Today's outline - February 10, 2022

Today's outline - February 10, 2022

- Uniform superposition state

Today's outline - February 10, 2022

- Uniform superposition state
- Hamming distance & weight

Today's outline - February 10, 2022

- Uniform superposition state
- Hamming distance & weight
- Walsh-Hadamard transformation

Today's outline - February 10, 2022

- Uniform superposition state
- Hamming distance & weight
- Walsh-Hadamard transformation
- Complexity

Today's outline - February 10, 2022

- Uniform superposition state
- Hamming distance & weight
- Walsh-Hadamard transformation
- Complexity

Reading Assignment: Chapter 7.3-7.4

Today's outline - February 10, 2022

- Uniform superposition state
- Hamming distance & weight
- Walsh-Hadamard transformation
- Complexity

Reading Assignment: Chapter 7.3-7.4

Homework Assignment #04:
Chapter 5:4,6,9,15,16,17
due Tuesday, February 15, 2022

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right]$$

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle$$

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle$$

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle$$

Since the individual states are orthogonal

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle = |A|^2\sum_{x=0}^{2^n-1}\langle x|x\rangle$$

Since the individual states are orthogonal

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle = |A|^2\sum_{x=0}^{2^n-1}\langle x|x\rangle$$

Since the individual states are orthogonal and normalized,

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle = |A|^2\sum_{x=0}^{2^n-1}\langle x|x\rangle = |A|^2 2^n$$

Since the individual states are orthogonal and normalized,

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle = |A|^2\sum_{x=0}^{2^n-1}\langle x|x\rangle = |A|^2 2^n$$

Since the individual states are orthogonal and normalized, the normalization constant is

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle = |A|^2\sum_{x=0}^{2^n-1}\langle x|x\rangle = |A|^2 2^n \;\rightarrow\; A = \frac{1}{\sqrt{2^n}}$$

Since the individual states are orthogonal and normalized, the normalization constant is

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle = |A|^2\sum_{x=0}^{2^n-1}\langle x|x\rangle = |A|^2 2^n \;\rightarrow\; A = \frac{1}{\sqrt{2^n}}$$

Since the individual states are orthogonal and normalized, the normalization constant is

What does this mean in practice for systems with 1-3 qubits?

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle = |A|^2\sum_{x=0}^{2^n-1}\langle x|x\rangle = |A|^2 2^n \;\rightarrow\; A = \frac{1}{\sqrt{2^n}}$$

Since the individual states are orthogonal and normalized, the normalization constant is

What does this mean in practice for systems with 1-3 qubits?

$$n = 1: \quad |s\rangle = \tfrac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle = |A|^2\sum_{x=0}^{2^n-1}\langle x|x\rangle = |A|^2 2^n \;\rightarrow\; A = \frac{1}{\sqrt{2^n}}$$

Since the individual states are orthogonal and normalized, the normalization constant is

What does this mean in practice for systems with 1-3 qubits?

$$n=1: \quad |s\rangle = \tfrac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

$$n=2: \quad |s\rangle = \tfrac{1}{\sqrt{4}}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right)$$

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle = |A|^2\sum_{x=0}^{2^n-1}\langle x|x\rangle = |A|^2 2^n \;\rightarrow\; A = \frac{1}{\sqrt{2^n}}$$

Since the individual states are orthogonal and normalized, the normalization constant is

What does this mean in practice for systems with 1-3 qubits?

$$n=1: \quad |s\rangle = \tfrac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)$$

$$n=2: \quad |s\rangle = \tfrac{1}{\sqrt{4}}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right) = \tfrac{1}{\sqrt{4}}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle\right)$$

$$n=3: \quad |s\rangle = \tfrac{1}{\sqrt{8}}\left(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\right)$$

The uniform superposition state

An important state of an n qubit system is the so-called uniform superposition state, $|s\rangle$ which can be written

$$|s\rangle = A\left[|0\ldots00\rangle + |0\ldots01\rangle + |0\ldots10\rangle + \cdots + |1\ldots11\rangle\right] = A\sum_{x=0}^{2^n-1}|x\rangle$$

The uniform superposition state must be normalized

$$1 = \langle s|s\rangle = |A|^2\sum_{x'=0}^{2^n-1}\sum_{x=0}^{2^n-1}\langle x'|x\rangle = |A|^2\sum_{x=0}^{2^n-1}\langle x|x\rangle = |A|^2 2^n \;\rightarrow\; A = \frac{1}{\sqrt{2^n}}$$

Since the individual states are orthogonal and normalized, the normalization constant is

What does this mean in practice for systems with 1-3 qubits?

$$\begin{aligned}
n=1: \quad & |s\rangle = \tfrac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)\\
n=2: \quad & |s\rangle = \tfrac{1}{\sqrt{4}}\left(|00\rangle + |01\rangle + |10\rangle + |11\rangle\right) = \tfrac{1}{\sqrt{4}}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle\right)\\
n=3: \quad & |s\rangle = \tfrac{1}{\sqrt{8}}\left(|000\rangle + |001\rangle + |010\rangle + |011\rangle + |100\rangle + |101\rangle + |110\rangle + |111\rangle\right)\\
& \quad = \tfrac{1}{\sqrt{8}}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle\right)
\end{aligned}$$

The Walsh-Hadamard transformation

From the $n = 1$ example it is clear that $|s\rangle \equiv H|0\rangle$

The Walsh-Hadamard transformation

From the $n = 1$ example it is clear that $|s\rangle \equiv H|0\rangle$

Thus the uniform superposition state can be generated by applying the Hadamard transformation to each of the n qubits

The Walsh-Hadamard transformation

From the $n = 1$ example it is clear that $|s\rangle \equiv H|0\rangle$

Thus the uniform superposition state can be generated by applying the Hadamard transformation to each of the n qubits

$$H_n \otimes H_{n-1} \otimes \cdots \otimes H_1 |0_n 0_{n-1} \ldots 0_1\rangle$$

The Walsh-Hadamard transformation

From the $n = 1$ example it is clear that $|s\rangle \equiv H|0\rangle$

Thus the uniform superposition state can be generated by applying the Hadamard transformation to each of the n qubits

$$H_n \otimes H_{n-1} \otimes \cdots \otimes H_1 |0_n 0_{n-1} \ldots 0_1\rangle = \frac{1}{\sqrt{2^n}} \left[(|0_n\rangle + |1_n\rangle)(|0_{n-1}\rangle + |1_{n-1}\rangle) \cdots (|0_1\rangle + |1_1\rangle) \right]$$

The Walsh-Hadamard transformation

From the $n = 1$ example it is clear that $|s\rangle \equiv H|0\rangle$

Thus the uniform superposition state can be generated by applying the Hadamard transformation to each of the n qubits

$$H_n \otimes H_{n-1} \otimes \cdots \otimes H_1 |0_n 0_{n-1} \ldots 0_1\rangle = \frac{1}{\sqrt{2^n}} \left[(|0_n\rangle + |1_n\rangle)(|0_{n-1}\rangle + |1_{n-1}\rangle) \cdots (|0_1\rangle + |1_1\rangle) \right]$$

$$= \frac{1}{\sqrt{2^n}} \left[|0 \ldots 00\rangle + |0 \ldots 01\rangle + |0 \ldots 10\rangle + \cdots + |1 \ldots 11\rangle \right]$$

The Walsh-Hadamard transformation

From the $n = 1$ example it is clear that $|s\rangle \equiv H|0\rangle$

Thus the uniform superposition state can be generated by applying the Hadamard transformation to each of the n qubits

$$H_n \otimes H_{n-1} \otimes \cdots \otimes H_1 |0_n 0_{n-1} \ldots 0_1\rangle = \frac{1}{\sqrt{2^n}} \left[(|0_n\rangle + |1_n\rangle)(|0_{n-1}\rangle + |1_{n-1}\rangle) \cdots (|0_1\rangle + |1_1\rangle) \right]$$

$$= \frac{1}{\sqrt{2^n}} \left[|0 \ldots 00\rangle + |0 \ldots 01\rangle + |0 \ldots 10\rangle + \cdots + |1 \ldots 11\rangle \right] = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle = |s\rangle$$

The Walsh-Hadamard transformation

From the $n = 1$ example it is clear that $|s\rangle \equiv H|0\rangle$

Thus the uniform superposition state can be generated by applying the Hadamard transformation to each of the n qubits

$$H_n \otimes H_{n-1} \otimes \cdots \otimes H_1 |0_n 0_{n-1} \ldots 0_1\rangle = \frac{1}{\sqrt{2^n}} [(|0_n\rangle + |1_n\rangle)(|0_{n-1}\rangle + |1_{n-1}\rangle) \cdots (|0_1\rangle + |1_1\rangle)]$$

$$= \frac{1}{\sqrt{2^n}} [|0 \ldots 00\rangle + |0 \ldots 01\rangle + |0 \ldots 10\rangle + \cdots + |1 \ldots 11\rangle] = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle = |s\rangle$$

This transformation, applying H to each of the qubits is called the Walsh-Hadamard transformation and can be written

The Walsh-Hadamard transformation

From the $n = 1$ example it is clear that $|s\rangle \equiv H|0\rangle$

Thus the uniform superposition state can be generated by applying the Hadamard transformation to each of the n qubits

$$H_n \otimes H_{n-1} \otimes \cdots \otimes H_1 |0_n 0_{n-1} \ldots 0_1\rangle = \frac{1}{\sqrt{2^n}} [(|0_n\rangle + |1_n\rangle)(|0_{n-1}\rangle + |1_{n-1}\rangle) \cdots (|0_1\rangle + |1_1\rangle)]$$

$$= \frac{1}{\sqrt{2^n}} [|0 \ldots 00\rangle + |0 \ldots 01\rangle + |0 \ldots 10\rangle + \cdots + |1 \ldots 11\rangle] = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle = |s\rangle$$

This transformation, applying H to each of the qubits is called the Walsh-Hadamard transformation and can be written

$$W = H \otimes H \otimes \cdots \otimes H$$

The Walsh-Hadamard transformation

From the $n = 1$ example it is clear that $|s\rangle \equiv H|0\rangle$

Thus the uniform superposition state can be generated by applying the Hadamard transformation to each of the n qubits

$$H_n \otimes H_{n-1} \otimes \cdots \otimes H_1 |0_n 0_{n-1} \ldots 0_1\rangle = \frac{1}{\sqrt{2^n}} \left[(|0_n\rangle + |1_n\rangle)(|0_{n-1}\rangle + |1_{n-1}\rangle) \cdots (|0_1\rangle + |1_1\rangle) \right]$$

$$= \frac{1}{\sqrt{2^n}} \left[|0 \ldots 00\rangle + |0 \ldots 01\rangle + |0 \ldots 10\rangle + \cdots + |1 \ldots 11\rangle \right] = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle = |s\rangle$$

This transformation, applying H to each of the qubits is called the Walsh-Hadamard transformation and can be written

$$W = H \otimes H \otimes \cdots \otimes H$$

In shorthand notation the application of W is written

The Walsh-Hadamard transformation

From the $n = 1$ example it is clear that $|s\rangle \equiv H|0\rangle$

Thus the uniform superposition state can be generated by applying the Hadamard transformation to each of the n qubits

$$H_n \otimes H_{n-1} \otimes \cdots \otimes H_1 |0_n 0_{n-1} \ldots 0_1\rangle = \frac{1}{\sqrt{2^n}} \left[(|0_n\rangle + |1_n\rangle)(|0_{n-1}\rangle + |1_{n-1}\rangle) \cdots (|0_1\rangle + |1_1\rangle) \right]$$

$$= \frac{1}{\sqrt{2^n}} \left[|0 \ldots 00\rangle + |0 \ldots 01\rangle + |0 \ldots 10\rangle + \cdots + |1 \ldots 11\rangle \right] = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle = |s\rangle$$

This transformation, applying H to each of the qubits is called the Walsh-Hadamard transformation and can be written

$$W = H \otimes H \otimes \cdots \otimes H$$

In shorthand notation the application of W is written

$$W|0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle, \qquad N = 2^n$$

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Operations on two bit strings, x and y, are defined as:

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Operations on two bit strings, x and y, are defined as:

$x \cdot y$	the number of common bits with a value of 1

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Operations on two bit strings, x and y, are defined as:

$x \cdot y$	the number of common bits with a value of 1
$x \oplus y$	the bitwise exclusive-OR of the strings

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Operations on two bit strings, x and y, are defined as:

$x \cdot y$	the number of common bits with a value of 1
$x \oplus y$	the bitwise exclusive-OR of the strings
$x \wedge y$	the bitwise AND of the strings

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Operations on two bit strings, x and y, are defined as:

$x \cdot y$	the number of common bits with a value of 1
$x \oplus y$	the bitwise exclusive-OR of the strings
$x \wedge y$	the bitwise AND of the strings
$x \oplus 11\ldots1 = \neg x$	the bit string that flips all the bits in x

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Operations on two bit strings, x and y, are defined as:

$x \cdot y$	the number of common bits with a value of 1
$x \oplus y$	the bitwise exclusive-OR of the strings
$x \wedge y$	the bitwise AND of the strings
$x \oplus 11\ldots1 = \neg x$	the bit string that flips all the bits in x

Several identities arise from these definitions

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Operations on two bit strings, x and y, are defined as:

$x \cdot y$	the number of common bits with a value of 1
$x \oplus y$	the bitwise exclusive-OR of the strings
$x \wedge y$	the bitwise AND of the strings
$x \oplus 11\ldots1 = \neg x$	the bit string that flips all the bits in x

Several identities arise from these definitions

$$x \cdot y = d_H(x \wedge y)$$

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Operations on two bit strings, x and y, are defined as:

$x \cdot y$	the number of common bits with a value of 1
$x \oplus y$	the bitwise exclusive-OR of the strings
$x \wedge y$	the bitwise AND of the strings
$x \oplus 11\ldots1 = \neg x$	the bit string that flips all the bits in x

Several identities arise from these definitions

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Operations on two bit strings, x and y, are defined as:

$x \cdot y$	the number of common bits with a value of 1
$x \oplus y$	the bitwise exclusive-OR of the strings
$x \wedge y$	the bitwise AND of the strings
$x \oplus 11\ldots1 = \neg x$	the bit string that flips all the bits in x

Several identities arise from these definitions

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z)$$

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Operations on two bit strings, x and y, are defined as:

$x \cdot y$	the number of common bits with a value of 1
$x \oplus y$	the bitwise exclusive-OR of the strings
$x \wedge y$	the bitwise AND of the strings
$x \oplus 11\ldots1 = \neg x$	the bit string that flips all the bits in x

Several identities arise from these definitions

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

Hamming distance

A useful concept is the Hamming distance, $d_H(x, y)$ which is defined as the number of bits in which two bit strings x and y differ

The Hamming weight is the number of bits in which a bit string x differs from a string which is all zeroes, $d_H \equiv d_H(x, 0)$

Operations on two bit strings, x and y, are defined as:

$x \cdot y$	the number of common bits with a value of 1
$x \oplus y$	the bitwise exclusive-OR of the strings
$x \wedge y$	the bitwise AND of the strings
$x \oplus 11\ldots1 = \neg x$	the bit string that flips all the bits in x

Several identities arise from these definitions

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) \qquad \sum_{x=0}^{2^n-1} (-1)^{x \cdot y} = \begin{cases} 2^n & y = 0 \\ 0 & y \neq 0 \end{cases}$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

Bit string identities

$$x \cdot y = d_H(x \wedge y)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z)$$

$$(x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \frac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \frac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \frac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \frac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011)$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \frac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \frac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$\begin{aligned}(x \cdot y) \bmod 2 &= (1011 \cdot 0111) \bmod 2 \\ &= 2 \bmod 2\end{aligned}$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$\begin{aligned}(x \cdot y) \bmod 2 &= (1011 \cdot 0111) \bmod 2 \\ &= 2 \bmod 2 = 0\end{aligned}$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2 \qquad \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) = \tfrac{1}{2}\left(1 - (-1)^{2}\right)$$

$$= 2 \bmod 2 = 0$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2 = 2 \bmod 2 = 0$$

$$\tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) = \tfrac{1}{2}\left(1 - (-1)^{2}\right) = \tfrac{1}{2}(1 - 1) = 0$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \frac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2 = 2 \bmod 2 = 0$$

$$\frac{1}{2}\left(1 - (-1)^{x \cdot y}\right) = \frac{1}{2}\left(1 - (-1)^{2}\right) = \frac{1}{2}(1 - 1) = 0$$

$$x \cdot y + x \cdot z = 1011 \cdot 0111 + 1011 \cdot 0010$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \frac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2 = 2 \bmod 2 = 0$$

$$\frac{1}{2}\left(1 - (-1)^{x \cdot y}\right) = \frac{1}{2}\left(1 - (-1)^2\right) = \frac{1}{2}(1 - 1) = 0$$

$$x \cdot y + x \cdot z = 1011 \cdot 0111 + 1011 \cdot 0010 = 2 + 1 = 3$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2 = 2 \bmod 2 = 0$$

$$\tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) = \tfrac{1}{2}\left(1 - (-1)^2\right) = \tfrac{1}{2}(1 - 1) = 0$$

$$x \cdot y + x \cdot z = 1011 \cdot 0111 + 1011 \cdot 0010 = 2 + 1 = 3 \qquad x \cdot (y \oplus z) = 1011 \cdot (0111 \oplus 0010)$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2 = 2 \bmod 2 = 0$$

$$\tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) = \tfrac{1}{2}\left(1 - (-1)^2\right) = \tfrac{1}{2}(1 - 1) = 0$$

$$x \cdot y + x \cdot z = 1011 \cdot 0111 + 1011 \cdot 0010 = 2 + 1 = 3$$

$$x \cdot (y \oplus z) = 1011 \cdot (0111 \oplus 0010) = 1011 \cdot 0101$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2 = 2 \bmod 2 = 0$$

$$\tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) = \tfrac{1}{2}\left(1 - (-1)^{2}\right) = \tfrac{1}{2}(1 - 1) = 0$$

$$x \cdot y + x \cdot z = 1011 \cdot 0111 + 1011 \cdot 0010 = 2 + 1 = 3 =_2 1$$

$$x \cdot (y \oplus z) = 1011 \cdot (0111 \oplus 0010) = 1011 \cdot 0101 = 1$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2 = 2 \bmod 2 = 0$$

$$\tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) = \tfrac{1}{2}\left(1 - (-1)^2\right) = \tfrac{1}{2}(1 - 1) = 0$$

$$x \cdot y + x \cdot z = 1011 \cdot 0111 + 1011 \cdot 0010 = 2 + 1 = 3 =_2 1$$

$$x \cdot (y \oplus z) = 1011 \cdot (0111 \oplus 0010) = 1011 \cdot 0101 = 1$$

$$d_H(x \oplus y) = d_H(1011 \oplus 0111)$$

Bit string identities

$$x \cdot y = d_H(x \wedge y)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z)$$

$$(x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2$$

$$d_H(x \wedge y) = d_H(0011) = 2$$

$$\begin{aligned}(x \cdot y) \bmod 2 &= (1011 \cdot 0111) \bmod 2 \\ &= 2 \bmod 2 = 0\end{aligned}$$

$$\begin{aligned}\tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) &= \tfrac{1}{2}\left(1 - (-1)^2\right) \\ &= \tfrac{1}{2}(1 - 1) = 0\end{aligned}$$

$$\begin{aligned}x \cdot y + x \cdot z &= 1011 \cdot 0111 + 1011 \cdot 0010 \\ &= 2 + 1 = 3 =_2 1\end{aligned}$$

$$\begin{aligned}x \cdot (y \oplus z) &= 1011 \cdot (0111 \oplus 0010) \\ &= 1011 \cdot 0101 = 1\end{aligned}$$

$$\begin{aligned}d_H(x \oplus y) &= d_H(1011 \oplus 0111) \\ &= d_H(1100)\end{aligned}$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$\begin{aligned}(x \cdot y) \bmod 2 &= (1011 \cdot 0111) \bmod 2 \\ &= 2 \bmod 2 = 0\end{aligned}$$

$$\begin{aligned}\tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) &= \tfrac{1}{2}\left(1 - (-1)^2\right) \\ &= \tfrac{1}{2}(1 - 1) = 0\end{aligned}$$

$$\begin{aligned}x \cdot y + x \cdot z &= 1011 \cdot 0111 + 1011 \cdot 0010 \\ &= 2 + 1 = 3 =_2 1\end{aligned}$$

$$\begin{aligned}x \cdot (y \oplus z) &= 1011 \cdot (0111 \oplus 0010) \\ &= 1011 \cdot 0101 = 1\end{aligned}$$

$$\begin{aligned}d_H(x \oplus y) &= d_H(1011 \oplus 0111) \\ &= d_H(1100) = 2\end{aligned}$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2 = 2 \bmod 2 = 0$$

$$\tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) = \tfrac{1}{2}\left(1 - (-1)^{2}\right) = \tfrac{1}{2}(1 - 1) = 0$$

$$x \cdot y + x \cdot z = 1011 \cdot 0111 + 1011 \cdot 0010 = 2 + 1 = 3 =_2 1$$

$$x \cdot (y \oplus z) = 1011 \cdot (0111 \oplus 0010) = 1011 \cdot 0101 = 1$$

$$d_H(x \oplus y) = d_H(1011 \oplus 0111) = d_H(1100) = 2$$

$$d_H(x) + d_H(y) = d_H(1011) + d_H(0111)$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2 = 2 \bmod 2 = 0$$

$$\tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) = \tfrac{1}{2}\left(1 - (-1)^{2}\right) = \tfrac{1}{2}(1 - 1) = 0$$

$$x \cdot y + x \cdot z = 1011 \cdot 0111 + 1011 \cdot 0010 = 2 + 1 = 3 =_2 1$$

$$x \cdot (y \oplus z) = 1011 \cdot (0111 \oplus 0010) = 1011 \cdot 0101 = 1$$

$$d_H(x \oplus y) = d_H(1011 \oplus 0111) = d_H(1100) = 2$$

$$d_H(x) + d_H(y) = d_H(1011) + d_H(0111) = 3 + 3 = 6$$

Bit string identities

$$x \cdot y = d_H(x \wedge y) \qquad (x \cdot y) \bmod 2 = \tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right)$$

$$x \cdot y + x \cdot z =_2 x \cdot (y \oplus z) \qquad d_H(x \oplus y) =_2 d_H(x) + d_H(y)$$

If we have $x = 1011$, $y = 0111$, and $z = 0010$ we these identities can be easily checked

$$x \cdot y = 1011 \cdot 0111 = 2 \qquad d_H(x \wedge y) = d_H(0011) = 2$$

$$(x \cdot y) \bmod 2 = (1011 \cdot 0111) \bmod 2 = 2 \bmod 2 = 0$$

$$\tfrac{1}{2}\left(1 - (-1)^{x \cdot y}\right) = \tfrac{1}{2}\left(1 - (-1)^2\right) = \tfrac{1}{2}(1 - 1) = 0$$

$$x \cdot y + x \cdot z = 1011 \cdot 0111 + 1011 \cdot 0010 = 2 + 1 = 3 =_2 1$$

$$x \cdot (y \oplus z) = 1011 \cdot (0111 \oplus 0010) = 1011 \cdot 0101 = 1$$

$$d_H(x \oplus y) = d_H(1011 \oplus 0111) = d_H(1100) = 2 =_2 0$$

$$d_H(x) + d_H(y) = d_H(1011) + d_H(0111) = 3 + 3 = 6 =_2 0$$

The Walsh-Hadamard matrix

In the standard basis, the matrix representation of W is a $2^n \times 2^n$ matrix with entries given by

The Walsh-Hadamard matrix

In the standard basis, the matrix representation of W is a $2^n \times 2^n$ matrix with entries given by

$$W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s},$$

The Walsh-Hadamard matrix

In the standard basis, the matrix representation of W is a $2^n \times 2^n$ matrix with entries given by

$$W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}, \qquad 0 \leq r, s \leq 2^n - 1$$

The Walsh-Hadamard matrix

In the standard basis, the matrix representation of W is a $2^n \times 2^n$ matrix with entries given by

$$W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}, \qquad 0 \leq r, s \leq 2^n - 1$$

This states that for a given string $|r\rangle$ the $|r\rangle^{th}$ column and row of W is a set of ± 1 values that depend on the number of common one-bits between $|r\rangle$ and each possible value of $|s\rangle$

The Walsh-Hadamard matrix

In the standard basis, the matrix representation of W is a $2^n \times 2^n$ matrix with entries given by

$$W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}, \quad 0 \leq r, s \leq 2^n - 1$$

This states that for a given string $|r\rangle$ the $|r\rangle^{th}$ column and row of W is a set of ± 1 values that depend on the number of common one-bits between $|r\rangle$ and each possible value of $|s\rangle$

The $|r\rangle^{th}$ column is the the Walsh-Hadamard transformation applied to $|r\rangle$ and is given by

The Walsh-Hadamard matrix

In the standard basis, the matrix representation of W is a $2^n \times 2^n$ matrix with entries given by

$$W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}, \quad 0 \leq r, s \leq 2^n - 1$$

This states that for a given string $|r\rangle$ the $|r\rangle^{th}$ column and row of W is a set of ± 1 values that depend on the number of common one-bits between $|r\rangle$ and each possible value of $|s\rangle$

The $|r\rangle^{th}$ column is the the Walsh-Hadamard transformation applied to $|r\rangle$ and is given by

$$W|r\rangle = \sum_{s=0}^{2^n-1} W_{rs}|s\rangle$$

The Walsh-Hadamard matrix

In the standard basis, the matrix representation of W is a $2^n \times 2^n$ matrix with entries given by

$$W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}, \qquad 0 \leq r, s \leq 2^n - 1$$

This states that for a given string $|r\rangle$ the $|r\rangle^{th}$ column and row of W is a set of ± 1 values that depend on the number of common one-bits between $|r\rangle$ and each possible value of $|s\rangle$

The $|r\rangle^{th}$ column is the the Walsh-Hadamard transformation applied to $|r\rangle$ and is given by

$$W|r\rangle = \sum_{s=0}^{2^n-1} W_{rs}|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{r \cdot s}|s\rangle$$

The Walsh-Hadamard matrix

In the standard basis, the matrix representation of W is a $2^n \times 2^n$ matrix with entries given by

$$W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}, \quad 0 \leq r, s \leq 2^n - 1$$

This states that for a given string $|r\rangle$ the $|r\rangle^{th}$ column and row of W is a set of ± 1 values that depend on the number of common one-bits between $|r\rangle$ and each possible value of $|s\rangle$

The $|r\rangle^{th}$ column is the the Walsh-Hadamard transformation applied to $|r\rangle$ and is given by

$$W|r\rangle = \sum_{s=0}^{2^n-1} W_{rs}|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{r \cdot s}|s\rangle$$

$$W|r\rangle = (H \otimes \cdots \otimes H)(|r_{n-1}\rangle \otimes \cdots \otimes |r_0\rangle)$$

The Walsh-Hadamard matrix

In the standard basis, the matrix representation of W is a $2^n \times 2^n$ matrix with entries given by

$$W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}, \quad 0 \leq r, s \leq 2^n - 1$$

This states that for a given string $|r\rangle$ the $|r\rangle^{th}$ column and row of W is a set of ± 1 values that depend on the number of common one-bits between $|r\rangle$ and each possible value of $|s\rangle$

The $|r\rangle^{th}$ column is the the Walsh-Hadamard transformation applied to $|r\rangle$ and is given by

$$W|r\rangle = \sum_{s=0}^{2^n-1} W_{rs}|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{r \cdot s}|s\rangle$$

$$W|r\rangle = (H \otimes \cdots \otimes H)(|r_{n-1}\rangle \otimes \cdots \otimes |r_0\rangle) = \frac{1}{\sqrt{2^n}}[|0\rangle + (-1)^{r_{n-1}}|1\rangle] \otimes \cdots \otimes [|0\rangle + (-1)^{r_0}|1\rangle]$$

The Walsh-Hadamard matrix

In the standard basis, the matrix representation of W is a $2^n \times 2^n$ matrix with entries given by

$$W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r\cdot s}, \qquad 0 \leq r, s \leq 2^n - 1$$

This states that for a given string $|r\rangle$ the $|r\rangle^{th}$ column and row of W is a set of ± 1 values that depend on the number of common one-bits between $|r\rangle$ and each possible value of $|s\rangle$

The $|r\rangle^{th}$ column is the the Walsh-Hadamard transformation applied to $|r\rangle$ and is given by

$$W|r\rangle = \sum_{s=0}^{2^n-1} W_{rs}|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{r\cdot s}|s\rangle$$

$$W|r\rangle = (H \otimes \cdots \otimes H)(|r_{n-1}\rangle \otimes \cdots \otimes |r_0\rangle) = \frac{1}{\sqrt{2^n}}[|0\rangle + (-1)^{r_{n-1}}|1\rangle] \otimes \cdots \otimes [|0\rangle + (-1)^{r_0}|1\rangle]$$

$$= \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{s_{n-1}r_{n-1}}|s_{n-1}\rangle \otimes \cdots \otimes (-1)^{s_0 r_0}|s_0\rangle$$

The Walsh-Hadamard matrix

In the standard basis, the matrix representation of W is a $2^n \times 2^n$ matrix with entries given by

$$W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r\cdot s}, \qquad 0 \leq r, s \leq 2^n - 1$$

This states that for a given string $|r\rangle$ the $|r\rangle^{th}$ column and row of W is a set of ± 1 values that depend on the number of common one-bits between $|r\rangle$ and each possible value of $|s\rangle$

The $|r\rangle^{th}$ column is the the Walsh-Hadamard transformation applied to $|r\rangle$ and is given by

$$W|r\rangle = \sum_{s=0}^{2^n-1} W_{rs}|s\rangle = \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{r\cdot s}|s\rangle$$

$$W|r\rangle = (H \otimes \cdots \otimes H)(|r_{n-1}\rangle \otimes \cdots \otimes |r_0\rangle) = \frac{1}{\sqrt{2^n}}[|0\rangle + (-1)^{r_{n-1}}|1\rangle] \otimes \cdots \otimes [|0\rangle + (-1)^{r_0}|1\rangle]$$

$$= \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{s_{n-1}r_{n-1}}|s_{n-1}\rangle \otimes \cdots \otimes (-1)^{s_0 r_0}|s_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{s=0}^{2^n-1} (-1)^{r\cdot s}|s\rangle$$

A simple example

Consider a 2-qubit system where we wish to define the Walsh-Hadamard transformation matrix

A simple example

Consider a 2-qubit system where we wish to define the Walsh-Hadamard transformation matrix

For each of the 4 possible states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ we can generate the transformation by $W|r\rangle = (H \otimes H)|r\rangle$

A simple example

Consider a 2-qubit system where we wish to define the Walsh-Hadamard transformation matrix

For each of the 4 possible states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ we can generate the transformation by $W|r\rangle = (H \otimes H)|r\rangle$

$$W|00\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)\right]$$

$$W = \frac{1}{2}\begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

A simple example

Consider a 2-qubit system where we wish to define the Walsh-Hadamard transformation matrix

For each of the 4 possible states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ we can generate the transformation by $W|r\rangle = (H \otimes H)|r\rangle$

$$W|00\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle + |01\rangle + |10\rangle + |11\rangle\right]$$

$$W = \frac{1}{2}\begin{pmatrix} 1 & & & \\ 1 & & & \\ 1 & & & \\ 1 & & & \end{pmatrix}$$

A simple example

Consider a 2-qubit system where we wish to define the Walsh-Hadamard transformation matrix

For each of the 4 possible states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ we can generate the transformation by $W|r\rangle = (H \otimes H)|r\rangle$

$$W|00\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle + |01\rangle + |10\rangle + |11\rangle\right]$$

$$W|01\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle)\right]$$

$$W = \frac{1}{2}\begin{pmatrix} 1 & & & \\ 1 & & & \\ 1 & & & \\ 1 & & & \end{pmatrix}$$

A simple example

Consider a 2-qubit system where we wish to define the Walsh-Hadamard transformation matrix

For each of the 4 possible states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ we can generate the transformation by $W|r\rangle = (H \otimes H)|r\rangle$

$$W|00\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle + |01\rangle + |10\rangle + |11\rangle\right]$$
$$W|01\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle - |01\rangle + |10\rangle - |11\rangle\right]$$

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 1 \\ 1 & -1 \end{pmatrix}$$

A simple example

Consider a 2-qubit system where we wish to define the Walsh-Hadamard transformation matrix

For each of the 4 possible states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ we can generate the transformation by $W|r\rangle = (H \otimes H)|r\rangle$

$$W|00\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle + |01\rangle + |10\rangle + |11\rangle\right]$$
$$W|01\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle - |01\rangle + |10\rangle - |11\rangle\right]$$
$$W|10\rangle = \tfrac{1}{2}\left[(|0\rangle - |1\rangle) \otimes (|0\rangle + |1\rangle)\right]$$

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 & & \\ 1 & -1 & & \\ 1 & 1 & & \\ 1 & -1 & & \end{pmatrix}$$

A simple example

Consider a 2-qubit system where we wish to define the Walsh-Hadamard transformation matrix

For each of the 4 possible states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ we can generate the transformation by $W|r\rangle = (H \otimes H)|r\rangle$

$$W|00\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle + |01\rangle + |10\rangle + |11\rangle\right]$$
$$W|01\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle - |01\rangle + |10\rangle - |11\rangle\right]$$
$$W|10\rangle = \tfrac{1}{2}\left[(|0\rangle - |1\rangle) \otimes (|0\rangle + |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle + |01\rangle - |10\rangle - |11\rangle\right]$$

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & \\ 1 & -1 & 1 & \\ 1 & 1 & -1 & \\ 1 & -1 & -1 & \end{pmatrix}$$

A simple example

Consider a 2-qubit system where we wish to define the Walsh-Hadamard transformation matrix

For each of the 4 possible states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ we can generate the transformation by $W|r\rangle = (H \otimes H)|r\rangle$

$$W|00\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle + |01\rangle + |10\rangle + |11\rangle\right]$$
$$W|01\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle - |01\rangle + |10\rangle - |11\rangle\right]$$
$$W|10\rangle = \tfrac{1}{2}\left[(|0\rangle - |1\rangle) \otimes (|0\rangle + |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle + |01\rangle - |10\rangle - |11\rangle\right]$$
$$W|11\rangle = \tfrac{1}{2}\left[(|0\rangle - |1\rangle) \otimes (|0\rangle - |1\rangle)\right]$$

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & \\ 1 & -1 & 1 & \\ 1 & 1 & -1 & \\ 1 & -1 & -1 & \end{pmatrix}$$

A simple example

Consider a 2-qubit system where we wish to define the Walsh-Hadamard transformation matrix

For each of the 4 possible states $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$ we can generate the transformation by $W|r\rangle = (H \otimes H)|r\rangle$

$$W|00\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle + |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle + |01\rangle + |10\rangle + |11\rangle\right]$$
$$W|01\rangle = \tfrac{1}{2}\left[(|0\rangle + |1\rangle) \otimes (|0\rangle - |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle - |01\rangle + |10\rangle - |11\rangle\right]$$
$$W|10\rangle = \tfrac{1}{2}\left[(|0\rangle - |1\rangle) \otimes (|0\rangle + |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle + |01\rangle - |10\rangle - |11\rangle\right]$$
$$W|11\rangle = \tfrac{1}{2}\left[(|0\rangle - |1\rangle) \otimes (|0\rangle - |1\rangle)\right] = \tfrac{1}{2}\left[|00\rangle - |01\rangle - |10\rangle + |11\rangle\right]$$

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$|r\rangle \quad |s\rangle \quad W_{sr}$

$$W = \frac{1}{2} \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$\|r\rangle$	$\|s\rangle$	W_{sr}
$\|00\rangle$	$\|00\rangle$	1

$$W = \frac{1}{2} \begin{pmatrix} 1 & & & \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$\|r\rangle$	$\|s\rangle$	W_{sr}
$\|00\rangle$	$\|00\rangle$	1
$\|00\rangle$	$\|01\rangle$	1

$$W = \frac{1}{2} \begin{pmatrix} 1 & 1 & & \\ 1 & & & \\ & & & \\ & & & \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$\|r\rangle$	$\|s\rangle$	W_{sr}
$\|00\rangle$	$\|00\rangle$	1
$\|00\rangle$	$\|01\rangle$	1
$\|00\rangle$	$\|10\rangle$	1

$$W = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & \\ 1 & & & \\ 1 & & & \\ & & & \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$\|r\rangle$	$\|s\rangle$	W_{sr}
$\|00\rangle$	$\|00\rangle$	1
$\|00\rangle$	$\|01\rangle$	1
$\|00\rangle$	$\|10\rangle$	1
$\|00\rangle$	$\|11\rangle$	1

$$W = \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & & & \\ 1 & & & \\ 1 & & & \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$\|r\rangle$	$\|s\rangle$	W_{sr}
$\|00\rangle$	$\|00\rangle$	1
$\|00\rangle$	$\|01\rangle$	1
$\|00\rangle$	$\|10\rangle$	1
$\|00\rangle$	$\|11\rangle$	1
$\|01\rangle$	$\|01\rangle$	-1

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & & \\ 1 & & & \\ 1 & & & \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$\|r\rangle$	$\|s\rangle$	W_{sr}
$\|00\rangle$	$\|00\rangle$	1
$\|00\rangle$	$\|01\rangle$	1
$\|00\rangle$	$\|10\rangle$	1
$\|00\rangle$	$\|11\rangle$	1
$\|01\rangle$	$\|01\rangle$	-1
$\|01\rangle$	$\|10\rangle$	1

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & \\ 1 & 1 & & \\ 1 & & & \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$\|r\rangle$	$\|s\rangle$	W_{sr}
$\|00\rangle$	$\|00\rangle$	1
$\|00\rangle$	$\|01\rangle$	1
$\|00\rangle$	$\|10\rangle$	1
$\|00\rangle$	$\|11\rangle$	1
$\|01\rangle$	$\|01\rangle$	-1
$\|01\rangle$	$\|10\rangle$	1
$\|01\rangle$	$\|11\rangle$	-1

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & & \\ 1 & -1 & & \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$\|r\rangle$	$\|s\rangle$	W_{sr}
$\|00\rangle$	$\|00\rangle$	1
$\|00\rangle$	$\|01\rangle$	1
$\|00\rangle$	$\|10\rangle$	1
$\|00\rangle$	$\|11\rangle$	1
$\|01\rangle$	$\|01\rangle$	-1
$\|01\rangle$	$\|10\rangle$	1
$\|01\rangle$	$\|11\rangle$	-1
$\|10\rangle$	$\|10\rangle$	-1

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & \\ 1 & -1 & & \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$\|r\rangle$	$\|s\rangle$	W_{sr}
$\|00\rangle$	$\|00\rangle$	1
$\|00\rangle$	$\|01\rangle$	1
$\|00\rangle$	$\|10\rangle$	1
$\|00\rangle$	$\|11\rangle$	1
$\|01\rangle$	$\|01\rangle$	-1
$\|01\rangle$	$\|10\rangle$	1
$\|01\rangle$	$\|11\rangle$	-1
$\|10\rangle$	$\|10\rangle$	-1
$\|10\rangle$	$\|11\rangle$	-1

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$\|r\rangle$	$\|s\rangle$	W_{sr}
$\|00\rangle$	$\|00\rangle$	1
$\|00\rangle$	$\|01\rangle$	1
$\|00\rangle$	$\|10\rangle$	1
$\|00\rangle$	$\|11\rangle$	1
$\|01\rangle$	$\|01\rangle$	-1
$\|01\rangle$	$\|10\rangle$	1
$\|01\rangle$	$\|11\rangle$	-1
$\|10\rangle$	$\|10\rangle$	-1
$\|10\rangle$	$\|11\rangle$	-1
$\|11\rangle$	$\|11\rangle$	1

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

A simple example

Now let's generate the W matrix using the relation $W_{sr} = W_{rs} = \frac{1}{\sqrt{2^n}}(-1)^{r \cdot s}$ with $0 \leq r, s \leq 2^n - 1$

$\|r\rangle$	$\|s\rangle$	W_{sr}
$\|00\rangle$	$\|00\rangle$	1
$\|00\rangle$	$\|01\rangle$	1
$\|00\rangle$	$\|10\rangle$	1
$\|00\rangle$	$\|11\rangle$	1
$\|01\rangle$	$\|01\rangle$	-1
$\|01\rangle$	$\|10\rangle$	1
$\|01\rangle$	$\|11\rangle$	-1
$\|10\rangle$	$\|10\rangle$	-1
$\|10\rangle$	$\|11\rangle$	-1
$\|11\rangle$	$\|11\rangle$	1

$$W = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$$

This is the identical matrix generated by the first method

Quantum parallelism

Suppose that we have two registers of qubits, $|x\rangle$ and $|y\rangle$ of length n and m, respectively

Quantum parallelism

Suppose that we have two registers of qubits, $|x\rangle$ and $|y\rangle$ of length n and m, respectively

A linear transformation U_f which acts on the combined registers $|x\rangle \otimes |y\rangle$ acts on the registers as $U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$

Quantum parallelism

Suppose that we have two registers of qubits, $|x\rangle$ and $|y\rangle$ of length n and m, respectively

A linear transformation U_f which acts on the combined registers $|x\rangle \otimes |y\rangle$ acts on the registers as $U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$

This operator can also act on a superposition $\sum a_x |x\rangle$ as

Quantum parallelism

Suppose that we have two registers of qubits, $|x\rangle$ and $|y\rangle$ of length n and m, respectively

A linear transformation U_f which acts on the combined registers $|x\rangle \otimes |y\rangle$ acts on the registers as $U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$

This operator can also act on a superposition $\sum a_x|x\rangle$ as

$$U_f : \sum_x a_x|x, 0\rangle \quad \longrightarrow \quad \sum_x a_x|x, f(x)\rangle$$

Quantum parallelism

Suppose that we have two registers of qubits, $|x\rangle$ and $|y\rangle$ of length n and m, respectively

A linear transformation U_f which acts on the combined registers $|x\rangle \otimes |y\rangle$ acts on the registers as $U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$

This operator can also act on a superposition $\sum a_x|x\rangle$ as

$$U_f : \sum_x a_x|x, 0\rangle \quad \longrightarrow \quad \sum_x a_x|x, f(x)\rangle$$

Apply the U_f operator to the uniform superposition state obtained from the Walsh-Hadamard transformation

Quantum parallelism

Suppose that we have two registers of qubits, $|x\rangle$ and $|y\rangle$ of length n and m, respectively

A linear transformation U_f which acts on the combined registers $|x\rangle \otimes |y\rangle$ acts on the registers as $U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$

This operator can also act on a superposition $\sum a_x|x\rangle$ as

$$U_f : \sum_x a_x|x, 0\rangle \quad \longrightarrow \quad \sum_x a_x|x, f(x)\rangle$$

Apply the U_f operator to the uniform superposition state obtained from the Walsh-Hadamard transformation

$$U_f : (W|0\rangle) \otimes |0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle|0\rangle$$

Quantum parallelism

Suppose that we have two registers of qubits, $|x\rangle$ and $|y\rangle$ of length n and m, respectively

A linear transformation U_f which acts on the combined registers $|x\rangle \otimes |y\rangle$ acts on the registers as $U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$

This operator can also act on a superposition $\sum a_x|x\rangle$ as

$$U_f : \sum_x a_x|x, 0\rangle \quad \longrightarrow \quad \sum_x a_x|x, f(x)\rangle$$

Apply the U_f operator to the uniform superposition state obtained from the Walsh-Hadamard transformation

$$U_f : (W|0\rangle) \otimes |0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle|0\rangle \quad \longrightarrow \quad \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle|f(x)\rangle$$

The resultant state is one where all 2^n $|f(x)\rangle$ values entangled with their corresponding input values, $|x\rangle$

Quantum parallelism

Suppose that we have two registers of qubits, $|x\rangle$ and $|y\rangle$ of length n and m, respectively

A linear transformation U_f which acts on the combined registers $|x\rangle \otimes |y\rangle$ acts on the registers as $U_f : |x, y\rangle \rightarrow |x, y \oplus f(x)\rangle$

This operator can also act on a superposition $\sum a_x|x\rangle$ as

$$U_f : \sum_x a_x|x, 0\rangle \quad \longrightarrow \quad \sum_x a_x|x, f(x)\rangle$$

Apply the U_f operator to the uniform superposition state obtained from the Walsh-Hadamard transformation

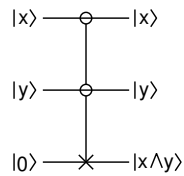
$$U_f : (W|0\rangle) \otimes |0\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle|0\rangle \quad \longrightarrow \quad \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle|f(x)\rangle$$

The resultant state is one where all 2^n $|f(x)\rangle$ values entangled with their corresponding input values, $|x\rangle$

In principle, it is now possible to operate on all possible combinations simultaneously in an effect called quantum parallelism but other transformations must be applied to make it useful

The Toffoli gate

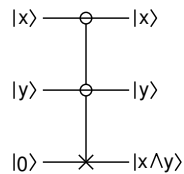
The Toffoli gate, computes the conjunction of two values, $|x\rangle$ and $|y\rangle$ with the output going to a register initially set to $|0\rangle$



The Toffoli gate

The Toffoli gate, computes the conjunction of two values, $|x\rangle$ and $|y\rangle$ with the output going to a register initially set to $|0\rangle$

First construct the universal superposition of the two input qubits

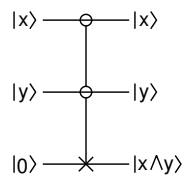


The Toffoli gate

The Toffoli gate, computes the conjunction of two values, $|x\rangle$ and $|y\rangle$ with the output going to a register initially set to $|0\rangle$

First construct the universal superposition of the two input qubits

$$W(|00\rangle) \otimes |0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle$$

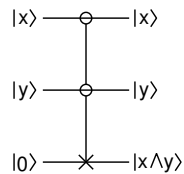


The Toffoli gate

The Toffoli gate, computes the conjunction of two values, $|x\rangle$ and $|y\rangle$ with the output going to a register initially set to $|0\rangle$

First construct the universal superposition of the two input qubits

$$\begin{aligned} W(|00\rangle) \otimes |0\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\ &= \tfrac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle) \end{aligned}$$



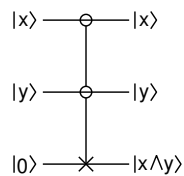
The Toffoli gate

The Toffoli gate, computes the conjunction of two values, $|x\rangle$ and $|y\rangle$ with the output going to a register initially set to $|0\rangle$

First construct the universal superposition of the two input qubits

$$\begin{aligned} W(|00\rangle) \otimes |0\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\ &= \tfrac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle) \end{aligned}$$

Applying the Toffoli gate, we have

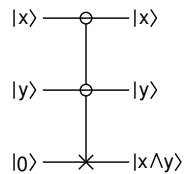


The Toffoli gate

The Toffoli gate, computes the conjunction of two values, $|x\rangle$ and $|y\rangle$ with the output going to a register initially set to $|0\rangle$

First construct the universal superposition of the two input qubits

$$\begin{aligned} W(|00\rangle) \otimes |0\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\ &= \tfrac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle) \end{aligned}$$



Applying the Toffoli gate, we have

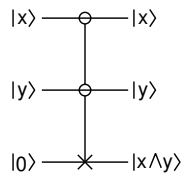
$$T\left[W|00\rangle \otimes |0\rangle\right] = T\left[\tfrac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle)\right]$$

The Toffoli gate

The Toffoli gate, computes the conjunction of two values, $|x\rangle$ and $|y\rangle$ with the output going to a register initially set to $|0\rangle$

First construct the universal superposition of the two input qubits

$$\begin{aligned} W(|00\rangle) \otimes |0\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\ &= \tfrac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle) \end{aligned}$$



Applying the Toffoli gate, we have

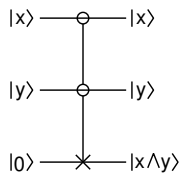
$$\begin{aligned} T\left[W|00\rangle \otimes |0\rangle\right] &= T\left[\tfrac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle)\right] \\ &= \tfrac{1}{2}\left[|000\rangle + |010\rangle + |100\rangle + |111\rangle\right] \end{aligned}$$

The Toffoli gate

The Toffoli gate, computes the conjunction of two values, $|x\rangle$ and $|y\rangle$ with the output going to a register initially set to $|0\rangle$

First construct the universal superposition of the two input qubits

$$\begin{aligned} W(|00\rangle) \otimes |0\rangle &= \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \tfrac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes |0\rangle \\ &= \tfrac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle) \end{aligned}$$



Applying the Toffoli gate, we have

$$\begin{aligned} T\left[W|00\rangle \otimes |0\rangle\right] &= T\left[\tfrac{1}{2}(|000\rangle + |010\rangle + |100\rangle + |110\rangle)\right] \\ &= \tfrac{1}{2}\left[|000\rangle + |010\rangle + |100\rangle + |111\rangle\right] \end{aligned}$$

While the entire truth table for the Toffoli gate is present in this entangled state, it is only possible to extract one value with a measurement of $|x\rangle \otimes |y\rangle$

Circuit complexity

A circuit family $\mathcal{C} = \{C_n\}$ is made up of circuits C_n indexed by their maximum input size, the circuit C_5 handles 5-qubit sized inputs

Circuit complexity

A circuit family $\mathcal{C} = \{C_n\}$ is made up of circuits C_n indexed by their maximum input size, the circuit C_5 handles 5-qubit sized inputs

The circuit complexity of a circuit is the number of simple gates in the circuit

Circuit complexity

A circuit family $\mathcal{C} = \{C_n\}$ is made up of circuits C_n indexed by their maximum input size, the circuit C_5 handles 5-qubit sized inputs

The circuit complexity of a circuit is the number of simple gates in the circuit

The complexity of a circuit family is the asymptotic number of simple gates expresesd in terms of the input size

Circuit complexity

A circuit family $\mathcal{C} = \{C_n\}$ is made up of circuits C_n indexed by their maximum input size, the circuit C_5 handles 5-qubit sized inputs

The circuit complexity of a circuit is the number of simple gates in the circuit

The complexity of a circuit family is the asymptotic number of simple gates expresesd in terms of the input size

Circuit complexity is important as it relates directly to the amount of resources required to perform the computation, thus a good model of circuit complexity is required when planning a quantum circuit

Circuit complexity

A circuit family $\mathcal{C} = \{C_n\}$ is made up of circuits C_n indexed by their maximum input size, the circuit C_5 handles 5-qubit sized inputs

The circuit complexity of a circuit is the number of simple gates in the circuit

The complexity of a circuit family is the asymptotic number of simple gates expresesd in terms of the input size

Circuit complexity is important as it relates directly to the amount of resources required to perform the computation, thus a good model of circuit complexity is required when planning a quantum circuit

A valid model for circuit complexity must be both uniform and consistent

Circuit complexity

A circuit family $\mathcal{C} = \{C_n\}$ is made up of circuits C_n indexed by their maximum input size, the circuit C_5 handles 5-qubit sized inputs

The circuit complexity of a circuit is the number of simple gates in the circuit

The complexity of a circuit family is the asymptotic number of simple gates expresesd in terms of the input size

Circuit complexity is important as it relates directly to the amount of resources required to perform the computation, thus a good model of circuit complexity is required when planning a quantum circuit

A valid model for circuit complexity must be both uniform and consistent

A quantum circuit family $\mathcal{C}$ is consistent if its circuits C_n gove consistent results: for all $m < n$, applying C_n to input $|x\rangle$ of size m must give the same result as applying C_m to the same input

Circuit complexity

A circuit family $\mathcal{C} = \{C_n\}$ is made up of circuits C_n indexed by their maximum input size, the circuit C_5 handles 5-qubit sized inputs

The circuit complexity of a circuit is the number of simple gates in the circuit

The complexity of a circuit family is the asymptotic number of simple gates expresesd in terms of the input size

Circuit complexity is important as it relates directly to the amount of resources required to perform the computation, thus a good model of circuit complexity is required when planning a quantum circuit

A valid model for circuit complexity must be both uniform and consistent

A quantum circuit family $\mathcal{C}$ is consistent if its circuits C_n gove consistent results: for all $m < n$, applying C_n to input $|x\rangle$ of size m must give the same result as applying C_m to the same input

A quantum circuit family $\mathcal{C}$ is polynomially uniform if there exists a polynomial $f(n)$ and a classical program that constructs the circuit C_n in at most $O(f(n))$ steps

Query complexity

The first quantum algorithms solve “black box” or “oracle” problems, where it is only possible to solve the problem by observing the output of the black box

Query complexity

The first quantum algorithms solve "black box" or "oracle" problems, where it is only possible to solve the problem by observing the output of the black box

A quantum black box behaves like the transformation U_f

Query complexity

The first quantum algorithms solve "black box" or "oracle" problems, where it is only possible to solve the problem by observing the output of the black box

A quantum black box behaves like the transformation U_f

$$U_f : \sum_x \alpha_x |x\rangle|y\rangle \quad \longrightarrow \quad \sum_x \alpha_x |x, f(x) \oplus y\rangle$$

Query complexity

The first quantum algorithms solve "black box" or "oracle" problems, where it is only possible to solve the problem by observing the output of the black box

A quantum black box behaves like the transformation U_f

$$U_f : \sum_x \alpha_x |x\rangle|y\rangle \quad \longrightarrow \quad \sum_x \alpha_x |x, f(x) \oplus y\rangle$$

The query complexity is defined as the number of times that the black box must be queried to solve the problem

Query complexity

The first quantum algorithms solve "black box" or "oracle" problems, where it is only possible to solve the problem by observing the output of the black box

A quantum black box behaves like the transformation U_f

$$U_f : \sum_x \alpha_x |x\rangle|y\rangle \quad \longrightarrow \quad \sum_x \alpha_x |x, f(x) \oplus y\rangle$$

The query complexity is defined as the number of times that the black box must be queried to solve the problem

If the query complexity of a black box is low, it is only of utility if its implementation is efficient, however, this approach is useful in setting lower bounds on the circuit complexity

Query complexity

The first quantum algorithms solve "black box" or "oracle" problems, where it is only possible to solve the problem by observing the output of the black box

A quantum black box behaves like the transformation U_f

$$U_f : \sum_x \alpha_x |x\rangle|y\rangle \quad \longrightarrow \quad \sum_x \alpha_x |x, f(x) \oplus y\rangle$$

The query complexity is defined as the number of times that the black box must be queried to solve the problem

If the query complexity of a black box is low, it is only of utility if its implementation is efficient, however, this approach is useful in setting lower bounds on the circuit complexity

If the query complexity is $\Omega(N)$, then the circuit complexity must be at least $\Omega(N)$

Query complexity

The first quantum algorithms solve "black box" or "oracle" problems, where it is only possible to solve the problem by observing the output of the black box

A quantum black box behaves like the transformation U_f

$$U_f : \sum_x \alpha_x |x\rangle|y\rangle \quad \longrightarrow \quad \sum_x \alpha_x |x, f(x) \oplus y\rangle$$

The query complexity is defined as the number of times that the black box must be queried to solve the problem

If the query complexity of a black box is low, it is only of utility if its implementation is efficient, however, this approach is useful in setting lower bounds on the circuit complexity

If the query complexity is $\Omega(N)$, then the circuit complexity must be at least $\Omega(N)$

The value of black box problems in quantum computing was to demonstrate that a quantum algorithm has lower query complexity than a classical circuit that solves the same problem

Communication complexity

For communication problems, a complexity measure is the miniumum number of qubits that must be transmitted to accomplish a task

Communication complexity

For communication problems, a complexity measure is the miniumum number of qubits that must be transmitted to accomplish a task

For example in the dense coding algorithm, complexity is related to the number of qubits that must be sent in order to communicate n bits of information

Communication complexity

For communication problems, a complexity measure is the miniumum number of qubits that must be transmitted to accomplish a task

For example in the dense coding algorithm, complexity is related to the number of qubits that must be sent in order to communicate n bits of information

Classical protocols require the transmission of n bits while $n/2$ qubits plus an additional $n/2$ EPR pairs (or ebits) are needed for a quantum protocol

Communication complexity

For communication problems, a complexity measure is the miniumum number of qubits that must be transmitted to accomplish a task

For example in the dense coding algorithm, complexity is related to the number of qubits that must be sent in order to communicate n bits of information

Classical protocols require the transmission of n bits while $n/2$ qubits plus an additional $n/2$ EPR pairs (or ebits) are needed for a quantum protocol

For quantum teleportation of n qubits, the number of classical bits sent is $2n$ plus an additional n ebits