Codes for Partially Stuck-at Memory Cells

Antonia Wachter-Zeh and Eitan Yaakobi

Computer Science Department
Technion—Israel Institute of Technology

May 3, 2015

Workshop on Coding for Emerging Memories and Storage Technologies
Motivation: Non-volatile memories

**Multilevel flash memories**
- electronic charge represents multiple levels
- if charge is trapped, level can only be increased
- degraded reliability: cells cannot/shouldn’t represent high levels

**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
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)**
- 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 $s$

**Known work:**
- masking stuck-at cells first considered by [Kuznetsov-Tsybakov, 1974]
- stuck-at cells & random errors [Heegard, 1983], [Tsybakov, 1984], [Borden-Vinck, 1987]
- recently increased interested in stuck-at cells in PCMs [Lastras-Montano et al., 2010], [Kim-Kumar, 2013]
(Partially) Stuck-at Cells

### Stuck-at Cells (memory with defects)
- **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 $s$

**Known work:**
- Masking stuck-at cells first considered by [Kuznetsov-Tsybakov, 1974]
- Stuck-at cells & random errors [Heegard, 1983], [Tsybakov, 1984], [Borden-Vinck, 1987]
- Recently increased interested in stuck-at cells in PCMs [Lastras-Montano et al., 2010], [Kim-Kumar, 2013]
Definitions

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

\(A \begin{cases} 
(u, s)\text{-SMC} \\
(u, s)\text{-PSMC} 
\end{cases}\)

is a coding scheme with encoder \(E\) & decoder \(D\):

- **\(E\):** input: message \(m\), locations & values (stored in \(s = (s_0, s_1, \ldots, s_{u-1})\)) of \(u\) (partially) stuck-at cells
- **\(D\):** input: \(y^{(m)}\)
- output: message

\[ y^{(m)}_i = s_i \text{ at stuck positions} \]
\[ y^{(m)}_i \geq s_i \]
Construction of \((u, s)\)-SMCs

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

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

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

- message \(m = (0, 1, 1, 0)\)
- stuck positions: 1, 5
- define \(w = (0, 0, 0, 0, 1, 1, 0)\)
- find \(z \in \{0, 1\}^3\) s.t. \(y = w + z \cdot H\) masks stuck-at cells
  \[H = \begin{pmatrix}
  1 & 0 & 0 & 0 & 1 & 1 & 1 \\
  0 & 1 & 0 & 1 & 0 & 1 & 1 \\
  0 & 0 & 1 & 1 & 1 & 0 & 1 \\
\end{pmatrix}\]
  \(y = w + z \cdot H = (0, 0, 0, 0, 1, 1, 0) + (1, 1, 0, 1, 1, 0, 0)
  = (1, 1, 0, 1, 0, 1, 0) \checkmark\)
- reconstruction possible since \((y_0, y_1, y_2) = z\)
Theorem (Masking Stuck-At Cells, [Heegard, 1983])

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

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

- message $m = (0, 1, 1, 0)$
- stuck positions: 1, 5
- define $w = (0, 0, 0, 0, 1, 1, 0)$

Find $z \in \{0, 1\}^3$ s.t. $y = w + z \cdot H$ masks stuck-at cells

$\implies z = (1, 1, 0)$

$y = w + z \cdot H = (0, 0, 0, 0, 1, 1, 0) + (1, 1, 0, 1, 1, 0, 0)$

$= (1, 1, 0, 1, 0, 1, 0, 0) \checkmark$

reconstruction possible since $(y_0, y_1, y_2) = z$
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
Bounds & Trivial Constructions of PSMCs

**Theorem (Bounds on the Redundancy)**

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

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

- **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 \((u, s)\)-SMC

**Theorem (Improved Lower Bound for \( s_i = s \))**

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

\[
 r_q(n, u, s) \geq \log_q(u + 1) - \log_q \left( 1 + u(1 - s/q)^n \right).
\]
Theorem (Bounds on the Redundancy)

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

\[
\begin{align*}
    u - \log_q \left( \prod_{i=0}^{u-1} (q - s_i) \right) &\leq r_q(n, u, s) \\
    &\leq \min \left\{ n \left( 1 - \log_q (q - \max_i s_i) \right), \ \rho_q(n, u + 1) \right\}.
\end{align*}
\]

- **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 $(u, s)$-SMC

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

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

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

**Theorem (Construction 1 for $u < q$)**

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

**Example:** $u = 2$, $q = 3$, $n = 5$

- message $m = (2, 0, 1, 0)$
- stuck positions: 1, 2
- define $w = (0, 2, 0, 1, 0)$
- find $z$ s.t. $y = w + z \cdot (1, 1, \ldots, 1)$ masks partially stuck-at cells:
  - $z = 0$: $y = w = (0, 2, 0, 1, 0)$ ✓
  - $z = 1$: $y = w + (1, 1, \ldots, 1) = (1, 0, 1, 2, 1)$ ✓
  - $z = 2$: $y = w + (2, 2, \ldots, 2) = (2, 1, 2, 0, 2)$ ✓
- reconstruction: $y_0 = z$ and thus, $y - y_0 \cdot (1, 1, \ldots, 1)$ gives $w$ and $m$

$\implies$ redundancy $r = 1$

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

$\implies$ 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 \( (n, M = q^{n-1})_q \) \((u, 1)\)-PSMC with redundancy of one symbol.

Example: \( u = 2, q = 3, n = 5 \)
- message \( m = (2, 0, 1, 0) \)
- stuck positions: 1, 2
- define \( w = (0, 2, 0, 1, 0) \)
- find \( z \) s.t. \( y = w + z \cdot (1, 1, \ldots, 1) \) masks partially stuck-at cells:
  - \( z = 0: y = w = (0, 2, 0, 1, 0) \checkmark \)
  - \( z = 1: y = w + (1, 1, \ldots, 1) = (1, 0, 1, 2, 1) \checkmark \)
  - \( z = 2: y = w + (2, 2, \ldots, 2) = (2, 1, 2, 0, 2) \checkmark \)
- reconstruction: \( y_0 = z \) and thus, \( y - y_0 \cdot (1, 1, \ldots, 1) \) gives \( w \) and \( m \)

\( \implies \) redundancy \( r = 1 \)
\( \implies \) lower bound: \( \max\{0.738, 0.787\} = 0.787 \)
\( \implies \) 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 $(u, 1)$-PSMC over $[q]$ of length $n$ and redundancy

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

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

(recall lower bound: $r_q (n, u, s) \geq \log_q (u + 1) - \log_q \left( 1 + u(1 - s/q)^n \right)$)
### Construction for $u < q$—Improvement

#### Theorem (Construction IB)

For any $u < q$, and $u \leq n$, there exists a $(u, 1)$-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 $(u, 1)$-PSMC from Construction IB is **asymptotically optimal** in terms of the redundancy.

(recall lower bound: $r_q(n, u, s) \geq \log_q(u + 1) - \log_q \left(1 + u(1 - s/q)^n\right)$)
Construction Based on \( q \)-ary Codes

**Theorem (Construction II)**

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

**Corollary**

If \( q \) is a prime power, then for all \( u \leq n \), there exists a \((u, 1)\)-PSMC with redundancy \( r = \max\{1, \rho_q(n, u - q + 3)\} \).

**Example:** \( u = 5 \), \( q = 5 \) and \( n = 30 \).

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

**Theorem (Construction III Using Binary Codes)**

Let $n$, $q \geq 4$ and $u \leq n$ be positive integers and let $\tilde{u} = \lfloor 2u/q \rfloor$. Assume that $\tilde{C}$ is an $[n, k, d \geq \tilde{u} + 1]_2$ binary $(\tilde{u}, \tilde{s})$-SMC. Then, there exists a $(u, 1)$-PSMC over $[q]$ of length $(n + 1)$ and redundancy

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

**Example (simplified):** Let $u = q = 3$, $n = 8$

- partially stuck positions: 0, 5, 6
- message $m = (1, 2, 2)$, define $w = (0, 0, 0, 0, 0, 1, 2, 2)$
- add a codeword of a binary SMC to mask the partially stuck-at cells: $w = (0, 0, 0, 0, 0, 1, 2, 2) \Rightarrow$ use binary SMC with $\tilde{u} = 2$
- Use parity-check matrix of an $[8, 4, 4]_2$ code:

$$y = w + (1, 0, 1, 0) \cdot \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix} = (1, 0, 1, 0, 1, 1, 2, 0) \checkmark$$
Construction Based on Binary $u$-SMCs

Theorem (Construction III Using Binary Codes)

Let $n$, $q \geq 4$ and $u \leq n$ be positive integers and let $\tilde{u} = \lceil 2u/q \rceil$. Assume that $\tilde{C}$ is an $[n, k, d \geq \tilde{u} + 1]_2$ binary $(\tilde{u}, \tilde{s})$-SMC. Then, there exists a $(u, 1)$-PSMC over $[q]$ of length $(n + 1)$ and redundancy

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

Example (simplified): Let $u = q = 3$, $n = 8$

- partially stuck positions: 0, 5, 6
- message $m = (1, 2, 2)$, define $w = (0, 0, 0, 0, 0, 1, 2, 2)$
- add a codeword of a binary SMC to mask the partially stuck-at cells: $w = (0, 0, 0, 0, 0, 1, 2, 2) \Rightarrow$ use binary SMC with $\tilde{u} = 2$
- Use parity-check matrix of an $[8, 4, 4]_2$ code:
  $$y = w + (1, 0, 1, 0) \cdot \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 1
\end{pmatrix} = (1, 0, 1, 0, 1, 1, 2, 0) \checkmark$$
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
Codes for Unreachable Levels

\((u, s)\)-unreachable-masking code

A \((u, s)\)-UMC is a coding scheme with encoder \(E\) and decoder \(D\):
- \(E\): input is message \(m\), locations & values of \(u\) stuck-at cells, output is \(y(m)\) with \(y_i(m) \leq s_i\) at unreliable positions
- \(D\): input is \(y(m)\), output is message

Theorem

Given a \((u, s)\)-PSMC with redundancy \(r\) and partially stuck levels \(s = (q - 1 - s_0, q - 1 - s_1, \ldots, q - 1 - s_{u-1})\).
Then, this code can be used as a \((u, \tilde{s})\)-UMC with redundancy \(r\) and levels \(\tilde{s} = (s_0, s_1, \ldots, s_{u-1})\).
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
The capacity of the $q$-ary partially stuck-at level $s$ channel is

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

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

$$R_q^{\text{max}}(p, s) = \begin{cases} 
1 - \frac{2sp}{q} \log_q (s + 1) & \text{if } p \leq \frac{q}{2s} \log_{s+1} \left( \frac{q}{q-s} \right) \\
\log_q (q - s) & \text{else}
\end{cases}$$
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
Overview of our Constructions

Partially stuck-at level 1:

<table>
<thead>
<tr>
<th>Upper bound on $r_q(n, u, 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u &lt; q$</td>
</tr>
<tr>
<td>$u \geq q$</td>
</tr>
<tr>
<td>any $u, q$</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Upper bound on $r_q(n, u, s)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sum_{i=0}^{u-1} s_i &lt; q$</td>
</tr>
<tr>
<td>$u \geq \left\lceil \frac{q}{\max_i s_i} \right\rceil$</td>
</tr>
<tr>
<td>any $u, q$</td>
</tr>
</tbody>
</table>
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
- new codes for cells with unreliable levels
- 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
- new codes for cells with unreliable levels
- capacity analysis

Outlook
- better code constructions
- combination with additional random errors

Thank you! Questions?
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 $s = (s_0, s_1, \ldots, s_{u-1}) \in \{1, \ldots, q - 1\}^u$. If $\sum_{i=0}^{u-1} s_i < q$, then there exists a $(u, s)$-PSMC over $[q]$ with redundancy

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

However, for general $s$, it is not clear if this construction is asymptotically optimal...
Construction Based on Binary $u$-SMCs—Example (i)

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

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

- messages $m = (0, 3, 2, 1, 2, 2, 3, 1, 3, 2, 2) \in [3]^{11}$, $m' = (1, 0, 1) \in [1]^3$
- define $w = (0, 0, 0, 0, 0, 3, 2, 1, 2, 2, 3, 1, 3, 2, 2, 0)$
- find $z \in [q]$ s.t. the number of 0s/3s is minimal in 
  \( (w^{(z)})_U = (w + z \cdot (1, 1, \ldots, 1))_U: \)
  $\implies z = 1$ and $w^{(z)} = (1, 1, 1, 1, 0, 3, 2, 3, 3, 0, 2, 0, 3, 3, 1)$
- mask the 2 usual stuck cells in $v = (0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0)$
  with binary 2-SMC: 
  $\tilde{c} = ((1, 0, 0, 0) \cdot H, 0) = (1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0)$
- encode additional message: $\tilde{m} = (2, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$
- store $y = w^{(z)} + \tilde{c} + \tilde{m} = (0, 1, 3, 1, 1, 0, 3, 2, 3, 0, 1, 3, 1, 0, 0, 1)$
Example: \( n = 15, q = 4, u = 5 \) and \( U = \{1, 4, 8, 12, 15\} \).

\[ \tilde{u} = \lfloor 2u/q \rfloor = 2, \text{ use } [15, 11, 3]_2 \text{ code } \tilde{C} \text{ as binary 2-SMC with} \]

\[ H = \begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1
0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1
\end{pmatrix}. \]

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

given \( y = (0, 1, 3, 1, 1, 0, 3, 2, 3, 0, 1, 3, 1, 0, 0, 1) \)

- recover \( z: \ y_3 - y_{15} = 0 \iff \hat{z} = 1 \)
- \( \hat{y} = y - \hat{z} \cdot (1, 1, \ldots, 1) = (3, 0, 2, 0, 0, 3, 2, 1, 2, 3, 0, 2, 0, 3, 3, 0) \)
- \( \hat{m}' = ([\hat{y}_0/2], \ldots, [\hat{y}_{n-k-2}/2]) = ([3/2], 0, [2/2]) = (1, 0, 1) \)
- \( \hat{t} = (\hat{y}_0 - 2\hat{m}', \ldots, \hat{y}_{n-k-2} - 2, \hat{m}'_{n-k-2}, \hat{y}_{n-k-1}) \mod q = (3 - 2, 0 - 0, 2 - 2, 0) = (1, 0, 0, 0) \)
- \( \hat{c}' = \hat{t} \cdot H = (1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1) \)
- \( \hat{m} = (0, 3, 2, 1, 2, 2, 3, 1, 3, 2, 2) \).

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