Explicit Constructions of **Finite-Length Write-Once Memory (WOM)** Codes

Han Mao Kiah, Nanyang Technological University, Singapore

Joint work with:

Yeow Meng Chee, Nanyang Technological University

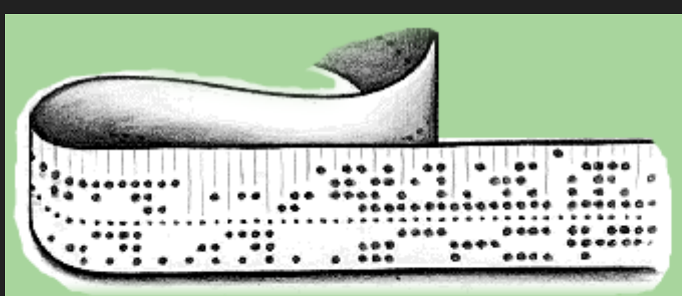
Alexander Vardy, University of California, San Diego

Eitan Yaakobi, Technion, Israel Institute of Technology

Punch-22 Puzzle

realmode.com/punch22.html

- Punch-22 use paper tape machines to store data.
- Once the data from a roll has been read and processed, the tape is of no further use and is thrown away.
- Because of recent budget cuts, they want to know if there is some way to reuse the roll.

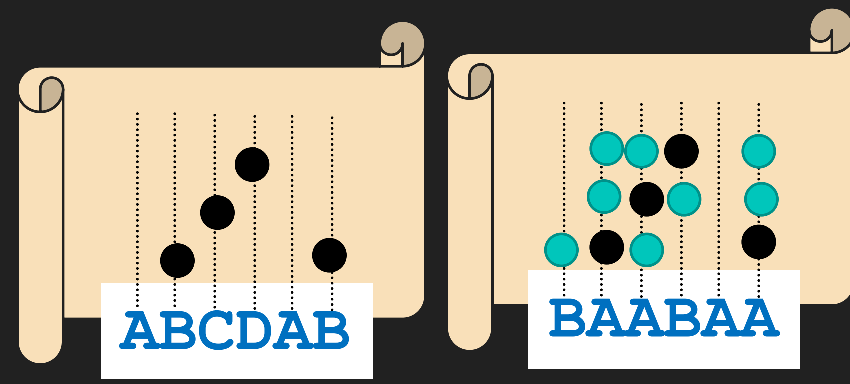


Task

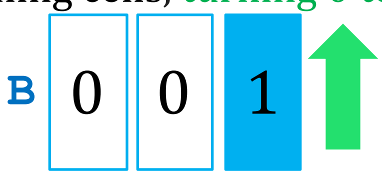
Come up with a **7-bit code** for the **symbols A-Z** that allows the tape to be reused once.

Write-Once Memory (WOM) Codes

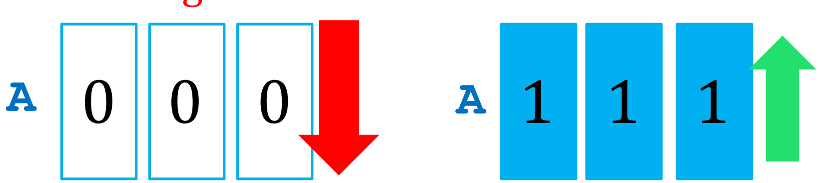
- In 1982, introduced by Rivest and Shamir, *How to Reuse a "Write-Once" Memory*
- In last decade, renewed interest due to applications in flash memories.



Programming cells, turning 0 to 1, is ***easy***.



But turning 1 to 0 is ***hard***.



First Write	
A	000
B	001
C	010
D	100

Second Write	
A	000, 111
B	001, 110
C	010, 101
D	100, 011

WOM Codes

Definition

An $[n, t; M_1, M_2, \dots, M_t]$ t-write WOM code is defined by

- t encoding/decoding maps $(E_1, D_1), \dots, (E_t, D_t)$
  $$E_i: [M_i] \times Im(E_{i-1}) \rightarrow \{0,1\}^n$$
- such that the encoding maps have the property:
  $$E_i(\boldsymbol{m}, \boldsymbol{c}) \geq \boldsymbol{c}.$$
- The decoding maps have the property:
  $$D_i\big(E_i(\boldsymbol{m}, \boldsymbol{c})\big) = \boldsymbol{m}.$$

A [3,2; 4,4] 2-write WOM code

First Write	
A	000
B	001
C	010
D	100

Second Write	
A	000, 111
B	001, 110
C	010, 101
D	100, 011

(E_1, D_1) (E_2, D_2)

$$E_2(A, 001) = 111 \geq 001$$

new message — current state — new state

WOM Codes

Definition

An $[n, t; M_1, M_2, \dots, M_t]$ t-write WOM code is defined by

- t encoding/decoding maps $(E_1, D_1), \dots, (E_t, D_t)$

$$E_i : [M_i] \times Im(E_{i-1}) \rightarrow \{0,1\}^n$$

- such that the encoding maps have the property:

$$E_i(\boldsymbol{m}, \boldsymbol{c}) \geq \boldsymbol{c}.$$

- Let $R_i = \frac{\log M_i}{n}$ for $1 \leq i \leq t$.
- The rate tuple is given by $(R_1, R_2, \dots, R_t)$.
- The sum-rate is given by $\sum_{i=1}^{t} R_1$.

A [3,2; 4,4] 2-write WOM code

First Write	
A	000
B	001
C	010
D	100

Second Write	
A	000, 111
B	001, 110
C	010, 101
D	100, 011

(E_1, D_1) (E_2, D_2)

$R_1 = \frac{2}{3}$ and $R_2 = \frac{2}{3}$.

The sum-rate is $\frac{4}{3} = 1.33$.

WOM Codes

What we want:

- High Rate-tuples
- High Sum-rates
- Efficient Encoding / Decoding Maps

Definition

An $[n, t; M_1, M_2, \dots, M_t]$ t-write WOM code is defined by

- t encoding/decoding maps $(E_1, D_1), \dots, (E_t, D_t)$

$$E_i: [M_i] \times Im(E_{i-1}) \rightarrow \{0,1\}^n$$

- such that the encoding maps have the property:

$$E_i(\boldsymbol{m}, \boldsymbol{c}) \geq \boldsymbol{c}.$$

- The decoding maps have the property:

$$D_i\big(E_i(\boldsymbol{m}, \boldsymbol{c})\big) = \boldsymbol{m}.$$

- Let $R_i = \frac{\log M_i}{n}$ for $1 \leq i \leq t$.
- The rate tuple is given by $(R_1, R_2, \dots, R_t)$.
- The sum-rate is given by $\sum_{i=1}^{t} R_1$.

Capacity Region

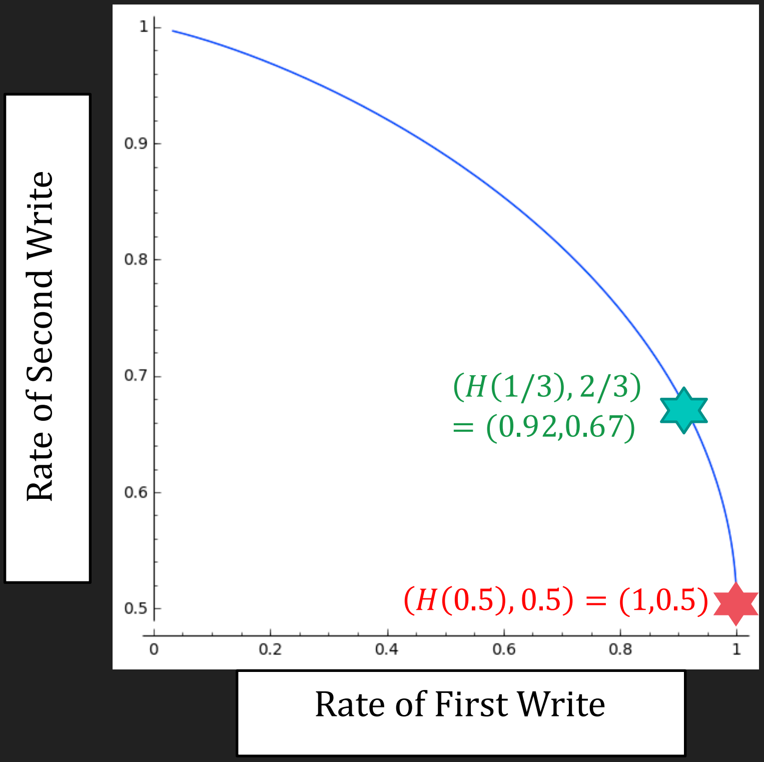
- (Heegard 1985, Fu and Vinck 1999) The rate tuple of a binary t-write WOM code belongs to the following capacity region.

$$\left\{ (R_1, R_2, \dots, R_t): \begin{array}{c} R_1 \le h(p_1), R_2 \le (1-p_1)h(p_2), \\ \dots, R_t \le \prod_{i=1}^{t-1}(1-p_i) \\ 0 \le p_1, p_2, \dots, p_{t-1} \le 1/2 \end{array} \right\}.$$

- When $t = 2$, the capacity region is

$$\left\{ (R_1, R_2): \begin{array}{c} R_1 \le h(p), R_2 \le (1-p) \\ 0 \le p \le 1/2 \end{array} \right\}.$$

- Maximum sum-rate is 1.58.



Previous Work
Capacity can be achieved, but...

Yaakobi et al. 2012

- Capacity-achieving two-writes WOM codes. However, encoding and decoding maps are complex
- Best sum-rate with two writes: 1.4928
- Best sum-rate with three writes: 1.61

Shpilka 2013

- Capacity-achieving two-writes WOM codes. However, code lengths need to be long
- Best sum-rate with two writes: 1.58 asymptotically
- Best sum-rate with three writes: 1.809 asymptotically

Yaakobi and Shpilka 2014

- Best sum-rate with three writes: 1.885 asymptotically

Shpilka 2014

- Capacity-achieving WOM for any number of writes. However, code lengths need to be long

This paper: WOM codes with

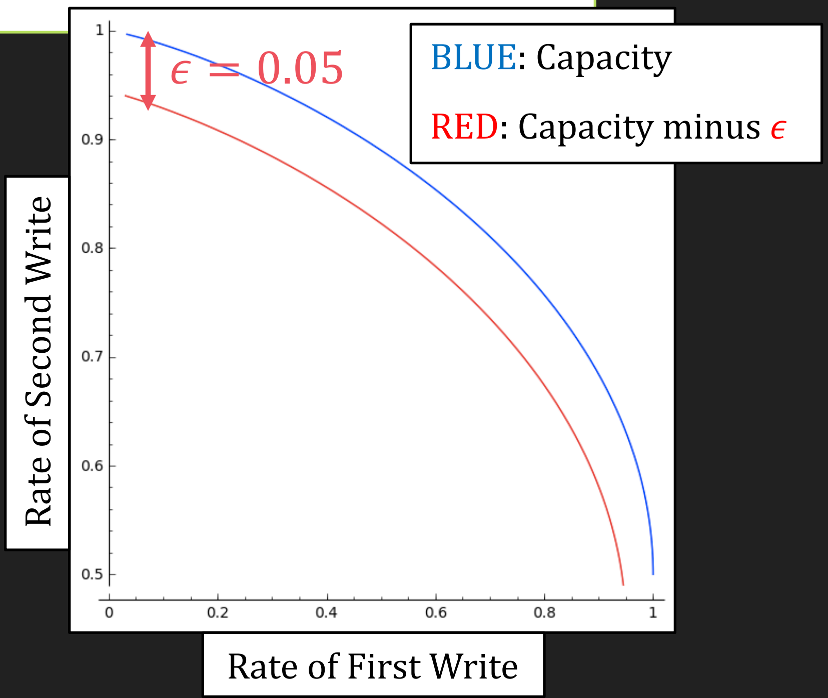
- efficient encoding /decoding maps
- short block lengths
- near capacity achieving

Convergence Rate

Definition

Given a family of t-write WOM codes, the convergence rate $n(\epsilon)$ to a rate tuple $(R_1, R_2, \ldots, R_t)$ if there is a WOM code in the family of length $n(\epsilon)$ such that its rate tuple is at least $(R_1 - \epsilon, R_2 - \epsilon, \ldots, R_t - \epsilon)$.

- We are interested in constructions that yield convergence rates $n(\epsilon)$ that are polynomial in $1/\epsilon$.
- Shpilka 2013, Shipilka 2014: Constructions rely on Wozencraft ensemble. Convergence rate is exponential in $1/\epsilon$.



Broad Strategy for 2-Write WOM

A [6,2; 22,9] 2-write WOM code with $R_1 = 0.743, R_2 = 0.528.$ Sum-rate = 1.272.

First Write					
A	000000	H	000011	O	010010
B	000001	I	000101	P	100010
C	000010	J	001001	Q	001100
D	000100	K	010001	R	010100
E	001000	L	100001	S	100100
F	010000	M	000110	T	011000
G	100000	N	001010	U	101000
				V	110000

Second Write	
A	011111, 010100, 000101, 110100, 101110, 001110, 100101
B	101111, 011100, 100010, 001100, 000001, 010001, 110010
C	110111, 011000, 000111, 010000, 100000, 101000, 001111
D	111011, 011010, 100011, 011110, 000110, 000010, 100111
E	111101, 011011, 110001, 011001, 010101, 010111, 110011
F	111110, 101101, 111000, 101100, 101010, 101011, 111001
G	001001, 110110, 001010, 000000, 000011, 110101, 111100
H	100100, 001101, 010011, 010110, 100001, 111010, 001000
I	000100, 100110, 101001, 011101, 110000, 001011, 010010

Broad Strategy for 2-Write WOM

A [6,2; 22,9] 2-write WOM code with $R_1 = 0.743, R_2 = 0.528$. Sum-rate = 1.272.

$n = 6, \tau = 2$

First Write					
A	000000	H	000011	O	010010
B	000001	I	000101	P	100010
C	000010	J	001001	Q	001100
D	000100	K	010001	R	010100
E	001000	L	100001	S	100100
F	010000	M	000110	T	011000
G	100000	N	001010	U	101000
				V	110000

On the first write, we program at most τ cells. So,

$$R_1 = \frac{\log\left(\sum_{i=0}^{\tau}\binom{n}{i}\right)}{n} \approx h\left(\frac{\tau}{n}\right)$$

Broad Strategy for 2-Write WOM

A [6,2; 22,9] 2-write WOM code with $R_1 = 0.743, R_2 = 0.528.$
Sum-rate = 1.272.

$n = 6, \tau = 2$

On the second write, we will the write $n - (\tau + 1)$ bits (somehow).
So,

$$R_2 = 1 - \frac{\tau + 1}{n} \approx 1 - \frac{\tau}{n}$$

Second Write	
A	011111, 010100, 000101, 110100, 101110, 001110, 100101
B	101111, 011100, 100010, 001100, 000001, 010001, 110010
C	110111, 011000, 000111, 010000, 100000, 101000, 001111
D	111011, 011010, 100011, 011110, 000110, 000010, 100111
E	111101, 011011, 110001, 011001, 010101, 010111, 110011
F	111110, 101101, 111000, 101100, 101010, 101011, 111001
G	001001, 110110, 001010, 000000, 000011, 110101, 111100
H	100100, 001101, 010011, 010110, 100001, 111010, 001000
I	000100, 100110, 101001, 011101, 110000, 001011, 010010

Broad Strategy for 2-Write WOM

A [6,2; 22,9] 2-write WOM code with $R_1 = 0.743, R_2 = 0.528$.
Sum-rate = 1.272.

$n = 6, \tau = 2$

On the first write, we program at most τ cells.
So,

$$R_1 = \frac{\log\left(\sum_{i=0}^{\tau}\binom{n}{i}\right)}{n} \approx h\left(\frac{\tau}{n}\right)$$

On the second write, we will the write $n - (\tau + 1)$ bits (somehow).
So,

$$R_2 = 1 - \frac{\tau + 1}{n} \approx 1 - \frac{\tau}{n}$$

Spreads and Partial Spreads

- A partial τ-spread of F_2^n is a collection of τ-dimensional subspaces $V_1, V_2, \dots, V_M$ of F_2^n such that
  - $V_i \cap V_j = \{\mathbf{0}\}$ for all $i \neq j$.
- If $\bigcup V_i = F_2^n$, then $V_1, V_2, \dots, V_M$ is called a τ-spread.

A 3-spread of F_2^6

$$
\begin{aligned}
V_1 &= \text{span}\{100000, 101011, 111010\},\\
V_2 &= \text{span}\{010000, 100011, 011101\},\\
V_3 &= \text{span}\{001000, 100111, 111000\},\\
V_4 &= \text{span}\{000100, 100101, 011100\},\\
V_5 &= \text{span}\{000010, 100100, 001110\},\\
V_6 &= \text{span}\{000001, 010010, 000111\},\\
V_7 &= \text{span}\{110110, 001001, 110101\},\\
V_8 &= \text{span}\{011011, 110010, 101100\},\\
V_9 &= \text{span}\{111011, 011001, 010110\}.
\end{aligned}
$$

Spreads for Second Write

S is the set of cells that was programmed on the **first write**.

- Let $V_1, V_2, \dots, V_M$ be a partial $(\tau + 1)$-spread.
- Set $V_i^* = V_i \setminus \{\mathbf{0}\}$.

Message for **second write**

Encoding: $[M] \times Im(E_1) \rightarrow \{0,1\}^n$

- Given V_i^* and S a subset of size at most τ, we find $\boldsymbol{x} \in V_i^*$ s.t. $x_j = 0$ for all $j \in S$.
- Write the codeword $\boldsymbol{y} = \overline{\boldsymbol{x}}$.

Decoding: $\{0,1\}^n \rightarrow [M]$

- Given $\boldsymbol{y}$, find the subspace V_i containing $\overline{\boldsymbol{y}}$.
- i is the decoded message.

Definition

A partial τ-spread of F_2^n is a collection of τ-dim subspaces $V_1, V_2, \dots, V_M$ of F_2^n such that $V_i \cap V_j = \{\mathbf{0}\}$ for all $i \neq j$.

$$V_1^* = \text{span}\{100000, 101011, 111010\} \setminus \{\mathbf{0}\},$$
$$V_2^* = \text{span}\{010000, 100011, 011101\} \setminus \{\mathbf{0}\},$$
$$V_3^* = \text{span}\{001000, 100111, 111000\} \setminus \{\mathbf{0}\},$$
$$V_4^* = \text{span}\{000100, 100101, 011100\} \setminus \{\mathbf{0}\},$$
$$V_5^* = \text{span}\{000010, 100100, 001110\} \setminus \{\mathbf{0}\},$$
$$V_6^* = \text{span}\{000001, 010010, 000111\} \setminus \{\mathbf{0}\},$$
$$V_7^* = \text{span}\{110110, 001001, 110101\} \setminus \{\mathbf{0}\},$$
$$V_8^* = \text{span}\{011011, 110010, 101100\} \setminus \{\mathbf{0}\},$$
$$V_9^* = \text{span}\{111011, 011001, 010110\} \setminus \{\mathbf{0}\}.$$

Instead of ‘1’ staying as ‘1’, we require ‘0’ to stay as ‘0’.

‘proof’

Spreads for Second Write

S is the set of cells that was programmed on the **first write**.

- Let $V_1, V_2, \dots, V_M$ be a partial $(\tau + 1)$-spread.
- Set $V_i^* = V_i \setminus \{\mathbf{0}\}$. ← Message for **second write**

Encoding: $[M] \times Im(E_1) \to \{0,1\}^n$

- Given V_i^* and S a subset of size at most τ, we find $\boldsymbol{x} \in V_i^*$ s.t. $x_j = 0$ for all $j \in S$.
- Write the codeword $\boldsymbol{y} = \overline{\boldsymbol{x}}$.

Decoding: $\{0,1\}^n \to [M]$

- Given $\boldsymbol{y}$, find the subspace V_i containing $\overline{\boldsymbol{y}}$.
- i is the decoded message.

Definition

A partial τ-spread of F_2^n is a collection of τ-dim subspaces $V_1, V_2, \dots, V_M$ of F_2^n such that $V_i \cap V_j = \{\mathbf{0}\}$ for all $i \neq j$.

$V_1^* = \text{span}\{100000, 101011, 111010\} \setminus \{\mathbf{0}\}$,
$V_2^* = \text{span}\{010000, 100011, 011101\} \setminus \{\mathbf{0}\}$,
$V_3^* = \text{span}\{001000, 100111, 111000\} \setminus \{\mathbf{0}\}$,
$V_4^* = \text{span}\{000100, 100101, 011100\} \setminus \{\mathbf{0}\}$,
$V_5^* = \text{span}\{000010, 100100, 001110\} \setminus \{\mathbf{0}\}$,
$V_6^* = \text{span}\{000001, 010010, 000111\} \setminus \{\mathbf{0}\}$,
$V_7^* = \text{span}\{110110, 001001, 110101\} \setminus \{\mathbf{0}\}$,
$V_8^* = \text{span}\{011011, 110010, 101100\} \setminus \{\mathbf{0}\}$,
$V_9^* = \text{span}\{111011, 011001, 010110\} \setminus \{\mathbf{0}\}$.

‘proof’
Spreads for Second Write

Definition

A partial τ-spread of F_2^n is a collection of τ-dim subspaces $V_1, V_2, \ldots, V_M$ of F_2^n such that $V_i \cap V_j = \{\mathbf{0}\}$ for all $i \neq j$.

S is the set of cells that was programmed on the **first write**.

- Let $V_1, V_2, \ldots, V_M$ be a partial $(\tau+1)$-spread.
- Set $V_i^* = V_i \setminus \{\mathbf{0}\}$. ← Message for **second write**

Encoding: $[M] \times Im(E_1) \rightarrow \{0,1\}^n$

- Given V_i^* and S a subset of size at most τ, we find $\boldsymbol{x} \in V_i^*$ s.t. $x_j = 0$ for all $j \in S$.
- Write the codeword $\boldsymbol{y} = \overline{\boldsymbol{x}}$.

Decoding: $\{0,1\}^n \rightarrow [M]$

- Given $\boldsymbol{y}$, find the subspace V_i containing $\overline{\boldsymbol{y}}$.
- i is the decoded message.

$$V_1^* = \text{span}\{100000, 101011, 111010\} \setminus \{\mathbf{0}\},$$
$$V_2^* = \text{span}\{010000, 100011, 011101\} \setminus \{\mathbf{0}\},$$
$$V_3^* = \text{span}\{001000, 100111, 111000\} \setminus \{\mathbf{0}\},$$
$$V_4^* = \text{span}\{000100, 100101, 011100\} \setminus \{\mathbf{0}\},$$
$$V_5^* = \text{span}\{000010, 100100, 001110\} \setminus \{\mathbf{0}\},$$
$$V_6^* = \text{span}\{000001, 010010, 000111\} \setminus \{\mathbf{0}\},$$
$$V_7^* = \text{span}\{110110, 001001, 110101\} \setminus \{\mathbf{0}\},$$
$$V_8^* = \text{span}\{011011, 110010, 101100\} \setminus \{\mathbf{0}\},$$
$$V_9^* = \text{span}\{111011, 011001, 010110\} \setminus \{\mathbf{0}\}.$$

Instead of ‘1’ staying as ‘1’, we require ‘0’ to stay as ‘0’.

‘proof’

Spreads for Second Write

S is the set of cells that was programmed on the **first write**.

- Let $V_1, V_2, \ldots, V_M$ be a partial $(\tau + 1)$-spread.
- Set $V_i^* = V_i \setminus \{\mathbf{0}\}$. ← Message for **second write**

Encoding: $[M] \times Im(E_1) \rightarrow \{0,1\}^n$

- Given V_i^* and S a subset of size at most τ, we find $\boldsymbol{x} \in V_i^*$ s.t. $x_j = 0$ for all $j \in S$.
- Write the codeword $\boldsymbol{y} = \overline{\boldsymbol{x}}$.

Decoding: $\{0,1\}^n \rightarrow [M]$

- Given $\boldsymbol{y}$, find the subspace V_i containing $\overline{\boldsymbol{y}}$.
- i is the decoded message.

Definition

A partial τ-spread of F_2^n is a collection of τ-dim subspaces $V_1, V_2, \ldots, V_M$ of F_2^n such that $V_i \cap V_j = \{\mathbf{0}\}$ for all $i \neq j$.

Consider the codeset V_1^* and $S = \{1,2\}$.

$V_1^* = \text{span}\{100000, 101011, 111010\} \setminus \{\mathbf{0}\}$.

Want to find $\boldsymbol{x} \in V_1^*$ such that

$\boldsymbol{x}_j = 0$ for all $j \in S$, or,

$x_1 = x_2 = 0$.

Instead of ‘1’ staying as ‘1’, we require ‘0’ to stay as ‘0’.

‘proof’

Spreads for Second Write

Consider the codeset V_1^* and $S = \{1,2\}$.

$$V_1^* = \text{span}\{100000, 101011, 111010\} \setminus \{\mathbf{0}\}.$$

Want to find

$$\boldsymbol{x} \in V_1^* \text{ such that } x_1 = x_2 = 0. \qquad (*)$$

- Let K be the collection of vectors that satisfy (*).
- Suppose $\boldsymbol{x}, \boldsymbol{y} \in K$. i.e.
  - $\boldsymbol{x} \in V_1^*$ such that $x_1 = x_2 = 0$.
  - $\boldsymbol{y} \in V_1^*$ such that $y_1 = y_2 = 0$.
- Then $\boldsymbol{x} + \boldsymbol{y} \in K \cup \{\mathbf{0}\}$.
- In other words, $K \cup \{\mathbf{0}\}$ is a vector subspace of V_1.

'proof'
Spreads for Second Write

Consider the codeset V_1^* and $S = \{1,2\}$.

$$V_1^* = \text{span}\{100000, 101011, 111010\} \setminus \{\mathbf{0}\}.$$

Want to find

$$\boldsymbol{x} \in V_1^* \text{ such that } x_1 = x_2 = 0. \qquad (*)$$

- Let K be the collection of vectors that satisfy (*).
- In fact, $K \cup \{\mathbf{0}\}$ is the kernel of the map ϕ that projects V_1 onto the coordinates at S.

$$\phi: V_1 \to F_2^{|S|}$$

- Since V_1 has dimension three and its image has dimension two, its kernel $K \cup \{\mathbf{0}\}$ must be nontrivial.
- So, there exists nonzero $\boldsymbol{x}$ that satisfies (*).

Spreads for Two-Write WOM

Theorem Suppose that

- there is a $(\tau + 1)$-partial spread of size M;
- at most τ cells are programmed on the first write.

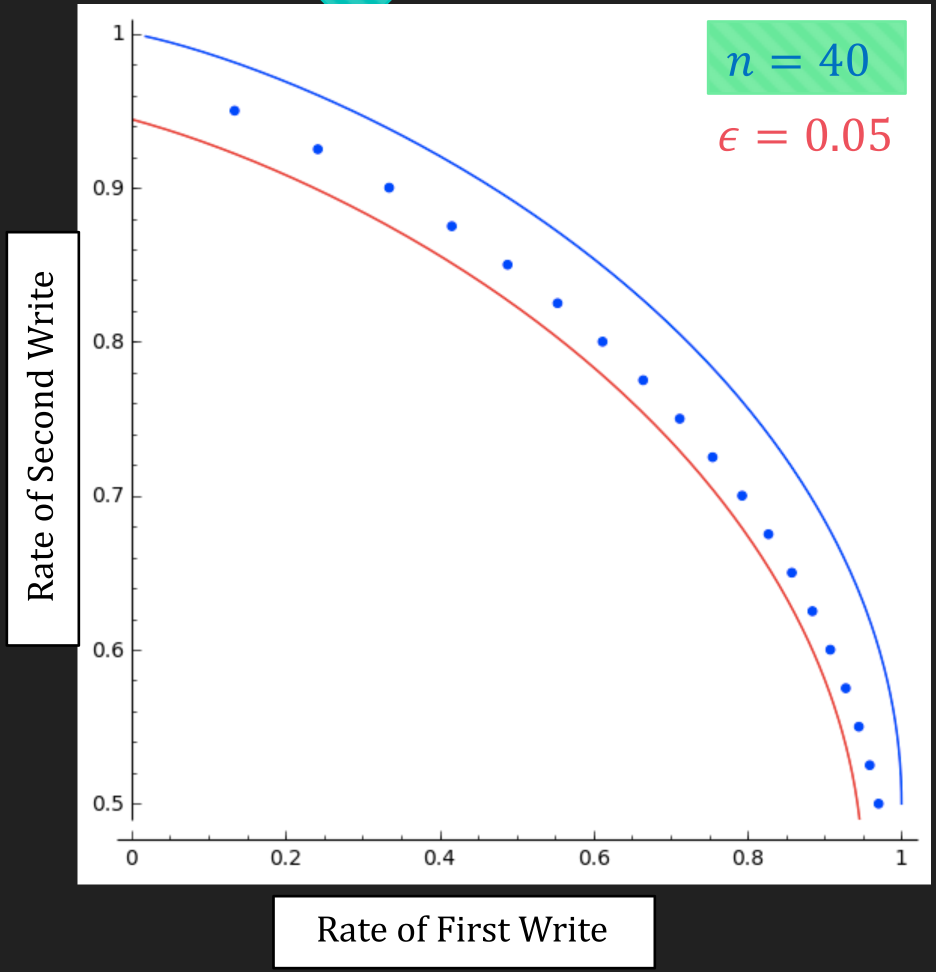
Then we can write at least M messages on the second write.

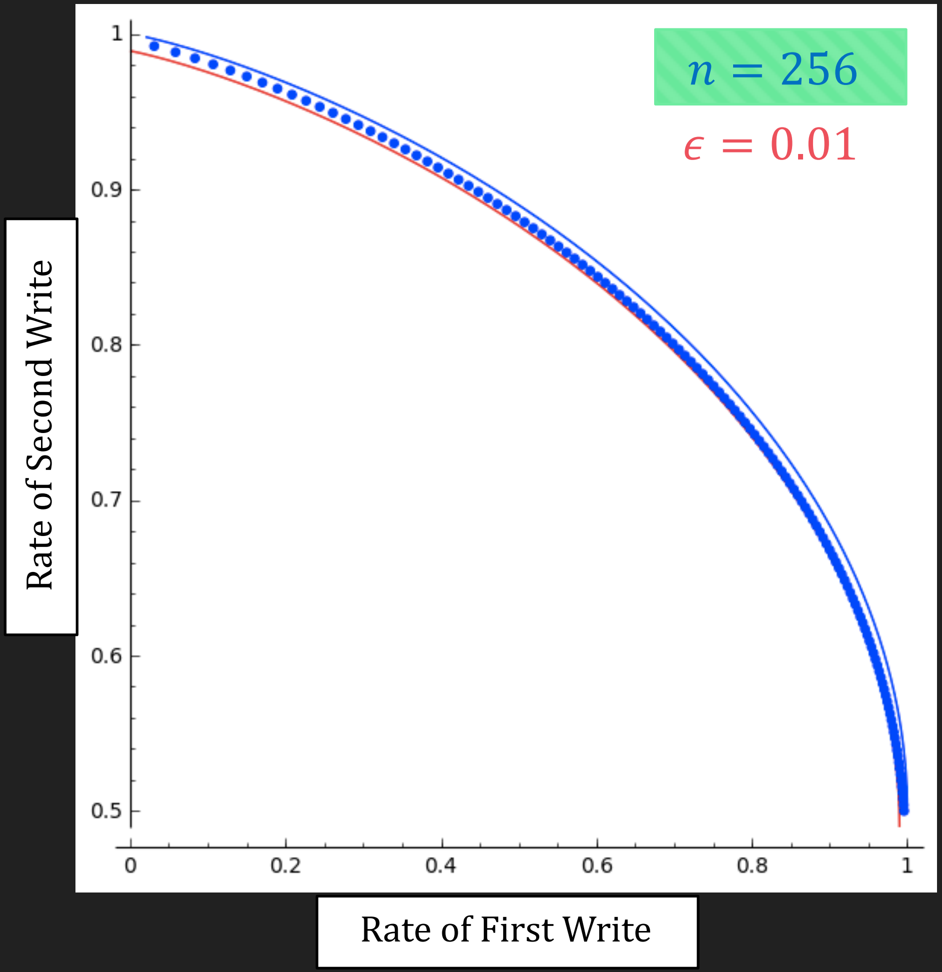
Theorem Let $\tau \leq n/2$. There is a τ-partial spread of size at least $2^{n-\tau}$.

Theorem Let $\tau + 1 \leq n/2$.

Then there is an $\left[n, 2; \sum_{i=0}^{\tau} \binom{n}{i}, 2^{n-\tau-1}\right]$ two-write WOM code.

Performance for Finite Lengths





Convergence Rate

Theorem Let $\tau + 1 \leq n/2$.

Then there is an $\left[n, 2; \sum_{i=0}^{\tau} \binom{n}{i}, 2^{n-\tau-1}\right]$ two-write WOM code.

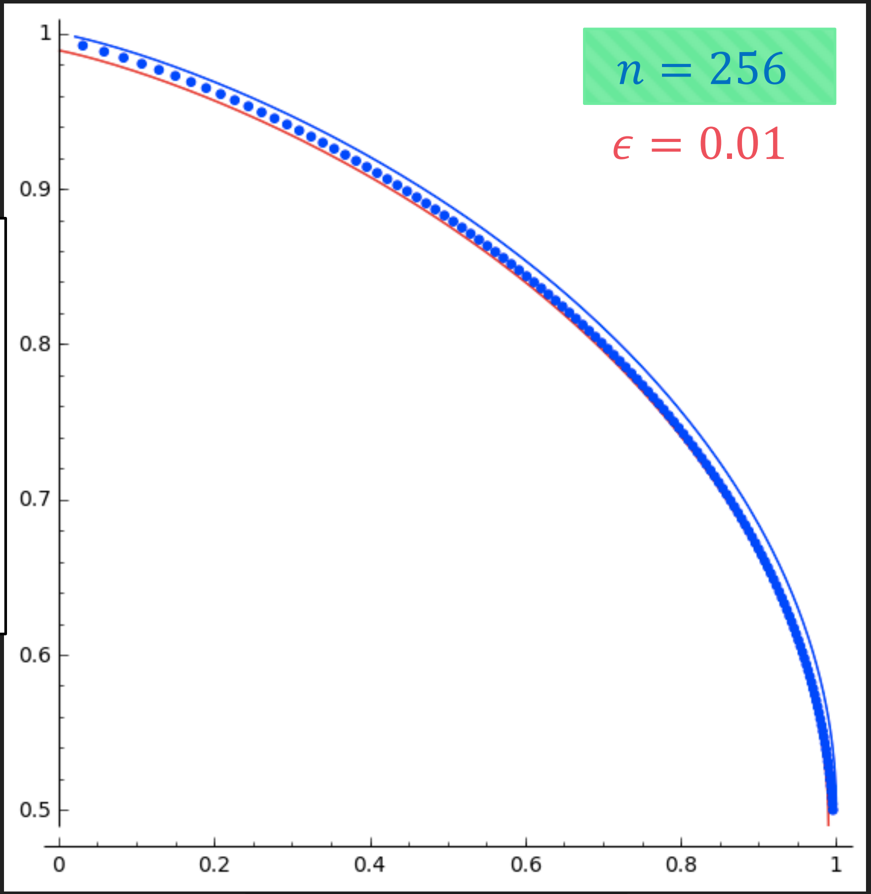
Convergence Rate

When $n = \frac{1}{\epsilon^2}$,

$$R_1 \geq h\left(\frac{\tau}{n}\right) - \epsilon, R_2 \geq 1 - \frac{\tau}{n} - \epsilon.$$

There exist **efficient** encoding and decoding maps (Dumer 1989).

Rate of Second Write



Rate of First Write

Three Writes WOM

- Unfortunately, we are unable to achieve capacity with our methods.
- Nevertheless, we can approach sum-rate 1.809 with polynomial convergence rate.
- For lengths of less than 100, we achieve sum-rate 1.71. Previous best sum-rate is 1.61.

Explicit Constructions of **Finite-Length Write-Once Memory (WOM)** Codes