash memories: only high levels
- in PCMs, a cell cannot reach the amorphous state (but all others)
- degraded reliability: cells cannot/shouldn't represent high levels

Definitions

$(u, \mathbf{s})$-stuck-at-masking code ($(u, \mathbf{s})$-SMC) &
$(u, \mathbf{s})$-partially-stuck-at-masking code ($(u, \mathbf{s})$-PSMC)

A $\begin{cases} (u, \mathbf{s})\text{-SMC} \\ (u, \mathbf{s})\text{-PSMC} \end{cases}$ is a coding scheme with encoder $\mathcal{E}$ & decoder $\mathcal{D}$:

- $\mathcal{E}$: input: message m, locations & values (stored in $\mathbf{s} = (s_0, s_1, \ldots, s_{u-1})$) of u (partially) stuck-at cells

  output: vector $\mathbf{y}^{(m)}$ with $\begin{cases} y_i^{(m)} = s_i \\ y_i^{(m)} \geq s_i \end{cases}$ at stuck positions
- $\mathcal{D}$: input: $\mathbf{y}^{(m)}$

  output: message

short-hand notations:

- (u, s)-PSMC for $s_i = s$
- u-PSMC for $s_i = 1$ (this talk)

Construction of $(u, \mathbf{s})$-SMCs

Theorem (Masking Stuck-At Cells, [Heegard, 1983])

Let $\mathcal{C}$ be an $[n, k, d]_q$ code with minimum distance $d \geq u + 1$. Then, there exists a $(u, \mathbf{s})$-SMC with redundancy $r = n - k$.

Example: $[7, 4, 3]_2$ Hamming code: $u = 2$ and $r = 3$ (store 4 bits)

- message $\mathbf{m} = (0, 1, 1, 0)$
- stuck positions: 1, 5
- define $\mathbf{w} = (0, 0, 0, \underbrace{0, 1, 1, 0}_{=\mathbf{m}})$

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

- find $\mathbf{z} \in \{0, 1\}^3$ s.t. $\mathbf{y} = \mathbf{w} + \mathbf{z} \cdot \mathbf{H}$ masks stuck-at cells $\implies \mathbf{z} = (1, 1, 0)$
- $\mathbf{y} = \mathbf{w} + \mathbf{z} \cdot \mathbf{H} = (0, 0, 0, 0, 1, 1, 0) + (1, 1, 0, 1, 1, 0, 0)$ $= (1, 1, 0, 1, 0, 1, 0)$
- reconstruction possible since $(y_0, y_1, y_2) = \mathbf{z}$

Construction of $(u, \mathbf{s})$-SMCs

Theorem (Masking Stuck-At Cells, [Heegard, 1983])

Let $\mathcal{C}$ be an $[n, k, d]_q$ code with minimum distance $d \geq u + 1$. Then, there exists a $(u, \mathbf{s})$-SMC with redundancy $r = n - k$.

Example: $[7, 4, 3]_2$ Hamming code: $u = 2$ and $r = 3$ (store 4 bits)

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

- message $\mathbf{m} = (0, 1, 1, 0)$
- stuck positions: 1, 5
- define $\mathbf{w} = (0, 0, 0, \underbrace{0, 1, 1, 0}_{=\mathbf{m}})$
- find $\mathbf{z} \in \{0, 1\}^3$ s.t. $\mathbf{y} = \mathbf{w} + \mathbf{z} \cdot \mathbf{H}$ masks stuck-at cells
  $\Longrightarrow \mathbf{z} = (1, 1, 0)$
- $\mathbf{y} = \mathbf{w} + \mathbf{z} \cdot \mathbf{H} = (0, 0, 0, 0, 1, 1, 0) + (1, 1, 0, 1, 1, 0, 0)$
  $= (1, 1, 0, 1, 0, 1, 0)$
- reconstruction possible since $(y_0, y_1, y_2) = \mathbf{z}$

Outline

Bounds & Trivial Constructions of PSMCs

Theorem (Bounds on the Redundancy)

For any u partially stuck-at cells and levels $\mathbf{s} = (s_0, s_1, \ldots, s_{u-1})$, the minimum redundancy $r_q(n, u, \mathbf{s})$ to mask these cells satisfies

$$u - \log_q \left(\prod_{i=0}^{u-1} (q - s_i) \right) \leq r_q(n, u, \mathbf{s}) \leq \min \left\{ n \left(1 - \log_q \left(q - \max_i \{s_i\} \right) \right), \ \rho_q(n, u+1) \right\}.$$

- lower bound: stuck cells can represent $q - s_i$ levels, non-stuck cells q levels $\Rightarrow M \leq q^{n-u} \prod_{i=0}^{u-1} (q - s_i)$
- upper bound: use only levels $\max_i\{s_i\}, \ldots, q-1 \Rightarrow M = (q - \max_i\{s_i\})^n$
- upper bound: use $(u, \mathbf{s})$-SMC

Theorem (Improved Lower Bound for $s_i = s$)

For any (u, s)-PSMC, we have

$$r_q(n, u, s) \geq \log_q(u+1) - \log_q \left(1 + u(1 - s/q)^n \right).$$

Bounds & Trivial Constructions of PSMCs

Theorem (Bounds on the Redundancy)

For any u *partially stuck-at cells and levels* $\mathbf{s} = (s_0, s_1, \ldots, s_{u-1})$, *the minimum redundancy* $r_q(n, u, \mathbf{s})$ *to mask these cells satisfies*

$$u - \log_q \left(\prod_{i=0}^{u-1} (q - s_i) \right) \leq r_q(n, u, \mathbf{s}) \leq \min \left\{ n \left(1 - \log_q \left(q - \max_i \{s_i\} \right) \right),\ \rho_q(n, u+1) \right\}.$$

- lower bound: stuck cells can represent $q - s_i$ levels, non-stuck cells q levels $\Rightarrow M \leq q^{n-u} \prod_{i=0}^{u-1} (q - s_i)$
- upper bound: use only levels $\max_i\{s_i\}, \ldots, q-1 \Rightarrow M = (q - \max_i\{s_i\})^n$
- upper bound: use $(u, \mathbf{s})$-SMC

Theorem (Improved Lower Bound for $s_i = s$)

For any (u, s)*-PSMC, we have*

$$r_q(n, u, s) \geq \log_q(u+1) - \log_q \left(1 + u(1 - s/q)^n \right).$$

Outline

Construction for $u < q$

Theorem (Construction I for $u < q$)

If $u < q$ and $u \leq n$, then for all n, there exists an $(n, M = q^{n-1})_q$ u-PSMC with redundancy of one symbol.

Example: $u = 2$, $q = 3$, $n = 5$, $s_i = 1$, $\forall i$

- message $\mathbf{m} = (2, 0, 1, 0)$
- stuck positions: 1, 2
- define $\mathbf{w} = (0, 2, 0, 1, 0)$
- find $z \in [q]$ s.t. $\mathbf{y} = \mathbf{w} + z \cdot (1, 1, \dots, 1)$ masks part. stuck-at cells:
  - $z = 0$: $\mathbf{y} = \mathbf{w} = (0, 2, 0, 1, 0)$ ↯
  - $z = 1$: $\mathbf{y} = \mathbf{w} + (1, 1, \dots, 1) = (1, 0, 1, 2, 1)$ ↯
  - $z = 2$: $\mathbf{y} = \mathbf{w} + (2, 2, \dots, 2) = (2, 1, 2, 0, 2)$ ✓
- reconstruction: $y_0 = z$ and thus, $\mathbf{y} - y_0 \cdot (1, 1, \dots, 1)$ gives $\mathbf{w}$ and $\mathbf{m}$

$\Longrightarrow$ redundancy $r = 1$

$\Longrightarrow$ lower bound: $\max\{0.738, 0.787\} = 0.787$

$\Longrightarrow$ upper bound: $\min\{5 \cdot (1 - \log_3 2),\ \rho_3(5, 3) = 3\} = 1.845$

Construction for $u < q$

Theorem (Construction I for $u < q$)

If $u < q$ and $u \leq n$, then for all n, there exists an $(n, M = q^{n-1})_q$ u-PSMC with redundancy of one symbol.

Example: $u = 2,\ q = 3,\ n = 5,\ s_i = 1,\ \forall i$

- message $\mathbf{m} = (2, 0, 1, 0)$
- stuck positions: 1, 2
- define $\mathbf{w} = (0, 2, 0, 1, 0)$
- find $z \in [q]$ s.t. $\mathbf{y} = \mathbf{w} + z \cdot (1, 1, \dots, 1)$ masks part. stuck-at cells:
  - $z = 0$: $\mathbf{y} = \mathbf{w} = (0, 2, 0, 1, 0)$ ϟ
  - $z = 1$: $\mathbf{y} = \mathbf{w} + (1, 1, \dots, 1) = (1, 0, 1, 2, 1)$ ϟ
  - $z = 2$: $\mathbf{y} = \mathbf{w} + (2, 2, \dots, 2) = (2, 1, 2, 0, 2)$ ✓
- reconstruction: $y_0 = z$ and thus, $\mathbf{y} - y_0 \cdot (1, 1, \dots, 1)$ gives $\mathbf{w}$ and $\mathbf{m}$

$\Longrightarrow$ redundancy $r = 1$

$\Longrightarrow$ lower bound: $\max\{0.738, 0.787\} = 0.787$

$\Longrightarrow$ upper bound: $\min\{5 \cdot (1 - \log_3 2),\ \rho_3(5, 3) = 3\} = 1.845$

Construction for $u < q$—Improvement

Theorem (Construction IB)

For any $u < q$, and $u \leq n$, there exists a u-PSMC over $[q]$ of length n and redundancy

$$r = 1 - \log_q \left\lfloor \frac{q}{u+1} \right\rfloor .$$

Example: $u = 2$ and $q = 6$, use same principle as before

- BUT: search $z \in \{0, 1, 2\}$ (instead of $z \in [q]$)
- store additional information in the redundancy cell by representing 0 by 0 and 3; 1 by 1 and 4; 2 by 2 and 5
- required redundancy: $1 - \log_6(2) \approx 0.613$ q-ary symbols

Theorem (Optimality of Construction IB)

If $(u+1)$ divides q, the u-PSMC from Construction IB is asymptotically optimal in terms of the redundancy.

Construction for $u < q$—Improvement

Theorem (Construction IB)

For any $u < q$, and $u \leq n$, there exists a u-PSMC over $[q]$ of length n and redundancy

$$r = 1 - \log_q \left\lfloor \frac{q}{u+1} \right\rfloor .$$

Example: $u = 2$ and $q = 6$, use same principle as before

- BUT: search $z \in \{0, 1, 2\}$ (instead of $z \in [q]$)
- store additional information in the redundancy cell by representing 0 by 0 and 3; 1 by 1 and 4; 2 by 2 and 5
- required redundancy: $1 - \log_6(2) \approx 0.613$ q-ary symbols

Theorem (Optimality of Construction IB)

If $(u + 1)$ divides q, the u-PSMC from Construction IB is asymptotically optimal in terms of the redundancy.

Generalization to Arbitrary Levels

Theorem (Generalized Construction I)

Let u *and* n *be positive integers and assume that* $u \leq n$ *cells are partially stuck-at levels* $\mathbf{s} = (s_0, s_1, \ldots, s_{u-1}) \in \{1, \ldots, q-1\}^u$*. If* $\sum_{i=0}^{u-1} s_i < q$*, then there exists a* $(u, \mathbf{s})$*-PSMC over* $[q]$ *with redundancy*

$$r = 1 - \log_q \left\lfloor \frac{q}{\sum_{i=0}^{u-1} s_i + 1} \right\rfloor .$$

However, for general $\mathbf{s}$, it is not clear if this construction is asymptotically optimal...

Construction Based on q-ary Codes

Theorem (Construction II)

Let $d-1 \le u \le q+d-3$, and let $\mathbf{H}$ be the $(n-k)\times n$ parity-check matrix of an $[n,k,d]_q$ code. Then, there exists a u-PSMC over $\mathbb{F}_q$ of length n and redundancy $r = n-k$.

Corollary

If q is a prime power, then for all $n \ge q$, there exists a u-PSMC with redundancy $r = \rho_q\left(n, u-q+3\right)$.

Example: $u=5$, $q=5$ and $n=30$.

- our construction: $r = \rho_q\left(n, u-q+3\right) = \rho_5\left(30,3\right) = 3$
- lower bound: $\max\{0.693, 1.11\} = 1.11$
- upper bound: $\min\{4.16, \rho_5\left(30,6\right) = 8\} = 4.16$

Construction Based on Binary u-SMCs

Theorem (Construction III Using Binary Codes)

Let n, $q \geq 4$ *and* u *be positive integers and let* $\widetilde{u} = \lfloor 2u/q \rfloor$*.*
Assume that $\widetilde{\mathcal{C}}$ *is an* $[n, k, d \geq \widetilde{u} + 1]_2$ *binary* $(\widetilde{u}, \widetilde{\mathbf{s}})$*-SMC with encoder* $\widetilde{\mathcal{E}}$ *and decoder* $\widetilde{\mathcal{D}}$ *given by Theorem 1.*
Then, there exists a q*-ary* u*-PSMC of length* $(n+1)$ *with redundancy*

$$r = (n - k - 1)\log_q\left(\frac{q}{\lfloor q/2 \rfloor}\right) + 2.$$

Principle:

- find $z \in [q]$ such that the number of zeros and $(q-1)$s is minimized in the vector $(\mathbf{w}^{(z)})_U = ((0, \ldots, 0, \mathbf{m}, 0) + z \cdot (1, 1, \ldots, 1))_U$, and denote this value by $\widetilde{u}$.
- use a binary $(\widetilde{u}, \widetilde{\mathbf{s}})$-SMC to mask these $\widetilde{u}$ stuck-at cells.

Construction Based on Binary u-SMCs—Example (i)

Example: $n = 15$, $q = 4$, $u = 5$ and $U = \{1, 4, 8, 12, 15\}$.
$\Longrightarrow \widetilde{u} = \lfloor 2u/q \rfloor = 2$, use $[15, 11, 3]_2$ code $\widetilde{\mathcal{C}}$ as binary 2-SMC with

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

- messages $\mathbf{m} = (0, 3, 2, 1, 2, 2, 3, 1, 3, 2, 2) \in [3]^{11}$, $\mathbf{m}' = (1, 0, 1) \in [1]^3$
- define $\mathbf{w} = (0, 0, 0, 0, \underbrace{0, 3, 2, 1, 2, 2, 3, 1, 3, 2, 2}_{\mathbf{m}}, 0)$
- find $z \in [q]$ s.t. the number of 0s/3s is minimal in $(\mathbf{w}^{(z)})_U = (\mathbf{w} + z \cdot (1, 1, \dots, 1))_U$:
  $\Longrightarrow z = 1$ and $\mathbf{w}^{(z)} = (1, 1, 1, 1, 1, 0, 3, 2, 3, 3, 0, 2, 0, 3, 3, 1)$
- mask the 2 usual stuck cells in $\mathbf{v} = (0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0)$ with binary 2-SMC:
  $\widetilde{\mathbf{c}} = ((1, 0, 0, 0) \cdot \mathbf{H}, 0) = (1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0)$
- encode additional message: $\widetilde{\mathbf{m}} = (2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$
- store $\mathbf{y} = \mathbf{w}^{(z)} + \widetilde{\mathbf{c}} + \widetilde{\mathbf{m}} = (0, 1, 3, 1, 1, 0, 3, 2, 3, 0, 1, 3, 1, 0, 0, 1)$

Example: $n = 15$, $q = 4$, $u = 5$ and $U = \{1, 4, 8, 12, 15\}$.
$\Longrightarrow \widetilde{u} = \lfloor 2u/q \rfloor = 2$, use $[15, 11, 3]_2$ code $\widetilde{\mathcal{C}}$ as binary 2-SMC with

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

- messages $\mathbf{m} = (0, 3, 2, 1, 2, 2, 3, 1, 3, 2, 2) \in [3]^{11}$, $\mathbf{m}' = (1, 0, 1) \in [1]^3$
- define $\mathbf{w} = (0, 0, 0, 0, \underbrace{0, 3, 2, 1, 2, 2, 3, 1, 3, 2, 2}_{\mathbf{m}}, 0)$
- find $z \in [q]$ s.t. the number of 0s/3s is minimal in $(\mathbf{w}^{(z)})_U = (\mathbf{w} + z \cdot (1, 1, \dots, 1))_U$:
  $\Longrightarrow z = 1$ and $\mathbf{w}^{(z)} = (1, 1, 1, 1, 1, 0, 3, 2, 3, 3, 0, 2, 0, 3, 3, 1)$
- mask the 2 usual stuck cells in $\mathbf{v} = (0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0)$ with binary 2-SMC:
  $\widetilde{\mathbf{c}} = ((1, 0, 0, 0) \cdot \mathbf{H}, 0) = (1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0)$
- encode additional message: $\widetilde{\mathbf{m}} = (2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$
- store $\mathbf{y} = \mathbf{w}^{(z)} + \widetilde{\mathbf{c}} + \widetilde{\mathbf{m}} = (0, 1, 3, 1, 1, 0, 3, 2, 3, 0, 1, 3, 1, 0, 0, 1)$

Decoding process:

given $\mathbf{y} = (0, 1, 3, 1, 1, 0, 3, 2, 3, 0, 1, 3, 1, 0, 0, 1)$

- recover z: $y_3 - y_{15} = 0 \Longrightarrow \widehat{z} = 1$
- $\widehat{\mathbf{y}} = \mathbf{y} - \widehat{z} \cdot (1, 1, \ldots, 1) = (3, 0, 2, 0, 0, 3, 2, 1, 2, 3, 0, 2, 0, 3, 3, 0)$
- $\widehat{\mathbf{m'}} = (\lfloor \widehat{y}_0/2 \rfloor, \ldots, \lfloor \widehat{y}_{n-k-2}/2 \rfloor) = (\lfloor 3/2 \rfloor, 0, \lfloor 2/2 \rfloor) = (1, 0, 1)$
- $\widehat{\mathbf{t}} = (\widehat{y}_0 - 2\widehat{m}'_0, \ldots, \widehat{y}_{n-k-2} - 2, \widehat{m}'_{n-k-2}, \widehat{y}_{n-k-1}) \bmod q =$ $(3 - 2, 0 - 0, 2 - 2, 0) = (1, 0, 0, 0)$
- $\widehat{\mathbf{c'}} = \widehat{\mathbf{t}} \cdot \mathbf{H} = (1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)$
- $\widehat{\mathbf{m}} = (0, 3, 2, 1, 2, 2, 3, 1, 3, 2, 2)$.

The redundancy to mask these five partially stuck-at-1 cells is therefore $r = 3.5$ q-ary cells.

Construction Based on Binary u-SMCs—Example 2

Let $q = 4$, $n = 63$, $k = 57$ and $u = 5$.

We want to construct a u-PSMC of length 64.

- redundancy of Construction III: $r = 4.5$
- redundancy of Construction II: $r = \rho_4\,(64, u - q + 3 = 4) = 6$
- upper bound: $\min\{13.1, \rho_4\,(64, 6) = 13\} = 13$
- lower bound: $\min\{1.04, 1.29\} = 1.04$

$\Longrightarrow$ Improvement compared to known constructions!

Outline

Codes for Unreachable Levels

$(\boldsymbol{u}, \mathbf{s})$-unreachable-masking code

A $(u, \mathbf{s})$-UMC is a coding scheme with encoder $\mathcal{E}$ and decoder $\mathcal{D}$:

- $\mathcal{E}$: input is message m, locations & values of u stuck-at cells, output is $\mathbf{y}^{(m)}$ with $y_i^{(m)} \leq s_i$ at unreliable positions
- $\mathcal{D}$: input is $\mathbf{y}^{(m)}$, output is message

Theorem

Given a $(u, \mathbf{s})$-PSMC with redundancy r and partially stuck levels $\mathbf{s} = (q-1-s_0, q-1-s_1, \dots, q-1-s_{u-1})$. Then, this code can be used as a $(u, \widetilde{\mathbf{s}})$-UMC with redundancy r and levels $\widetilde{\mathbf{s}} = (s_0\ s_1\ \dots\ s_{u-1})$.

This construction outperforms the one of [Gabrys, Sala, Dolecek].

Outline

Capacity

Theorem

The capacity of the q-ary partially stuck-at level s channel is

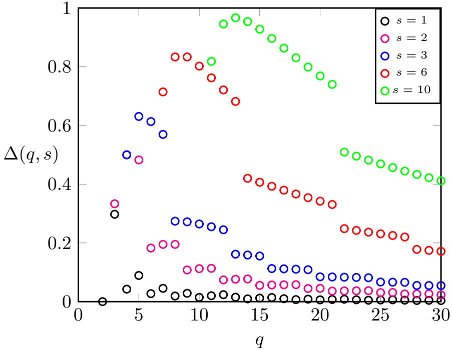
$$C(p,s) = 1 - p \log_q \left(\frac{q}{q-s} \right),$$

where p is the probability that a cell is partially stuck-at level s.

Rate of Construction III: $R(p,s) = 1 - \frac{2sp}{q} \log_q \left(\frac{q}{\lfloor q/(s+1) \rfloor} \right)$

Difference:

$$C(p,s) - R(p,s) = \underbrace{p\left(\frac{2s}{q}\log_q\left(\frac{q}{\lfloor q/(s+1)\rfloor}\right) - \log_q\left(\frac{q}{q-s}\right)\right)}_{\stackrel{\text{def}}{=} \Delta(q,s)}$$



Outline

Overview of our Constructions

Partially stuck-at level 1:

	Upper bound on $r_q(n, u, \mathbf{1})$
$u < q$	$1 - \log_q \left\lfloor \frac{q}{u+1} \right\rfloor$
$u \geq q$	$\rho_q(n, u - q + 3)$
any u, q	$\left(\rho_q\left(n, \lfloor \frac{2u}{q} \rfloor + 1\right) - 1\right) \cdot \log_q \left(\frac{q}{\lfloor q/2 \rfloor}\right) + 2$

Generalized levels $\mathbf{s} = (s_0, s_1, \ldots, s_{u-1})$:

	Upper bound on $r_q(n, u, \mathbf{s})$
$\sum_{i=0}^{u-1} s_i < q$	$r = 1 - \log_q \left\lfloor \frac{q}{\sum_{i=0}^{u-1} s_i + 1} \right\rfloor$
$u \geq \frac{q}{\max_i s_i}$	$\rho_q\left(n, u - \frac{q}{\max_i s_i} + 3\right)$
any u, q	$\left(\rho_q\left(n, \left\lfloor \frac{\sum_{i=1}^{q-1} u_i \cdot \sigma_i}{q} \right\rfloor + 1\right) - 1\right) \cdot \log_q \left(\frac{q}{\lfloor q/Q \rfloor}\right) + 2$, where $Q \geq \max_i\{s_i\} + 1$ is a prime power and $\sigma_i = \min\{q, Q + i - 1\}$, $\forall i \in [q]$.

Conclusion

Our Contribution

- new model of partially stuck-at memory cells
- lower & upper bounds on the redundancy of PSMCs
- three constructions for different ranges of parameters
- capacity analysis

Outlook

- better code constructions
- combination with additional random errors

Thank you! Questions?

Conclusion

Our Contribution

- new model of partially stuck-at memory cells
- lower & upper bounds on the redundancy of PSMCs
- three constructions for different ranges of parameters
- capacity analysis

Outlook

- better code constructions
- combination with additional random errors

Thank you! Questions?