Today's outline - April 12, 2022

Today's outline - April 12, 2022

- Measurements and quantum state transformations

Today's outline - April 12, 2022

- Measurements and quantum state transformations
- Decoherence

Today's outline - April 12, 2022

- Measurements and quantum state transformations
- Decoherence
- Single-bit bit-flip correction

Today's outline - April 12, 2022

- Measurements and quantum state transformations
- Decoherence
- Single-bit bit-flip correction
- Single-bit phase-flip correction

Today's outline - April 12, 2022

- Measurements and quantum state transformations
- Decoherence
- Single-bit bit-flip correction
- Single-bit phase-flip correction
- Error correcting framework

Today's outline - April 12, 2022

- Measurements and quantum state transformations
- Decoherence
- Single-bit bit-flip correction
- Single-bit phase-flip correction
- Error correcting framework
- Examples

Today's outline - April 12, 2022

- Measurements and quantum state transformations
- Decoherence
- Single-bit bit-flip correction
- Single-bit phase-flip correction
- Error correcting framework
- Examples

Reading assignment: 11.2 – 11.3

Today's outline - April 12, 2022

- Measurements and quantum state transformations
- Decoherence
- Single-bit bit-flip correction
- Single-bit phase-flip correction
- Error correcting framework
- Examples

Reading assignment: 11.2 – 11.3

Homework Assignment #07:
Chapter 9:2,3,4; Chapter 10:3,4,11
Due Thursday, April 21, 2022

Today's outline - April 12, 2022

- Measurements and quantum state transformations
- Decoherence
- Single-bit bit-flip correction
- Single-bit phase-flip correction
- Error correcting framework
- Examples

Reading assignment: 11.2 – 11.3

Exam #2 - April 26, 2022

Homework Assignment #07:
Chapter 9:2,3,4; Chapter 10:3,4,11
Due Thursday, April 21, 2022

Today's outline - April 12, 2022

- Measurements and quantum state transformations
- Decoherence
- Single-bit bit-flip correction
- Single-bit phase-flip correction
- Error correcting framework
- Examples

Reading assignment: 11.2 – 11.3

Homework Assignment #07:
Chapter 9:2,3,4; Chapter 10:3,4,11
Due Thursday, April 21, 2022

Exam #2 - April 26, 2022

Final Exam will be 10 minute presentations on a quantum computing journal article

Today's outline - April 12, 2022

- Measurements and quantum state transformations
- Decoherence
- Single-bit bit-flip correction
- Single-bit phase-flip correction
- Error correcting framework
- Examples

Reading assignment: 11.2 – 11.3

Homework Assignment #07:
Chapter 9:2,3,4; Chapter 10:3,4,11
Due Thursday, April 21, 2022

Exam #2 - April 26, 2022

Final Exam will be 10 minute presentations on a quantum computing journal article

Get my approval by Friday, April 15, 2022

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

If U is any unitary operator on $X = A \otimes B$
such that

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

If U is any unitary operator on $X = A \otimes B$ such that

$$U: \ |\psi\rangle|\phi\rangle \mapsto \sum_{i=0}^{M-1} P_i|\psi\rangle|\beta_i\rangle$$

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

If U is any unitary operator on $X = A \otimes B$ such that

$$U: \ |\psi\rangle|\phi\rangle \mapsto \sum_{i=0}^{M-1} P_i|\psi\rangle|\beta_i\rangle$$

If $\rho = |\psi\rangle\langle\psi|$ then the superoperator S_U^ϕ is

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

If U is any unitary operator on $X = A \otimes B$ such that

$$U: \ |\psi\rangle|\phi\rangle \mapsto \sum_{i=0}^{M-1} P_i|\psi\rangle|\beta_i\rangle$$

If $\rho = |\psi\rangle\langle\psi|$ then the superoperator S_U^ϕ is

$$S_U^\phi = \text{Tr}_B\left(U(\rho \otimes |\phi\rangle\langle\phi|)U^\dagger\right)$$

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

If U is any unitary operator on $X = A \otimes B$ such that

$$U: \ |\psi\rangle|\phi\rangle \mapsto \sum_{i=0}^{M-1} P_i|\psi\rangle|\beta_i\rangle$$

If $\rho = |\psi\rangle\langle\psi|$ then the superoperator S_U^ϕ is

$$S_U^\phi = \mathrm{Tr}_B \left(U(\rho \otimes |\phi\rangle\langle\phi|) U^\dagger \right)$$
$$= \sum_{i=0}^{M-1} A_i \rho A_i^\dagger$$

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

If U is any unitary operator on $X = A \otimes B$ such that

$$U: \; |\psi\rangle|\phi\rangle \mapsto \sum_{i=0}^{M-1} P_i|\psi\rangle|\beta_i\rangle$$

If $\rho = |\psi\rangle\langle\psi|$ then the superoperator S_U^ϕ is

$$S_U^\phi = \mathrm{Tr}_B\left(U(\rho \otimes |\phi\rangle\langle\phi|)U^\dagger\right)$$
$$= \sum_{i=0}^{M-1} A_i \rho A_i^\dagger = \sum_{i=0}^{M-1} A_i|\psi\rangle\langle\psi|A_i^\dagger$$

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

If U is any unitary operator on $X = A \otimes B$ such that

$$U: \ |\psi\rangle|\phi\rangle \mapsto \sum_{i=0}^{M-1} P_i|\psi\rangle|\beta_i\rangle$$

If $\rho = |\psi\rangle\langle\psi|$ then the superoperator S_U^ϕ is

$$S_U^\phi = \mathrm{Tr}_B \left(U(\rho \otimes |\phi\rangle\langle\phi|)U^\dagger \right)$$

Where $A_i = \langle\beta_i|U|\phi\rangle$ so that

$$= \sum_{i=0}^{M-1} A_i \rho A_i^\dagger = \sum_{i=0}^{M-1} A_i|\psi\rangle\langle\psi|A_i^\dagger$$

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

If U is any unitary operator on $X = A \otimes B$ such that

$$U: \; |\psi\rangle|\phi\rangle \mapsto \sum_{i=0}^{M-1} P_i|\psi\rangle|\beta_i\rangle$$

If $\rho = |\psi\rangle\langle\psi|$ then the superoperator S_U^ϕ is

$$S_U^\phi = \mathrm{Tr}_B \left(U(\rho \otimes |\phi\rangle\langle\phi|) U^\dagger \right)$$

$$= \sum_{i=0}^{M-1} A_i \rho A_i^\dagger = \sum_{i=0}^{M-1} A_i|\psi\rangle\langle\psi|A_i^\dagger$$

Where $A_i = \langle\beta_i|U|\phi\rangle$ so that

$$A_i|\psi\rangle = \sum_{j=0}^{N-1} \left(\langle\alpha_j|\langle\beta_i|U|\psi\rangle|\phi\rangle \right) |\alpha_j\rangle$$

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

If U is any unitary operator on $X = A \otimes B$ such that

$$U: \ |\psi\rangle|\phi\rangle \mapsto \sum_{i=0}^{M-1} P_i|\psi\rangle|\beta_i\rangle$$

If $\rho = |\psi\rangle\langle\psi|$ then the superoperator S_U^ϕ is

$$S_U^\phi = \mathrm{Tr}_B\left(U(\rho \otimes |\phi\rangle\langle\phi|)U^\dagger\right)$$

$$= \sum_{i=0}^{M-1} A_i \rho A_i^\dagger = \sum_{i=0}^{M-1} A_i|\psi\rangle\langle\psi|A_i^\dagger$$

Where $A_i = \langle\beta_i|U|\phi\rangle$ so that

$$A_i|\psi\rangle = \sum_{j=0}^{N-1}\left(\langle\alpha_j|\langle\beta_i|U|\psi\rangle|\phi\rangle\right)|\alpha_j\rangle = \sum_{j=0}^{N-1}\langle\alpha_j|\langle\beta_i|\left(\sum_{k=0}^{M-1} P_k|\psi\rangle|\beta_k\rangle\right)|\alpha_j\rangle$$

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

If U is any unitary operator on $X = A \otimes B$ such that

$$U: \ |\psi\rangle|\phi\rangle \mapsto \sum_{i=0}^{M-1} P_i|\psi\rangle|\beta_i\rangle$$

If $\rho = |\psi\rangle\langle\psi|$ then the superoperator S_U^ϕ is

$$S_U^\phi = \text{Tr}_B\left(U(\rho \otimes |\phi\rangle\langle\phi|)U^\dagger\right)$$

$$= \sum_{i=0}^{M-1} A_i \rho A_i^\dagger = \sum_{i=0}^{M-1} A_i|\psi\rangle\langle\psi|A_i^\dagger$$

Where $A_i = \langle\beta_i|U|\phi\rangle$ so that

$$A_i|\psi\rangle = \sum_{j=0}^{N-1}\left(\langle\alpha_j|\langle\beta_i|U|\psi\rangle|\phi\rangle\right)|\alpha_j\rangle = \sum_{j=0}^{N-1}\langle\alpha_j|\langle\beta_i|\left(\sum_{k=0}^{M-1} P_k|\psi\rangle|\beta_k\rangle\right)|\alpha_j\rangle$$

$$= \sum_{j=0}^{N-1}\langle\alpha_j|P_i|\psi\rangle\,|\alpha_j\rangle$$

Measurements and quantum state transformations

Let O be an operator with M distinct eigenvalues and suppose that B is a system of dimension M with basis $\{|\beta_i\rangle\}$

If B is initially in the state $|\phi\rangle$

If U is any unitary operator on $X = A \otimes B$ such that

$$U: \ |\psi\rangle|\phi\rangle \mapsto \sum_{i=0}^{M-1} P_i|\psi\rangle|\beta_i\rangle$$

If $\rho = |\psi\rangle\langle\psi|$ then the superoperator S_U^ϕ is

$$S_U^\phi = \mathrm{Tr}_B\left(U(\rho \otimes |\phi\rangle\langle\phi|)U^\dagger\right)$$

$$= \sum_{i=0}^{M-1} A_i \rho A_i^\dagger = \sum_{i=0}^{M-1} A_i|\psi\rangle\langle\psi|A_i^\dagger$$

Where $A_i = \langle\beta_i|U|\phi\rangle$ so that

$$A_i|\psi\rangle = \sum_{j=0}^{N-1}\left(\langle\alpha_j|\langle\beta_i|U|\psi\rangle|\phi\rangle\right)|\alpha_j\rangle = \sum_{j=0}^{N-1}\langle\alpha_j|\langle\beta_i|\left(\sum_{k=0}^{M-1} P_k|\psi\rangle|\beta_k\rangle\right)|\alpha_j\rangle$$

$$= \sum_{j=0}^{N-1}\langle\alpha_j|P_i|\psi\rangle\,|\alpha_j\rangle = P_i|\psi\rangle$$

Measurements and quantum state transformations

Since $A_i|\psi\rangle = P_i|\psi\rangle$ the superoperator S_U^{ϕ} becomes

Measurements and quantum state transformations

Since $A_i|\psi\rangle = P_i|\psi\rangle$ the superoperator S_U^ϕ becomes

$$S_U^\phi = \sum_{i=0}^{M-1} A_i|\psi\rangle\langle\psi|A_i^\dagger$$

Measurements and quantum state transformations

Since $A_i|\psi\rangle = P_i|\psi\rangle$ the superoperator S_U^ϕ becomes

$$S_U^\phi = \sum_{i=0}^{M-1} A_i|\psi\rangle\langle\psi|A_i^\dagger = \sum_{i=0}^{M-1} P_i|\psi\rangle\langle\psi|P_i^\dagger$$

Measurements and quantum state transformations

Since $A_i|\psi\rangle = P_i|\psi\rangle$ the superoperator S_U^ϕ becomes

$$S_U^\phi = \sum_{i=0}^{M-1} A_i|\psi\rangle\langle\psi|A_i^\dagger = \sum_{i=0}^{M-1} P_i|\psi\rangle\langle\psi|P_i^\dagger = \sum_{i=0}^{M-1} p_i \frac{P_i|\psi\rangle\langle\psi|P_i^\dagger}{\mathrm{Tr}(P_i|\psi\rangle\langle\psi|P_i^\dagger)}, \qquad p_i = \mathrm{Tr}(P_i|\psi\rangle\langle\psi|P_i^\dagger)$$

Measurements and quantum state transformations

Since $A_i|\psi\rangle = P_i|\psi\rangle$ the superoperator S_U^ϕ becomes

$$S_U^\phi = \sum_{i=0}^{M-1} A_i|\psi\rangle\langle\psi|A_i^\dagger = \sum_{i=0}^{M-1} P_i|\psi\rangle\langle\psi|P_i^\dagger = \sum_{i=0}^{M-1} p_i \frac{P_i|\psi\rangle\langle\psi|P_i^\dagger}{\mathrm{Tr}(P_i|\psi\rangle\langle\psi|P_i^\dagger)}, \qquad p_i = \mathrm{Tr}(P_i|\psi\rangle\langle\psi|P_i^\dagger)$$

This is just the density operator ρ' that represents the probabilistic mixture of outcomes of a measurement O with associated projectors P_i on system A initially represented by the mixed state $\rho = |\psi\rangle\langle\psi|$

Measurements and quantum state transformations

Since $A_i|\psi\rangle = P_i|\psi\rangle$ the superoperator S_U^ϕ becomes

$$S_U^\phi = \sum_{i=0}^{M-1} A_i|\psi\rangle\langle\psi|A_i^\dagger = \sum_{i=0}^{M-1} P_i|\psi\rangle\langle\psi|P_i^\dagger = \sum_{i=0}^{M-1} p_i \frac{P_i|\psi\rangle\langle\psi|P_i^\dagger}{\mathrm{Tr}(P_i|\psi\rangle\langle\psi|P_i^\dagger)}, \qquad p_i = \mathrm{Tr}(P_i|\psi\rangle\langle\psi|P_i^\dagger)$$

This is just the density operator ρ' that represents the probabilistic mixture of outcomes of a measurement O with associated projectors P_i on system A initially represented by the mixed state $\rho = |\psi\rangle\langle\psi|$

Thus for any observable O on system A there exists a larger system $X = A \otimes B$, a unitary operator $U : X \to X$, and a state $|\phi\rangle$ such that $S_U^\phi \equiv S_O$

Decoherence

Our understanding of how qubit systems can be manipulated and measured depends on the ability to isolate them completely from other interactions

Decoherence

Our understanding of how qubit systems can be manipulated and measured depends on the ability to isolate them completely from other interactions

Unfortunately, this is never possible as quantum systems are always part of a larger system consisting of the environment over which we have no control

Decoherence

Our understanding of how qubit systems can be manipulated and measured depends on the ability to isolate them completely from other interactions

Unfortunately, this is never possible as quantum systems are always part of a larger system consisting of the environment over which we have no control

When an interaction with the environment causes decoherence, information about the computational subsystem is lost

Decoherence

Our understanding of how qubit systems can be manipulated and measured depends on the ability to isolate them completely from other interactions

Unfortunately, this is never possible as quantum systems are always part of a larger system consisting of the environment over which we have no control

When an interaction with the environment causes decoherence, information about the computational subsystem is lost

Quantum error correction techniques depend on having a method of understanding the effect of a subsystem interaction with other subsystem, that is operator sum decomposition

Decoherence

Our understanding of how qubit systems can be manipulated and measured depends on the ability to isolate them completely from other interactions

Unfortunately, this is never possible as quantum systems are always part of a larger system consisting of the environment over which we have no control

When an interaction with the environment causes decoherence, information about the computational subsystem is lost

Quantum error correction techniques depend on having a method of understanding the effect of a subsystem interaction with other subsystem, that is operator sum decomposition

The effect of on the computational subsystem of any interaction with the environment can be viewed as a mixture of K errors resulting in K mixed states

Decoherence

Our understanding of how qubit systems can be manipulated and measured depends on the ability to isolate them completely from other interactions

Unfortunately, this is never possible as quantum systems are always part of a larger system consisting of the environment over which we have no control

When an interaction with the environment causes decoherence, information about the computational subsystem is lost

Quantum error correction techniques depend on having a method of understanding the effect of a subsystem interaction with other subsystem, that is operator sum decomposition

The effect of on the computational subsystem of any interaction with the environment can be viewed as a mixture of K errors resulting in K mixed states

The discussion on error correction will start by discussing three simple codes which correct single-qubit bit-flip errors, single-qubit phase errors, and all single-qubit errors

Single-qubit bit-flip correction

The bit-flip error effectively applies the X gate to one of the qubits in a subsystem

Single-qubit bit-flip correction

The bit-flip error effectively applies the X gate to one of the qubits in a subsystem

The error correction requires 3 qubits for each bit of information with an initial encoding of

Single-qubit bit-flip correction

The bit-flip error effectively applies the X gate to one of the qubits in a subsystem

The error correction requires 3 qubits for each bit of information with an initial encoding of

$$|0\rangle \rightarrow |000\rangle$$
$$|1\rangle \rightarrow |111\rangle$$

Single-qubit bit-flip correction

The bit-flip error effectively applies the X gate to one of the qubits in a subsystem

The error correction requires 3 qubits for each bit of information with an initial encoding of

$$|0\rangle \rightarrow |000\rangle$$
$$|1\rangle \rightarrow |111\rangle$$

This has the goal of correcting a bit-flip error in any of the three qubits

Single-qubit bit-flip correction

The bit-flip error effectively applies the X gate to one of the qubits in a subsystem

The error correction requires 3 qubits for each bit of information with an initial encoding of

$$|0\rangle \rightarrow |000\rangle$$
$$|1\rangle \rightarrow |111\rangle$$

This has the goal of correcting a bit-flip error in any of the three qubits

$$\{X_2 = X \otimes I \otimes I, X_1 = I \otimes X \otimes I, X_0 = I \otimes I \otimes X\}$$

Single-qubit bit-flip correction

The bit-flip error effectively applies the X gate to one of the qubits in a subsystem

The error correction requires 3 qubits for each bit of information with an initial encoding of

$$|0\rangle \rightarrow |000\rangle$$
$$|1\rangle \rightarrow |111\rangle$$

This has the goal of correcting a bit-flip error in any of the three qubits

$$\{X_2 = X \otimes I \otimes I, X_1 = I \otimes X \otimes I, X_0 = I \otimes I \otimes X\}$$

More formally, if C_{BF} is the subspace spanned by $\{|000\rangle, |111\rangle\}$, c_{BF} is a general encoding that takes single-qubit states into C_{BF}

Single-qubit bit-flip correction

The bit-flip error effectively applies the X gate to one of the qubits in a subsystem

The error correction requires 3 qubits for each bit of information with an initial encoding of

$$|0\rangle \rightarrow |000\rangle$$
$$|1\rangle \rightarrow |111\rangle$$

This has the goal of correcting a bit-flip error in any of the three qubits

$$\{X_2 = X \otimes I \otimes I, X_1 = I \otimes X \otimes I, X_0 = I \otimes I \otimes X\}$$

More formally, if C_{BF} is the subspace spanned by $\{|000\rangle, |111\rangle\}$, c_{BF} is a general encoding that takes single-qubit states into C_{BF}

$$c_{BF} : |0\rangle \otimes |00\rangle \rightarrow |000\rangle = |\tilde{0}\rangle$$
$$|1\rangle \otimes |00\rangle \rightarrow |111\rangle = |\tilde{1}\rangle$$

Single-qubit bit-flip correction

The bit-flip error effectively applies the X gate to one of the qubits in a subsystem

The error correction requires 3 qubits for each bit of information with an initial encoding of

$$|0\rangle \rightarrow |000\rangle$$
$$|1\rangle \rightarrow |111\rangle$$

This has the goal of correcting a bit-flip error in any of the three qubits

$$\{X_2 = X \otimes I \otimes I, X_1 = I \otimes X \otimes I, X_0 = I \otimes I \otimes X\}$$

More formally, if C_{BF} is the subspace spanned by $\{|000\rangle, |111\rangle\}$, c_{BF} is a general encoding that takes single-qubit states into C_{BF}

$$c_{BF} : |0\rangle \otimes |00\rangle \rightarrow |000\rangle = |\tilde{0}\rangle$$
$$|1\rangle \otimes |00\rangle \rightarrow |111\rangle = |\tilde{1}\rangle$$

Where $|\tilde{0}\rangle$ means the encoding of $|0\rangle$ and is considered to be a logical qubit composed of three computation qubits

Single-qubit bit-flip correction

The bit-flip error effectively applies the X gate to one of the qubits in a subsystem

The error correction requires 3 qubits for each bit of information with an initial encoding of

$$|0\rangle \rightarrow |000\rangle$$
$$|1\rangle \rightarrow |111\rangle$$

This has the goal of correcting a bit-flip error in any of the three qubits

$$\{X_2 = X \otimes I \otimes I, X_1 = I \otimes X \otimes I, X_0 = I \otimes I \otimes X\}$$

More formally, if C_{BF} is the subspace spanned by $\{|000\rangle, |111\rangle\}$, c_{BF} is a general encoding that takes single-qubit states into C_{BF}

$$c_{BF} : |0\rangle \otimes |00\rangle \rightarrow |000\rangle = |\tilde{0}\rangle$$
$$|1\rangle \otimes |00\rangle \rightarrow |111\rangle = |\tilde{1}\rangle$$

Where $|\tilde{0}\rangle$ means the encoding of $|0\rangle$ and is considered to be a logical qubit composed of three computation qubits

3-qubit states such as $|101\rangle$ are not legitimate logical qubit states which are referred to as codewords

Single-qubit bit-flip correction

Suppose a logical qubit is in a state

Single-qubit bit-flip correction

Suppose a logical qubit is in a state

$$a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

Single-qubit bit-flip correction

Suppose a logical qubit is in a state

$$a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If a bit flip error occurs in b_2 the state is not a codeword and not in C_{BF}

Single-qubit bit-flip correction

Suppose a logical qubit is in a state

$$a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If a bit flip error occurs in b_2 the state is not a codeword and not in C_{BF}

$$\longrightarrow a|100\rangle + b|011\rangle$$

Single-qubit bit-flip correction

Suppose a logical qubit is in a state

$$a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If a bit flip error occurs in b_2 the state is not a codeword and not in C_{BF}

$$\longrightarrow a|100\rangle + b|011\rangle$$

In order to detect the non-codeword states and transform them back to codewords, the transformation U_{BF}, called the syndrome extraction operator is used

Single-qubit bit-flip correction

Suppose a logical qubit is in a state

$$a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If a bit flip error occurs in b_2 the state is not a codeword and not in C_{BF}

$$\longrightarrow a|100\rangle + b|011\rangle$$

In order to detect the non-codeword states and transform them back to codewords, the transformation U_{BF}, called the syndrome extraction operator is used

$$U_{BF} : |b_2, b_1, b_0, 0, 0\rangle \longrightarrow |b_2, b_1, b_0, b_2 \oplus b_1, b_2 \oplus b_0\rangle$$

Single-qubit bit-flip correction

Suppose a logical qubit is in a state

$$a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If a bit flip error occurs in b_2 the state is not a codeword and not in C_{BF}

$$\longrightarrow a|100\rangle + b|011\rangle$$

In order to detect the non-codeword states and transform them back to codewords, the transformation U_{BF}, called the syndrome extraction operator is used

$$U_{BF} : |b_2, b_1, b_0, 0, 0\rangle \longrightarrow |b_2, b_1, b_0, b_2 \oplus b_1, b_2 \oplus b_0\rangle$$

The circuit for U_{BF} is made up of 4 C_{not} gates

Single-qubit bit-flip correction

Suppose a logical qubit is in a state

$$a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

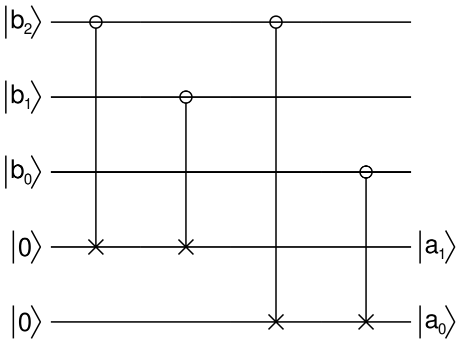
If a bit flip error occurs in b_2 the state is not a codeword and not in C_{BF}

$$\longrightarrow a|100\rangle + b|011\rangle$$

In order to detect the non-codeword states and transform them back to codewords, the transformation U_{BF}, called the syndrome extraction operator is used

$$U_{BF} : |b_2, b_1, b_0, 0, 0\rangle \longrightarrow |b_2, b_1, b_0, b_2 \oplus b_1, b_2 \oplus b_0\rangle$$

The circuit for U_{BF} is made up of 4 C_{not} gates



Single-qubit bit-flip correction

Suppose a logical qubit is in a state

$$a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$
$$\longrightarrow a|100\rangle + b|011\rangle$$

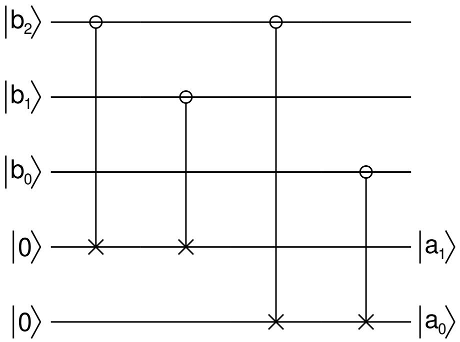
If a bit flip error occurs in b_2 the state is not a codeword and not in C_{BF}

In order to detect the non-codeword states and transform them back to codewords, the transformation U_{BF}, called the syndrome extraction operator is used

$$U_{BF} : |b_2, b_1, b_0, 0, 0\rangle \longrightarrow |b_2, b_1, b_0, b_2 \oplus b_1, b_2 \oplus b_0\rangle$$

The circuit for U_{BF} is made up of 4 C_{not} gates

The ancilla qubits are then measured in the standard basis to determine the correction to be applied



Single-qubit bit-flip correction

Suppose a logical qubit is in a state

$$a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

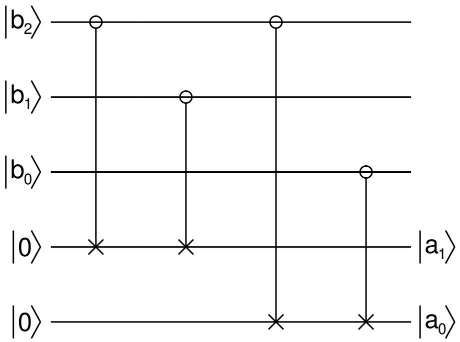
If a bit flip error occurs in b_2 the state is not a codeword and not in C_{BF}

$$\longrightarrow a|100\rangle + b|011\rangle$$

In order to detect the non-codeword states and transform them back to codewords, the transformation U_{BF}, called the syndrome extraction operator is used

$$U_{BF} : |b_2, b_1, b_0, 0, 0\rangle \longrightarrow |b_2, b_1, b_0, b_2 \oplus b_1, b_2 \oplus b_0\rangle$$

The circuit for U_{BF} is made up of 4 C_{not} gates



The ancilla qubits are then measured in the standard basis to determine the correction to be applied

The correction circuit must not only repair any bit-flip errors but not corrupt valid codewords

Single-qubit bit-flip correction

Qubit $|a_1\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_1\rangle$ and qubit $|a_0\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_0\rangle$

Single-qubit bit-flip correction

Qubit $|a_1\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_1\rangle$ and qubit $|a_0\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_0\rangle$

The algorithm for correction is based on majority rule, assuming that only a single qubit is flipped

Single-qubit bit-flip correction

Qubit $|a_1\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_1\rangle$ and qubit $|a_0\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_0\rangle$

The algorithm for correction is based on majority rule, assuming that only a single qubit is flipped

$$\left.\begin{array}{c} 000 \\ 001 \\ 010 \\ 100 \end{array}\right\} \mapsto 0 \ ,$$

Single-qubit bit-flip correction

Qubit $|a_1\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_1\rangle$ and qubit $|a_0\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_0\rangle$

The algorithm for correction is based on majority rule, assuming that only a single qubit is flipped

$$\left.\begin{matrix} 000 \\ 001 \\ 010 \\ 100 \end{matrix}\right\} \mapsto 0\,, \qquad \left.\begin{matrix} 011 \\ 101 \\ 110 \\ 111 \end{matrix}\right\} \mapsto 1$$

Single-qubit bit-flip correction

Qubit $|a_1\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_1\rangle$ and qubit $|a_0\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_0\rangle$

The algorithm for correction is based on majority rule, assuming that only a single qubit is flipped

$$\left.\begin{matrix} 000 \\ 001 \\ 010 \\ 100 \end{matrix}\right\} \mapsto 0 \,, \qquad \left.\begin{matrix} 011 \\ 101 \\ 110 \\ 111 \end{matrix}\right\} \mapsto 1$$

Syndrome	Bit flipped	Error correction
$\|00\rangle$	–	$I \otimes I \otimes I$

Single-qubit bit-flip correction

Qubit $|a_1\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_1\rangle$ and qubit $|a_0\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_0\rangle$

The algorithm for correction is based on majority rule, assuming that only a single qubit is flipped

$$\left.\begin{matrix} 000 \\ 001 \\ 010 \\ 100 \end{matrix}\right\} \mapsto 0\,, \qquad \left.\begin{matrix} 011 \\ 101 \\ 110 \\ 111 \end{matrix}\right\} \mapsto 1$$

Syndrome	Bit flipped	Error correction
$\|00\rangle$	–	$I \otimes I \otimes I$
$\|01\rangle$	b_0	$X_0 = I \otimes I \otimes X$

Single-qubit bit-flip correction

Qubit $|a_1\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_1\rangle$ and qubit $|a_0\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_0\rangle$

The algorithm for correction is based on majority rule, assuming that only a single qubit is flipped

$$\left.\begin{matrix} 000 \\ 001 \\ 010 \\ 100 \end{matrix}\right\} \mapsto 0\ , \qquad \left.\begin{matrix} 011 \\ 101 \\ 110 \\ 111 \end{matrix}\right\} \mapsto 1$$

Syndrome	Bit flipped	Error correction
$\|00\rangle$	–	$I \otimes I \otimes I$
$\|01\rangle$	b_0	$X_0 = I \otimes I \otimes X$
$\|10\rangle$	b_1	$X_1 = I \otimes X \otimes I$

Single-qubit bit-flip correction

Qubit $|a_1\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_1\rangle$ and qubit $|a_0\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_0\rangle$

The algorithm for correction is based on majority rule, assuming that only a single qubit is flipped

$$\left.\begin{array}{c}000\\001\\010\\100\end{array}\right\} \mapsto 0\ , \qquad \left.\begin{array}{c}011\\101\\110\\111\end{array}\right\} \mapsto 1$$

Syndrome	Bit flipped	Error correction
$\|00\rangle$	–	$I \otimes I \otimes I$
$\|01\rangle$	b_0	$X_0 = I \otimes I \otimes X$
$\|10\rangle$	b_1	$X_1 = I \otimes X \otimes I$
$\|11\rangle$	b_2	$X_2 = X \otimes I \otimes I$

Single-qubit bit-flip correction

Qubit $|a_1\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_1\rangle$ and qubit $|a_0\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_0\rangle$

The algorithm for correction is based on majority rule, assuming that only a single qubit is flipped

$$\left.\begin{matrix} 000 \\ 001 \\ 010 \\ 100 \end{matrix}\right\} \mapsto 0 \, , \qquad \left.\begin{matrix} 011 \\ 101 \\ 110 \\ 111 \end{matrix}\right\} \mapsto 1$$

Syndrome	Bit flipped	Error correction
$\|00\rangle$	–	$I \otimes I \otimes I$
$\|01\rangle$	b_0	$X_0 = I \otimes I \otimes X$
$\|10\rangle$	b_1	$X_1 = I \otimes X \otimes I$
$\|11\rangle$	b_2	$X_2 = X \otimes I \otimes I$

Note that this correction works on any superposition of valid codewords

Single-qubit bit-flip correction

Qubit $|a_1\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_1\rangle$ and qubit $|a_0\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_0\rangle$

The algorithm for correction is based on majority rule, assuming that only a single qubit is flipped

$$\left.\begin{array}{c} 000 \\ 001 \\ 010 \\ 100 \end{array}\right\} \mapsto 0\ , \qquad \left.\begin{array}{c} 011 \\ 101 \\ 110 \\ 111 \end{array}\right\} \mapsto 1$$

Syndrome	Bit flipped	Error correction
$\|00\rangle$	–	$I \otimes I \otimes I$
$\|01\rangle$	b_0	$X_0 = I \otimes I \otimes X$
$\|10\rangle$	b_1	$X_1 = I \otimes X \otimes I$
$\|11\rangle$	b_2	$X_2 = X \otimes I \otimes I$

Note that this correction works on any superposition of valid codewords

Furthermore, the syndrome extraction operator does not disturb the quantum state as it merely determines if the system is in a valid codeword state, but not what state it is in

Single-qubit bit-flip correction

Qubit $|a_1\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_1\rangle$ and qubit $|a_0\rangle$ will be $|1\rangle$ if $|b_2\rangle \neq |b_0\rangle$

The algorithm for correction is based on majority rule, assuming that only a single qubit is flipped

$$\left.\begin{matrix} 000 \\ 001 \\ 010 \\ 100 \end{matrix}\right\} \mapsto 0 \,, \qquad \left.\begin{matrix} 011 \\ 101 \\ 110 \\ 111 \end{matrix}\right\} \mapsto 1$$

Syndrome	Bit flipped	Error correction
$\lvert 00\rangle$	–	$I \otimes I \otimes I$
$\lvert 01\rangle$	b_0	$X_0 = I \otimes I \otimes X$
$\lvert 10\rangle$	b_1	$X_1 = I \otimes X \otimes I$
$\lvert 11\rangle$	b_2	$X_2 = X \otimes I \otimes I$

Note that this correction works on any superposition of valid codewords

Furthermore, the syndrome extraction operator does not disturb the quantum state as it merely determines if the system is in a valid codeword state, but not what state it is in

Finally, this error correction will only work for a single qubit error, a longer codeword will permit larger number of errors to be corrected

Example 11.1.1

A general superposition $|\psi\rangle = a|0\rangle + b|1\rangle$ is encoded as

Example 11.1.1

A general superposition $|\psi\rangle = a|0\rangle + b|1\rangle$ is encoded as

$$|\tilde{\psi}\rangle = a|\tilde{0}\rangle + b|\tilde{1}\rangle$$

Example 11.1.1

A general superposition $|\psi\rangle = a|0\rangle + b|1\rangle$ is encoded as

$$|\tilde{\psi}\rangle = a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

Example 11.1.1

A general superposition $|\psi\rangle = a|0\rangle + b|1\rangle$ is encoded as

$$|\tilde{\psi}\rangle = a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If $|\tilde{\psi}\rangle$ is subjected to the single bit-flip error $X_2 = X \otimes I \otimes I$ giving

Example 11.1.1

A general superposition $|\psi\rangle = a|0\rangle + b|1\rangle$ is encoded as

$$|\tilde{\psi}\rangle = a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If $|\tilde{\psi}\rangle$ is subjected to the single bit-flip error $X_2 = X \otimes I \otimes I$ giving

$$X_2|\tilde{\psi}\rangle = a|100\rangle + b|011\rangle$$

Example 11.1.1

A general superposition $|\psi\rangle = a|0\rangle + b|1\rangle$ is encoded as

$$|\tilde{\psi}\rangle = a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If $|\tilde{\psi}\rangle$ is subjected to the single bit-flip error $X_2 = X \otimes I \otimes I$ giving

$$X_2|\tilde{\psi}\rangle = a|100\rangle + b|011\rangle$$

Applying the syndrome extraction operator we have

Example 11.1.1

A general superposition $|\psi\rangle = a|0\rangle + b|1\rangle$ is encoded as

$$|\tilde{\psi}\rangle = a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If $|\tilde{\psi}\rangle$ is subjected to the single bit-flip error $X_2 = X \otimes I \otimes I$ giving

$$X_2|\tilde{\psi}\rangle = a|100\rangle + b|011\rangle$$

Applying the syndrome extraction operator we have

$$U_{BF}\big((X_2|\tilde{\psi}\rangle) \otimes |00\rangle\big)$$

Example 11.1.1

A general superposition $|\psi\rangle = a|0\rangle + b|1\rangle$ is encoded as

$$|\tilde{\psi}\rangle = a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If $|\tilde{\psi}\rangle$ is subjected to the single bit-flip error $X_2 = X \otimes I \otimes I$ giving

$$X_2|\tilde{\psi}\rangle = a|100\rangle + b|011\rangle$$

Applying the syndrome extraction operator we have

$$U_{BF}\big((X_2|\tilde{\psi}\rangle) \otimes |00\rangle\big) = a|100\rangle|11\rangle + b|011\rangle|11\rangle$$

Example 11.1.1

A general superposition $|\psi\rangle = a|0\rangle + b|1\rangle$ is encoded as

$$|\tilde{\psi}\rangle = a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If $|\tilde{\psi}\rangle$ is subjected to the single bit-flip error $X_2 = X \otimes I \otimes I$ giving

$$X_2|\tilde{\psi}\rangle = a|100\rangle + b|011\rangle$$

Applying the syndrome extraction operator we have

$$U_{BF}\big((X_2|\tilde{\psi}\rangle) \otimes |00\rangle\big) = a|100\rangle|11\rangle + b|011\rangle|11\rangle = (a|100\rangle + b|011\rangle)|11\rangle$$

Example 11.1.1

A general superposition $|\psi\rangle = a|0\rangle + b|1\rangle$ is encoded as

$$|\tilde{\psi}\rangle = a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If $|\tilde{\psi}\rangle$ is subjected to the single bit-flip error $X_2 = X \otimes I \otimes I$ giving

$$X_2|\tilde{\psi}\rangle = a|100\rangle + b|011\rangle$$

Applying the syndrome extraction operator we have

$$U_{BF}\big((X_2|\tilde{\psi}\rangle) \otimes |00\rangle\big) = a|100\rangle|11\rangle + b|011\rangle|11\rangle = (a|100\rangle + b|011\rangle)|11\rangle$$

Applying the X_2 transformation to $X_2|\tilde{\psi}\rangle$ will remove the error from the encoded qubit state

Example 11.1.1

A general superposition $|\psi\rangle = a|0\rangle + b|1\rangle$ is encoded as

$$|\tilde{\psi}\rangle = a|\tilde{0}\rangle + b|\tilde{1}\rangle = a|000\rangle + b|111\rangle$$

If $|\tilde{\psi}\rangle$ is subjected to the single bit-flip error $X_2 = X \otimes I \otimes I$ giving

$$X_2|\tilde{\psi}\rangle = a|100\rangle + b|011\rangle$$

Applying the syndrome extraction operator we have

$$U_{BF}\big((X_2|\tilde{\psi}\rangle) \otimes |00\rangle\big) = a|100\rangle|11\rangle + b|011\rangle|11\rangle = (a|100\rangle + b|011\rangle)|11\rangle$$

Applying the X_2 transformation to $X_2|\tilde{\psi}\rangle$ will remove the error from the encoded qubit state

$$X_2X_2|\tilde{\psi}\rangle = |\tilde{\psi}\rangle = a|000\rangle + b|111\rangle$$

Example 11.1.2

The state $|0\rangle$ is encoded as

Example 11.1.2

The state $|0\rangle$ is encoded as $$|\tilde{0}\rangle = |000\rangle$$

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Applying the syndrome extraction operator gives

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Applying the syndrome extraction operator gives

$$U_{BF}\left((E|\tilde{0}\rangle) \otimes |00\rangle\right)$$

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Applying the syndrome extraction operator gives

$$U_{BF}\big((E|\tilde{0}\rangle) \otimes |00\rangle\big) = U_{BF}\big((\alpha|100\rangle + \beta|010\rangle) \otimes |00\rangle\big)$$

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Applying the syndrome extraction operator gives

$$U_{BF}\big((E|\tilde{0}\rangle) \otimes |00\rangle\big) = U_{BF}\big((\alpha|100\rangle + \beta|010\rangle) \otimes |00\rangle\big) = \alpha|100\rangle|11\rangle + \beta|010\rangle|10\rangle$$

Example 11.1.2

The state $|0\rangle$ is encoded as
$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1
$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by
$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Applying the syndrome extraction operator gives

$$U_{BF}\big((E|\tilde{0}\rangle) \otimes |00\rangle\big) = U_{BF}\big((\alpha|100\rangle + \beta|010\rangle) \otimes |00\rangle\big) = \alpha|100\rangle|11\rangle + \beta|010\rangle|10\rangle$$

When the ancilla register is measured, the result is either $|11\rangle$ or $|10\rangle$ and the 3-qubit encoded state is collapsed

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Applying the syndrome extraction operator gives

$$U_{BF}\big((E|\tilde{0}\rangle) \otimes |00\rangle\big) = U_{BF}\big((\alpha|100\rangle + \beta|010\rangle) \otimes |00\rangle\big) = \alpha|100\rangle|11\rangle + \beta|010\rangle|10\rangle$$

When the ancilla register is measured, the result is either $|11\rangle$ or $|10\rangle$ and the 3-qubit encoded state is collapsed

If the ancilla is measured to be $|11\rangle$ the encoded state is now $|100\rangle$ and the correction is

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Applying the syndrome extraction operator gives

$$U_{BF}\big((E|\tilde{0}\rangle) \otimes |00\rangle\big) = U_{BF}\big((\alpha|100\rangle + \beta|010\rangle) \otimes |00\rangle\big) = \alpha|100\rangle|11\rangle + \beta|010\rangle|10\rangle$$

When the ancilla register is measured, the result is either $|11\rangle$ or $|10\rangle$ and the 3-qubit encoded state is collapsed

If the ancilla is measured to be $|11\rangle$ the encoded state is now $|100\rangle$ and the correction is

$$X_2|100\rangle = X \otimes I \otimes I|100\rangle$$

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Applying the syndrome extraction operator gives

$$U_{BF}\big((E|\tilde{0}\rangle) \otimes |00\rangle\big) = U_{BF}\big((\alpha|100\rangle + \beta|010\rangle) \otimes |00\rangle\big) = \alpha|100\rangle|11\rangle + \beta|010\rangle|10\rangle$$

When the ancilla register is measured, the result is either $|11\rangle$ or $|10\rangle$ and the 3-qubit encoded state is collapsed

If the ancilla is measured to be $|11\rangle$ the encoded state is now $|100\rangle$ and the correction is

$$X_2|100\rangle = X \otimes I \otimes I|100\rangle = |000\rangle$$

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Applying the syndrome extraction operator gives

$$U_{BF}\big((E|\tilde{0}\rangle) \otimes |00\rangle\big) = U_{BF}\big((\alpha|100\rangle + \beta|010\rangle) \otimes |00\rangle\big) = \alpha|100\rangle|11\rangle + \beta|010\rangle|10\rangle$$

When the ancilla register is measured, the result is either $|11\rangle$ or $|10\rangle$ and the 3-qubit encoded state is collapsed

If the ancilla is measured to be $|11\rangle$ the encoded state is now $|100\rangle$ and the correction is

$$X_2|100\rangle = X \otimes I \otimes I|100\rangle = |000\rangle$$

If the ancilla is measured to be $|10\rangle$ the encoded state is now $|010\rangle$ and the correction is

Example 11.1.2

The state $|0\rangle$ is encoded as $|\tilde{0}\rangle = |000\rangle$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Applying the syndrome extraction operator gives

$$U_{BF}\big((E|\tilde{0}\rangle) \otimes |00\rangle\big) = U_{BF}\big((\alpha|100\rangle + \beta|010\rangle) \otimes |00\rangle\big) = \alpha|100\rangle|11\rangle + \beta|010\rangle|10\rangle$$

When the ancilla register is measured, the result is either $|11\rangle$ or $|10\rangle$ and the 3-qubit encoded state is collapsed

If the ancilla is measured to be $|11\rangle$ the encoded state is now $|100\rangle$ and the correction is

$$X_2|100\rangle = X \otimes I \otimes I|100\rangle = |000\rangle$$

If the ancilla is measured to be $|10\rangle$ the encoded state is now $|010\rangle$ and the correction is

$$X_1|100\rangle = I \otimes X \otimes I|010\rangle$$

Example 11.1.2

The state $|0\rangle$ is encoded as

$$|\tilde{0}\rangle = |000\rangle$$

This state is subjected to linear combination of two single bit-flip errors, X_2 and X_1

$$E = \alpha X \otimes I \otimes I + \beta I \otimes X \otimes I$$

The resulting state is given by

$$E|\tilde{0}\rangle = \alpha|100\rangle + \beta|010\rangle$$

Applying the syndrome extraction operator gives

$$U_{BF}\big((E|\tilde{0}\rangle) \otimes |00\rangle\big) = U_{BF}\big((\alpha|100\rangle + \beta|010\rangle) \otimes |00\rangle\big) = \alpha|100\rangle|11\rangle + \beta|010\rangle|10\rangle$$

When the ancilla register is measured, the result is either $|11\rangle$ or $|10\rangle$ and the 3-qubit encoded state is collapsed

If the ancilla is measured to be $|11\rangle$ the encoded state is now $|100\rangle$ and the correction is

$$X_2|100\rangle = X \otimes I \otimes I|100\rangle = |000\rangle$$

If the ancilla is measured to be $|10\rangle$ the encoded state is now $|010\rangle$ and the correction is

$$X_1|100\rangle = I \otimes X \otimes I|010\rangle = |000\rangle$$

Example 11.1.3

The quantum state $|+\rangle$ is encoded as

Example 11.1.3

The quantum state $|+\rangle$ is encoded as

$$|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Example 11.1.3

The quantum state $|+\rangle$ is encoded as

$$|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Suppose that the state is subjected by an phase error

Example 11.1.3

The quantum state $|+\rangle$ is encoded as

$$|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Suppose that the state is subjected by an phase error

$$E = Z \otimes I \otimes I$$

Example 11.1.3

The quantum state $|+\rangle$ is encoded as

$$|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Suppose that the state is subjected by an phase error

$$E = Z \otimes I \otimes I$$

The initial $|\tilde{+}\rangle$ state becomes

Example 11.1.3

The quantum state $|+\rangle$ is encoded as

$$|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Suppose that the state is subjected by an phase error

$$E = Z \otimes I \otimes I$$

The initial $|\tilde{+}\rangle$ state becomes

$$E|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

Example 11.1.3

The quantum state $|+\rangle$ is encoded as

$$|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Suppose that the state is subjected by an phase error

$$E = Z \otimes I \otimes I$$

The initial $|\tilde{+}\rangle$ state becomes

$$E|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

The syndrome extraction operator is now applied to the corrupted state

Example 11.1.3

The quantum state $|+\rangle$ is encoded as

$$|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Suppose that the state is subjected by an phase error

$$E = Z \otimes I \otimes I$$

The initial $|\tilde{+}\rangle$ state becomes

$$E|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

The syndrome extraction operator is now applied to the corrupted state

$$U_{BF}\big((E|\tilde{+}\rangle) \otimes |00\rangle\big)$$

Example 11.1.3

The quantum state $|+\rangle$ is encoded as

$$|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Suppose that the state is subjected by an phase error

$$E = Z \otimes I \otimes I$$

The initial $|\tilde{+}\rangle$ state becomes

$$E|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

The syndrome extraction operator is now applied to the corrupted state

$$U_{BF}\big((E|\tilde{+}\rangle) \otimes |00\rangle\big) = U_{BF}\frac{1}{\sqrt{2}}\big((|000\rangle - |111\rangle) \otimes |00\rangle\big)$$

Example 11.1.3

The quantum state $|+\rangle$ is encoded as

$$|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Suppose that the state is subjected by an phase error

$$E = Z \otimes I \otimes I$$

The initial $|\tilde{+}\rangle$ state becomes

$$E|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

The syndrome extraction operator is now applied to the corrupted state

$$U_{BF}\big((E|\tilde{+}\rangle) \otimes |00\rangle\big) = U_{BF}\frac{1}{\sqrt{2}}\big((|000\rangle - |111\rangle) \otimes |00\rangle\big) = \frac{1}{\sqrt{2}}(|000\rangle|00\rangle - |111\rangle|00\rangle)$$

Example 11.1.3

The quantum state $|+\rangle$ is encoded as

$$|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$$

Suppose that the state is subjected by an phase error

$$E = Z \otimes I \otimes I$$

The initial $|\tilde{+}\rangle$ state becomes

$$E|\tilde{+}\rangle = \frac{1}{\sqrt{2}}(|000\rangle - |111\rangle)$$

The syndrome extraction operator is now applied to the corrupted state

$$U_{BF}\big((E|\tilde{+}\rangle) \otimes |00\rangle\big) = U_{BF}\frac{1}{\sqrt{2}}\big((|000\rangle - |111\rangle) \otimes |00\rangle\big) = \frac{1}{\sqrt{2}}(|000\rangle|00\rangle - |111\rangle|00\rangle)$$

Since the ancilla register is $|00\rangle$ for both terms no error is detected with this scheme

Single-qubit phase-flip correction

The three possible single-qubit phase-flip errors are

Single-qubit phase-flip correction

The three possible single-qubit phase-flip errors are

$$\{Z_2 = Z \otimes I \otimes I, Z_1 = I \otimes Z \otimes I, Z_0 = I \otimes I \otimes Z\}$$

Single-qubit phase-flip correction

The three possible single-qubit phase-flip errors are

$$\{Z_2 = Z \otimes I \otimes I, Z_1 = I \otimes Z \otimes I, Z_0 = I \otimes I \otimes Z\}$$

A phase-flip error in the standard basis is simply a bit-flip error in the Hadamard basis, $\{|+\rangle, |-\rangle\}$ since $Z = HXH$

Single-qubit phase-flip correction

The three possible single-qubit phase-flip errors are

$$\{Z_2 = Z \otimes I \otimes I, Z_1 = I \otimes Z \otimes I, Z_0 = I \otimes I \otimes Z\}$$

A phase-flip error in the standard basis is simply a bit-flip error in the Hadamard basis, $\{|+\rangle, |-\rangle\}$ since $Z = HXH$

This suggests that a code C_{PF} with elements $\{|+++\rangle, |---\rangle\}$ for phase-flip correction can be generated by applying the Walsh-Hadamard transformation $W^{(3)} = H \otimes H \otimes H$ to the C_{BF} code to be able to detect phase-flip errors

Single-qubit phase-flip correction

The three possible single-qubit phase-flip errors are

$$\{Z_2 = Z \otimes I \otimes I, Z_1 = I \otimes Z \otimes I, Z_0 = I \otimes I \otimes Z\}$$

A phase-flip error in the standard basis is simply a bit-flip error in the Hadamard basis, $\{|+\rangle, |-\rangle\}$ since $Z = HXH$

This suggests that a code C_{PF} with elements $\{|+++\rangle, |---\rangle\}$ for phase-flip correction can be generated by applying the Walsh-Hadamard transformation $W^{(3)} = H \otimes H \otimes H$ to the C_{BF} code to be able to detect phase-flip errors

Once in the Hadamard basis, the usual syndrome extraction operator can be used to obtain the ancilla code

Single-qubit phase-flip correction

The three possible single-qubit phase-flip errors are

$$\{Z_2 = Z \otimes I \otimes I, Z_1 = I \otimes Z \otimes I, Z_0 = I \otimes I \otimes Z\}$$

A phase-flip error in the standard basis is simply a bit-flip error in the Hadamard basis, $\{|+\rangle, |-\rangle\}$ since $Z = HXH$

This suggests that a code C_{PF} with elements $\{| + ++\rangle, | - --\rangle\}$ for phase-flip correction can be generated by applying the Walsh-Hadamard transformation $W^{(3)} = H \otimes H \otimes H$ to the C_{BF} code to be able to detect phase-flip errors

Once in the Hadamard basis, the usual syndrome extraction operator can be used to obtain the ancilla code

The correction is applied after returning to the standard basis with a second W transformation

Single-qubit phase-flip correction

The three possible single-qubit phase-flip errors are

$$\{Z_2 = Z \otimes I \otimes I, Z_1 = I \otimes Z \otimes I, Z_0 = I \otimes I \otimes Z\}$$

A phase-flip error in the standard basis is simply a bit-flip error in the Hadamard basis, $\{|+\rangle, |-\rangle\}$ since $Z = HXH$

This suggests that a code C_{PF} with elements $\{|+++\rangle, |---\rangle\}$ for phase-flip correction can be generated by applying the Walsh-Hadamard transformation $W^{(3)} = H \otimes H \otimes H$ to the C_{BF} code to be able to detect phase-flip errors

Once in the Hadamard basis, the usual syndrome extraction operator can be used to obtain the ancilla code

The correction is applied after returning to the standard basis with a second W transformation

Syndrome	Bit shifted	Error correction
$\|00\rangle$	–	$I \otimes I \otimes I$

Single-qubit phase-flip correction

The three possible single-qubit phase-flip errors are

$$\{Z_2 = Z \otimes I \otimes I, Z_1 = I \otimes Z \otimes I, Z_0 = I \otimes I \otimes Z\}$$

A phase-flip error in the standard basis is simply a bit-flip error in the Hadamard basis, $\{|+\rangle, |-\rangle\}$ since $Z = HXH$

This suggests that a code C_{PF} with elements $\{|+++\rangle, |---\rangle\}$ for phase-flip correction can be generated by applying the Walsh-Hadamard transformation $W^{(3)} = H \otimes H \otimes H$ to the C_{BF} code to be able to detect phase-flip errors

Once in the Hadamard basis, the usual syndrome extraction operator can be used to obtain the ancilla code

The correction is applied after returning to the standard basis with a second W transformation

Syndrome	Bit shifted	Error correction
$\|00\rangle$	–	$I \otimes I \otimes I$
$\|01\rangle$	b_0	$Z_0 = I \otimes I \otimes Z$

Single-qubit phase-flip correction

The three possible single-qubit phase-flip errors are

$$\{Z_2 = Z \otimes I \otimes I, Z_1 = I \otimes Z \otimes I, Z_0 = I \otimes I \otimes Z\}$$

A phase-flip error in the standard basis is simply a bit-flip error in the Hadamard basis, $\{|+\rangle, |-\rangle\}$ since $Z = HXH$

This suggests that a code C_{PF} with elements $\{|+++\rangle, |---\rangle\}$ for phase-flip correction can be generated by applying the Walsh-Hadamard transformation $W^{(3)} = H \otimes H \otimes H$ to the C_{BF} code to be able to detect phase-flip errors

Once in the Hadamard basis, the usual syndrome extraction operator can be used to obtain the ancilla code

The correction is applied after returning to the standard basis with a second W transformation

Syndrome	Bit shifted	Error correction
$\|00\rangle$	–	$I \otimes I \otimes I$
$\|01\rangle$	b_0	$Z_0 = I \otimes I \otimes Z$
$\|10\rangle$	b_1	$Z_1 = I \otimes Z \otimes I$

Single-qubit phase-flip correction

The three possible single-qubit phase-flip errors are

$$\{Z_2 = Z \otimes I \otimes I, Z_1 = I \otimes Z \otimes I, Z_0 = I \otimes I \otimes Z\}$$

A phase-flip error in the standard basis is simply a bit-flip error in the Hadamard basis, $\{|+\rangle, |-\rangle\}$ since $Z = HXH$

This suggests that a code C_{PF} with elements $\{|+++\rangle, |---\rangle\}$ for phase-flip correction can be generated by applying the Walsh-Hadamard transformation $W^{(3)} = H \otimes H \otimes H$ to the C_{BF} code to be able to detect phase-flip errors

Once in the Hadamard basis, the usual syndrome extraction operator can be used to obtain the ancilla code

The correction is applied after returning to the standard basis with a second W transformation

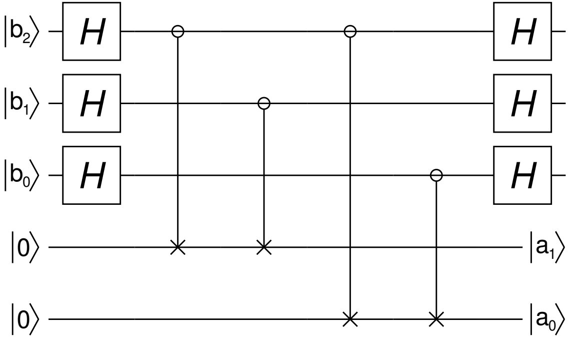
Syndrome	Bit shifted	Error correction
$\|00\rangle$	–	$I \otimes I \otimes I$
$\|01\rangle$	b_0	$Z_0 = I \otimes I \otimes Z$
$\|10\rangle$	b_1	$Z_1 = I \otimes Z \otimes I$
$\|11\rangle$	b_2	$Z_2 = Z \otimes I \otimes I$

Single-qubit phase-flip correction

The syndrome extraction operator for the phase-flip error is thus $U_{PF} = W U_{BF} W$ and has a circuit diagram

Single-qubit phase-flip correction

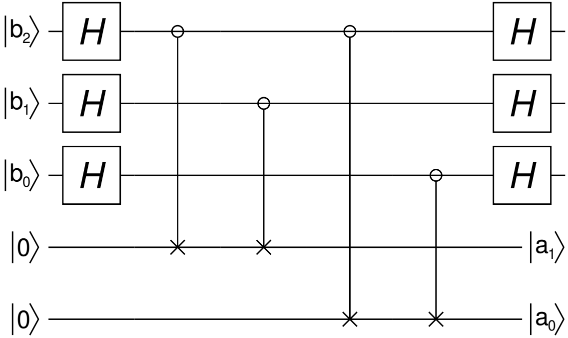
The syndrome extraction operator for the phase-flip error is thus $U_{PF} = WU_{BF}W$ and has a circuit diagram



Single-qubit phase-flip correction

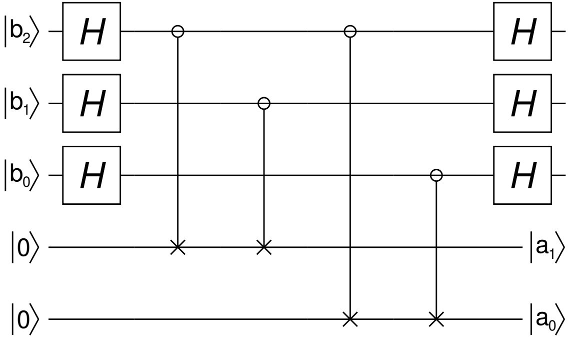
The syndrome extraction operator for the phase-flip error is thus $U_{PF} = WU_{BF}W$ and has a circuit diagram

Note that the C_{PF} code corrects all single-qubit relative phase errors, not just the phase-flip error, because any single qubit phase error is a linear combination of Z and I up to an irrelevant global phase factor



Single-qubit phase-flip correction

The syndrome extraction operator for the phase-flip error is thus $U_{PF} = WU_{BF}W$ and has a circuit diagram

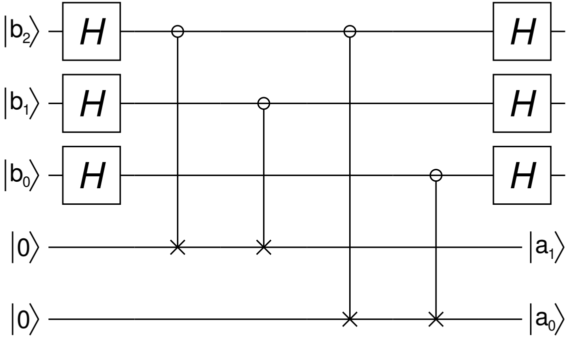


Note that the C_{PF} code corrects all single-qubit relative phase errors, not just the phase-flip error, because any single qubit phase error is a linear combination of Z and I up to an irrelevant global phase factor

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}$$

Single-qubit phase-flip correction

The syndrome extraction operator for the phase-flip error is thus $U_{PF} = W U_{BF} W$ and has a circuit diagram

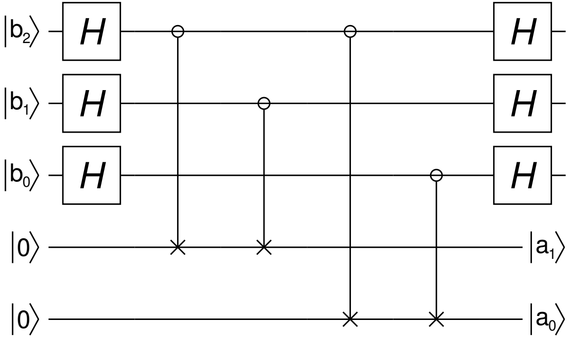


Note that the C_{PF} code corrects all single-qubit relative phase errors, not just the phase-flip error, because any single qubit phase error is a linear combination of Z and I up to an irrelevant global phase factor

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} = e^{i\frac{\phi}{2}} \begin{pmatrix} e^{-i\frac{\phi}{2}} & 0 \\ 0 & e^{+i\frac{\phi}{2}} \end{pmatrix}$$

Single-qubit phase-flip correction

The syndrome extraction operator for the phase-flip error is thus $U_{PF} = W U_{BF} W$ and has a circuit diagram

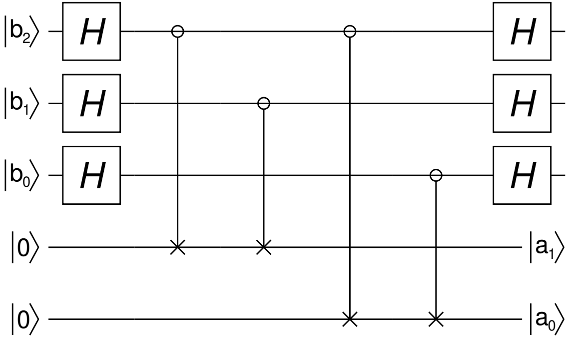


Note that the C_{PF} code corrects all single-qubit relative phase errors, not just the phase-flip error, because any single qubit phase error is a linear combination of Z and I up to an irrelevant global phase factor

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} = e^{i\frac{\phi}{2}} \begin{pmatrix} e^{-i\frac{\phi}{2}} & 0 \\ 0 & e^{+i\frac{\phi}{2}} \end{pmatrix} = e^{i\frac{\phi}{2}} \begin{pmatrix} \cos\frac{\phi}{2} - i\sin\frac{\phi}{2} & 0 \\ 0 & \cos\frac{\phi}{2} + i\sin\frac{\phi}{2} \end{pmatrix}$$

Single-qubit phase-flip correction

The syndrome extraction operator for the phase-flip error is thus $U_{PF} = WU_{BF}W$ and has a circuit diagram



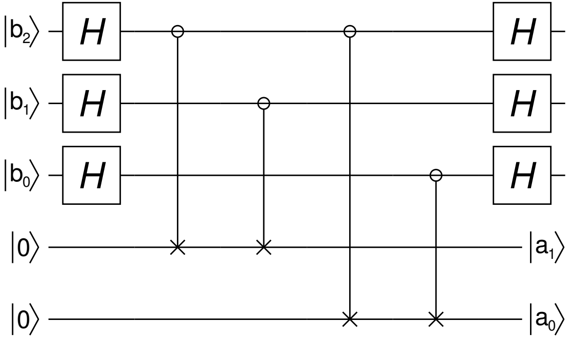
Note that the C_{PF} code corrects all single-qubit relative phase errors, not just the phase-flip error, because any single qubit phase error is a linear combination of Z and I up to an irrelevant global phase factor

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} = e^{i\frac{\phi}{2}} \begin{pmatrix} e^{-i\frac{\phi}{2}} & 0 \\ 0 & e^{+i\frac{\phi}{2}} \end{pmatrix} = e^{i\frac{\phi}{2}} \begin{pmatrix} \cos\frac{\phi}{2} - i\sin\frac{\phi}{2} & 0 \\ 0 & \cos\frac{\phi}{2} + i\sin\frac{\phi}{2} \end{pmatrix}$$

$$= e^{i\frac{\phi}{2}} \left(\cos\frac{\phi}{2} I - i\sin\frac{\phi}{2} Z \right)$$

Single-qubit phase-flip correction

The syndrome extraction operator for the phase-flip error is thus $U_{PF} = WU_{BF}W$ and has a circuit diagram



Note that the C_{PF} code corrects all single-qubit relative phase errors, not just the phase-flip error, because any single qubit phase error is a linear combination of Z and I up to an irrelevant global phase factor

$$\begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix} = e^{i\frac{\phi}{2}} \begin{pmatrix} e^{-i\frac{\phi}{2}} & 0 \\ 0 & e^{+i\frac{\phi}{2}} \end{pmatrix} = e^{i\frac{\phi}{2}} \begin{pmatrix} \cos\frac{\phi}{2} - i\sin\frac{\phi}{2} & 0 \\ 0 & \cos\frac{\phi}{2} + i\sin\frac{\phi}{2} \end{pmatrix}$$

$$= e^{i\frac{\phi}{2}} \left(\cos\frac{\phi}{2} I - i\sin\frac{\phi}{2} Z \right) \mapsto \cos\frac{\phi}{2} I - i\sin\frac{\phi}{2} Z$$

Complete single-qubit error correction

As will be shown later, a quantum code that can correct X and Z single-qubit errors can be used to correct any single-qubit error

Complete single-qubit error correction

As will be shown later, a quantum code that can correct X and Z single-qubit errors can be used to correct any single-qubit error

By combining the C_{BF} and C_{PF} codes it is possible to catch both types of errors at the cost of additional qubits

Complete single-qubit error correction

As will be shown later, a quantum code that can correct X and Z single-qubit errors can be used to correct any single-qubit error

By combining the C_{BF} and C_{PF} codes it is possible to catch both types of errors at the cost of additional qubits

The algorithm is to first encode using C_{PF} and then with C_{BF} using 9-qubits per initial qubit

Complete single-qubit error correction

As will be shown later, a quantum code that can correct X and Z single-qubit errors can be used to correct any single-qubit error

By combining the C_{BF} and C_{PF} codes it is possible to catch both types of errors at the cost of additional qubits

The algorithm is to first encode using C_{PF} and then with C_{BF} using 9-qubits per initial qubit

$$|0\rangle \to |\tilde{0}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big)$$

Complete single-qubit error correction

As will be shown later, a quantum code that can correct X and Z single-qubit errors can be used to correct any single-qubit error

By combining the C_{BF} and C_{PF} codes it is possible to catch both types of errors at the cost of additional qubits

The algorithm is to first encode using C_{PF} and then with C_{BF} using 9-qubits per initial qubit

$$|0\rangle \to |\tilde{0}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big)^{\otimes 3}$$

Complete single-qubit error correction

As will be shown later, a quantum code that can correct X and Z single-qubit errors can be used to correct any single-qubit error

By combining the C_{BF} and C_{PF} codes it is possible to catch both types of errors at the cost of additional qubits

The algorithm is to first encode using C_{PF} and then with C_{BF} using 9-qubits per initial qubit

$$|0\rangle \to |\tilde{0}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big)^{\otimes 3}$$

$$|1\rangle \to |\tilde{1}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle - |111\rangle\big) \otimes \big(|000\rangle - |111\rangle\big) \otimes \big(|000\rangle - |111\rangle\big)$$

Complete single-qubit error correction

As will be shown later, a quantum code that can correct X and Z single-qubit errors can be used to correct any single-qubit error

By combining the C_{BF} and C_{PF} codes it is possible to catch both types of errors at the cost of additional qubits

The algorithm is to first encode using C_{PF} and then with C_{BF} using 9-qubits per initial qubit

$$|0\rangle \rightarrow |\tilde{0}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big)^{\otimes 3}$$

$$|1\rangle \rightarrow |\tilde{1}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle - |111\rangle\big) \otimes \big(|000\rangle - |111\rangle\big) \otimes \big(|000\rangle - |111\rangle\big) = \frac{1}{\sqrt{8}}\big(|000\rangle - |111\rangle\big)^{\otimes 3}$$

Complete single-qubit error correction

As will be shown later, a quantum code that can correct X and Z single-qubit errors can be used to correct any single-qubit error

By combining the C_{BF} and C_{PF} codes it is possible to catch both types of errors at the cost of additional qubits

The algorithm is to first encode using C_{PF} and then with C_{BF} using 9-qubits per initial qubit

$$|0\rangle \to |\tilde{0}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big)^{\otimes 3}$$

$$|1\rangle \to |\tilde{1}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle - |111\rangle\big) \otimes \big(|000\rangle - |111\rangle\big) \otimes \big(|000\rangle - |111\rangle\big) = \frac{1}{\sqrt{8}}\big(|000\rangle - |111\rangle\big)^{\otimes 3}$$

Error correction is performed on each block of 3 qubits to correct for X errors

Complete single-qubit error correction

As will be shown later, a quantum code that can correct X and Z single-qubit errors can be used to correct any single-qubit error

By combining the C_{BF} and C_{PF} codes it is possible to catch both types of errors at the cost of additional qubits

The algorithm is to first encode using C_{PF} and then with C_{BF} using 9-qubits per initial qubit

$$|0\rangle \rightarrow |\tilde{0}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big)^{\otimes 3}$$

$$|1\rangle \rightarrow |\tilde{1}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle - |111\rangle\big) \otimes \big(|000\rangle - |111\rangle\big) \otimes \big(|000\rangle - |111\rangle\big) = \frac{1}{\sqrt{8}}\big(|000\rangle - |111\rangle\big)^{\otimes 3}$$

Error correction is performed on each block of 3 qubits to correct for X errors

Then phase shifts are corrected using a variant of U_{PF} applied to the three blocks as a codeword

Complete single-qubit error correction

As will be shown later, a quantum code that can correct X and Z single-qubit errors can be used to correct any single-qubit error

By combining the C_{BF} and C_{PF} codes it is possible to catch both types of errors at the cost of additional qubits

The algorithm is to first encode using C_{PF} and then with C_{BF} using 9-qubits per initial qubit

$$|0\rangle \to |\tilde{0}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) \otimes \big(|000\rangle + |111\rangle\big) = \frac{1}{\sqrt{8}}\big(|000\rangle + |111\rangle\big)^{\otimes 3}$$

$$|1\rangle \to |\tilde{1}\rangle = \frac{1}{\sqrt{8}}\big(|000\rangle - |111\rangle\big) \otimes \big(|000\rangle - |111\rangle\big) \otimes \big(|000\rangle - |111\rangle\big) = \frac{1}{\sqrt{8}}\big(|000\rangle - |111\rangle\big)^{\otimes 3}$$

Error correction is performed on each block of 3 qubits to correct for X errors

Then phase shifts are corrected using a variant of U_{PF} applied to the three blocks as a codeword

In this implementation each logical qubit is made up of 9 physical qubits and gates are implemented to act on the logical qubits

Error correcting framework

The general theory behind error correcting codes comes from classical error correction

Error correcting framework

The general theory behind error correcting codes comes from classical error correction

A $[n, k]$ block code C is a 2^k subset of the 2^n possible n-bit strings which form a group, $\mathbf{Z}_2^n$ under bitwise addition modulo 2

Error correcting framework

The general theory behind error correcting codes comes from classical error correction

A $[n, k]$ block code C is a 2^k subset of the 2^n possible n-bit strings which form a group, $\mathbf{Z}_2^n$ under bitwise addition modulo 2

The code is linear when the 2^k subset C is a subgroup of $\mathbf{Z}_2^n$

Error correcting framework

The general theory behind error correcting codes comes from classical error correction

A $[n, k]$ block code C is a 2^k subset of the 2^n possible n-bit strings which form a group, $\mathbf{Z}_2^n$ under bitwise addition modulo 2

The code is linear when the 2^k subset C is a subgroup of $\mathbf{Z}_2^n$

A specific (but non-unique) encoding function $c : \mathbf{Z}_2^k \rightarrow \mathbf{Z}_2^n$ is chosen to take the message space (all k-bit strings) into the n-bit code space C

Error correcting framework

The general theory behind error correcting codes comes from classical error correction

A $[n, k]$ block code C is a 2^k subset of the 2^n possible n-bit strings which form a group, $\mathbf{Z}_2^n$ under bitwise addition modulo 2

The code is linear when the 2^k subset C is a subgroup of $\mathbf{Z}_2^n$

A specific (but non-unique) encoding function $c : \mathbf{Z}_2^k \to \mathbf{Z}_2^n$ is chosen to take the message space (all k-bit strings) into the n-bit code space C

An $m \cdot k$ length message is encoded by separately encoding each of the m k-bit blocks with c to give an $m \cdot n$ length ciphertext

Error correcting framework

The general theory behind error correcting codes comes from classical error correction

A $[n, k]$ block code C is a 2^k subset of the 2^n possible n-bit strings which form a group, $\mathbf{Z}_2^n$ under bitwise addition modulo 2

The code is linear when the 2^k subset C is a subgroup of $\mathbf{Z}_2^n$

A specific (but non-unique) encoding function $c : \mathbf{Z}_2^k \to \mathbf{Z}_2^n$ is chosen to take the message space (all k-bit strings) into the n-bit code space C

An $m \cdot k$ length message is encoded by separately encoding each of the m k-bit blocks with c to give an $m \cdot n$ length ciphertext

The encoding function, c can be represented by an $n \times k$ generator matrix, G whose k columns form a linearly independent set of binary words

Error correcting framework

The general theory behind error correcting codes comes from classical error correction

A $[n, k]$ block code C is a 2^k subset of the 2^n possible n-bit strings which form a group, $\mathbf{Z}_2^n$ under bitwise addition modulo 2

The code is linear when the 2^k subset C is a subgroup of $\mathbf{Z}_2^n$

A specific (but non-unique) encoding function $c : \mathbf{Z}_2^k \to \mathbf{Z}_2^n$ is chosen to take the message space (all k-bit strings) into the n-bit code space C

An $m \cdot k$ length message is encoded by separately encoding each of the m k-bit blocks with c to give an $m \cdot n$ length ciphertext

The encoding function, c can be represented by an $n \times k$ generator matrix, G whose k columns form a linearly independent set of binary words

For example for the $[3, 1]$ repetition code we have already seen

Error correcting framework

The general theory behind error correcting codes comes from classical error correction

A $[n, k]$ block code C is a 2^k subset of the 2^n possible n-bit strings which form a group, $\mathbf{Z}_2^n$ under bitwise addition modulo 2

The code is linear when the 2^k subset C is a subgroup of $\mathbf{Z}_2^n$

A specific (but non-unique) encoding function $c : \mathbf{Z}_2^k \to \mathbf{Z}_2^n$ is chosen to take the message space (all k-bit strings) into the n-bit code space C

An $m \cdot k$ length message is encoded by separately encoding each of the m k-bit blocks with c to give an $m \cdot n$ length ciphertext

The encoding function, c can be represented by an $n \times k$ generator matrix, G whose k columns form a linearly independent set of binary words

For example for the $[3, 1]$ repetition code we have already seen

$$G = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix},$$

Error correcting framework

The general theory behind error correcting codes comes from classical error correction

A $[n, k]$ block code C is a 2^k subset of the 2^n possible n-bit strings which form a group, $\mathbf{Z}_2^n$ under bitwise addition modulo 2

The code is linear when the 2^k subset C is a subgroup of $\mathbf{Z}_2^n$

A specific (but non-unique) encoding function $c : \mathbf{Z}_2^k \rightarrow \mathbf{Z}_2^n$ is chosen to take the message space (all k-bit strings) into the n-bit code space C

An $m \cdot k$ length message is encoded by separately encoding each of the m k-bit blocks with c to give an $m \cdot n$ length ciphertext

The encoding function, c can be represented by an $n \times k$ generator matrix, G whose k columns form a linearly independent set of binary words

For example for the $[3, 1]$ repetition code we have already seen

$$G = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} (0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

Error correcting framework

The general theory behind error correcting codes comes from classical error correction

A $[n, k]$ block code C is a 2^k subset of the 2^n possible n-bit strings which form a group, $\mathbf{Z}_2^n$ under bitwise addition modulo 2

The code is linear when the 2^k subset C is a subgroup of $\mathbf{Z}_2^n$

A specific (but non-unique) encoding function $c : \mathbf{Z}_2^k \to \mathbf{Z}_2^n$ is chosen to take the message space (all k-bit strings) into the n-bit code space C

An $m \cdot k$ length message is encoded by separately encoding each of the m k-bit blocks with c to give an $m \cdot n$ length ciphertext

The encoding function, c can be represented by an $n \times k$ generator matrix, G whose k columns form a linearly independent set of binary words

For example for the $[3, 1]$ repetition code we have already seen

$$G = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}(1) = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Example 11.2.2

The $[7, 4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

Example 11.2.2

The $[7,4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

One possible encoding is

Example 11.2.2

The $[7, 4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

One possible encoding is

$$1000 \mapsto 1110100$$

Example 11.2.2

The $[7,4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

One possible encoding is

$$1000 \mapsto 1110100$$

$$0100 \mapsto 1101010$$

Example 11.2.2

The $[7,4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

One possible encoding is

$$1000 \mapsto 1110100$$

$$0100 \mapsto 1101010$$

$$0010 \mapsto 1011001$$

Example 11.2.2

The $[7, 4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

One possible encoding is

$$1000 \mapsto 1110100$$

$$0100 \mapsto 1101010$$

$$0010 \mapsto 1011001$$

$$0001 \mapsto 1111111$$

Example 11.2.2

The $[7, 4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

One possible encoding is with generating matrix

$$1000 \mapsto 1110100$$

$$0100 \mapsto 1101010$$

$$0010 \mapsto 1011001$$

$$0001 \mapsto 1111111$$

Example 11.2.2

The $[7, 4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

One possible encoding is with generating matrix

$$1000 \mapsto 1110100$$

$$0100 \mapsto 1101010$$

$$0010 \mapsto 1011001$$

$$0001 \mapsto 1111111$$

$$G = \left(\quad \right)$$

Example 11.2.2

The $[7, 4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

One possible encoding is with generating matrix

$$1000 \mapsto 1110100$$

$$0100 \mapsto 1101010$$

$$0010 \mapsto 1011001$$

$$0001 \mapsto 1111111$$

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

Example 11.2.2

The $[7, 4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

One possible encoding is with generating matrix

$$1000 \mapsto 1110100$$

$$0100 \mapsto 1101010$$

$$0010 \mapsto 1011001$$

$$0001 \mapsto 1111111$$

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

Example 11.2.2

The $[7,4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

One possible encoding is with generating matrix

$$1000 \mapsto 1110100$$

$$0100 \mapsto 1101010$$

$$0010 \mapsto 1011001$$

$$0001 \mapsto 1111111$$

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

Example 11.2.2

The $[7, 4]$ Hamming code encodes 4-bit strings, elements of $\mathbf{Z}_2^4$, into 7-bit strings, elements of $\mathbf{Z}_2^7$

One possible encoding is with generating matrix

$$1000 \mapsto 1110100$$

$$0100 \mapsto 1101010$$

$$0010 \mapsto 1011001$$

$$0001 \mapsto 1111111$$

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

Example 11.2.2

An alternative encoding could be

Example 11.2.2

An alternative encoding could be

$$G = \begin{pmatrix} \\ \\ \\ \end{pmatrix}$$

Example 11.2.2

An alternative encoding could be

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

Example 11.2.2

An alternative encoding could be

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

Example 11.2.2

An alternative encoding could be

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

Example 11.2.2

An alternative encoding could be

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

Example 11.2.2

An alternative encoding could be

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

$$1000 \mapsto 1000111$$

The single "1" bit values are mapped to the values of the individual columns

Example 11.2.2

An alternative encoding could be

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

$$0100 \mapsto 0100110$$

$$1000 \mapsto 1000111$$

The single "1" bit values are mapped to the values of the individual columns

Example 11.2.2

An alternative encoding could be

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

$$0010 \mapsto 0010101$$

$$0100 \mapsto 0100110$$

$$1000 \mapsto 1000111$$

The single "1" bit values are mapped to the values of the individual columns

Example 11.2.2

An alternative encoding could be

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

$$0001 \mapsto 0001011$$

$$0010 \mapsto 0010101$$

$$0100 \mapsto 0100110$$

$$1000 \mapsto 1000111$$

The single "1" bit values are mapped to the values of the individual columns

Example 11.2.2

An alternative encoding could be

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

$$0001 \mapsto 0001011$$
$$0010 \mapsto 0010101$$
$$0011 \mapsto 0011110$$
$$0100 \mapsto 0100110$$

$$1000 \mapsto 1000111$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

Example 11.2.2

An alternative encoding could be

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

$$0001 \mapsto 0001011$$
$$0010 \mapsto 0010101$$
$$0011 \mapsto 0011110$$
$$0100 \mapsto 0100110$$
$$0101 \mapsto 0101100$$

$$1000 \mapsto 1000111$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

Example 11.2.2

An alternative encoding could be

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

$$0001 \mapsto 0001011$$
$$0010 \mapsto 0010101$$
$$0011 \mapsto 0011110$$
$$0100 \mapsto 0100110$$
$$0101 \mapsto 0101100$$
$$0110 \mapsto 0110011$$

$$1000 \mapsto 1000111$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

Example 11.2.2

An alternative encoding could be

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

$$\begin{aligned} 0001 &\mapsto 0001011 \\ 0010 &\mapsto 0010101 \\ 0011 &\mapsto 0011110 \\ 0100 &\mapsto 0100110 \\ 0101 &\mapsto 0101100 \\ 0110 &\mapsto 0110011 \\ 0111 &\mapsto 0111001 \\ 1000 &\mapsto 1000111 \end{aligned}$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

Example 11.2.2

An alternative encoding could be

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

$$0001 \mapsto 0001011$$
$$0010 \mapsto 0010101$$
$$0011 \mapsto 0011110$$
$$0100 \mapsto 0100110$$
$$0101 \mapsto 0101100$$
$$0110 \mapsto 0110011$$
$$0111 \mapsto 0111001$$
$$1000 \mapsto 1000111$$
$$1001 \mapsto 1001100$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

Example 11.2.2

An alternative encoding could be

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

$$0001 \mapsto 0001011$$
$$0010 \mapsto 0010101$$
$$0011 \mapsto 0011110$$
$$0100 \mapsto 0100110$$
$$0101 \mapsto 0101100$$
$$0110 \mapsto 0110011$$
$$0111 \mapsto 0111001$$
$$1000 \mapsto 1000111$$
$$1001 \mapsto 1001100$$
$$1010 \mapsto 1010010$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

Example 11.2.2

An alternative encoding could be

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

$$0001 \mapsto 0001011$$
$$0010 \mapsto 0010101$$
$$0011 \mapsto 0011110$$
$$0100 \mapsto 0100110$$
$$0101 \mapsto 0101100$$
$$0110 \mapsto 0110011$$
$$0111 \mapsto 0111001$$
$$1000 \mapsto 1000111$$
$$1001 \mapsto 1001100$$
$$1010 \mapsto 1010010$$
$$1011 \mapsto 1011001$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

Example 11.2.2

An alternative encoding could be

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

$$0001 \mapsto 0001011$$
$$0010 \mapsto 0010101$$
$$0011 \mapsto 0011110$$
$$0100 \mapsto 0100110$$
$$0101 \mapsto 0101100$$
$$0110 \mapsto 0110011$$
$$0111 \mapsto 0111001$$
$$1000 \mapsto 1000111$$
$$1001 \mapsto 1001100$$
$$1010 \mapsto 1010010$$
$$1011 \mapsto 1011001$$
$$1100 \mapsto 1100001$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

Example 11.2.2

An alternative encoding could be

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

$$\begin{aligned} 0001 &\mapsto 0001011 \\ 0010 &\mapsto 0010101 \\ 0011 &\mapsto 0011110 \\ 0100 &\mapsto 0100110 \\ 0101 &\mapsto 0101100 \\ 0110 &\mapsto 0110011 \\ 0111 &\mapsto 0111001 \\ 1000 &\mapsto 1000111 \end{aligned}$$

$$\begin{aligned} 1001 &\mapsto 1001100 \\ 1010 &\mapsto 1010010 \\ 1011 &\mapsto 1011001 \\ 1100 &\mapsto 1100001 \\ 1101 &\mapsto 1101010 \end{aligned}$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

Example 11.2.2

An alternative encoding could be

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

$$\begin{aligned}
0001 &\mapsto 0001011 \\
0010 &\mapsto 0010101 \\
0011 &\mapsto 0011110 \\
0100 &\mapsto 0100110 \\
0101 &\mapsto 0101100 \\
0110 &\mapsto 0110011 \\
0111 &\mapsto 0111001 \\
1000 &\mapsto 1000111
\end{aligned}$$

$$\begin{aligned}
1001 &\mapsto 1001100 \\
1010 &\mapsto 1010010 \\
1011 &\mapsto 1011001 \\
1100 &\mapsto 1100001 \\
1101 &\mapsto 1101010 \\
1110 &\mapsto 1110100
\end{aligned}$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

Example 11.2.2

An alternative encoding could be

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

$$0001 \mapsto 0001011$$
$$0010 \mapsto 0010101$$
$$0011 \mapsto 0011110$$
$$0100 \mapsto 0100110$$
$$0101 \mapsto 0101100$$
$$0110 \mapsto 0110011$$
$$0111 \mapsto 0111001$$
$$1000 \mapsto 1000111$$
$$1001 \mapsto 1001100$$
$$1010 \mapsto 1010010$$
$$1011 \mapsto 1011001$$
$$1100 \mapsto 1100001$$
$$1101 \mapsto 1101010$$
$$1110 \mapsto 1110100$$
$$1111 \mapsto 1111111$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

Example 11.2.2

An alternative encoding could be

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

$$\begin{array}{ll} 0001 \mapsto 0001011 & 1001 \mapsto 1001100 \\ 0010 \mapsto 0010101 & 1010 \mapsto 1010010 \\ 0011 \mapsto 0011110 & 1011 \mapsto 1011001 \\ 0100 \mapsto 0100110 & 1100 \mapsto 1100001 \\ 0101 \mapsto 0101100 & 1101 \mapsto 1101010 \\ 0110 \mapsto 0110011 & 1110 \mapsto 1110100 \\ 0111 \mapsto 0111001 & 1111 \mapsto 1111111 \\ 1000 \mapsto 1000111 & 0000 \mapsto 0000000 \end{array}$$

The single "1" bit values are mapped to the values of the individual columns

The other values are just sums of columns modulo 2

All m zeroes always maps to n zeroes