Notes on the implementation of the Hidden Shift Algorithm

Sebastián Grijalva

github.com/sebgrijalva

These are some calculations that give more detail about how the implementation of the Hidden Shift Algorithm works.

1 Introduction

Hidden Shift Algorithms are interesting, the example given is written in Cirq.

1.1 Problem Statement and Results

Consider the set of N-bit strings $\{0,1\}^N$. Let f and g be two function oracles $f, g : \{0,1\}^N \to \{0,1\}^N$, which are the same up to a hidden bit string $s \in \{0,1\}^N$ such that $g(x) = f(x \oplus s)$. The Hidden Shift Algorithm determines s by quering the two oracles. The implementation in the example considers the following definition (called *bent* function):

$$f(x) = \sum_i x_{2i-1}x_{2i} \tag{1}$$

where x_i is the i-th bit of x. While a classical algorithm requires $\sim 2^{N/2}$ queries, the Hidden Shift Algorithm solves the problem in $O(1)$ steps. We thus have an *exponential* reduction.

2 Application of Quantum Gates to the initial state

2.1 Description of the Algorithm

Let us call the initial state $|0^N\rangle$ by which we will mean the tensor product $|0\rangle \otimes \cdots \otimes |0\rangle$, N times. After application of the first Hadamard set of gates, we create the superposition

$$H^N|0^N\rangle = \frac{1}{\sqrt{2}^N} \sum_{x\in\{0,1\}^N} |x\rangle$$

Now we apply the shift s by means of a set of X gates representing the bit-string s at each position of the circuit (the flip induced by the X gate is equivalent to the modulo-2 addition that represents the action of the shift). By straightforwardly using the linearity of this operation, we obtain the state

$$\frac{1}{\sqrt{2}^N} \sum_{x\in 0,1^N} |x \oplus s\rangle$$

At this point we apply the oracle function (1) by means of a series of Controlled-Z gates: To see this, consider first the action of CZ on two succesive sites $a, a+1$:

$$\mathsf{CZ}(a,a+1)|\cdots x_a x_{a+1}\cdots\rangle = (-1)^{x_a x_{a+1}}|\cdots x_a x_{a+1}\cdots\rangle$$

indeed, if x_a is "active" (i.e. 1), the phase of the state becomes the eigenvalue of $|\cdots x_{a+1}\cdots\rangle$. Applyting this to each pair of channels, we reproduce the function *on the phases* of the amplitudes of each state $|x\oplus s\rangle$:

$$\frac{1}{\sqrt{2}^N}\sum_{x\in\{0,1\}^N}(-1)^{f(x\oplus s)}|x\oplus s\rangle$$

Notice that this sum can be equivalently translated over the shift s. More explicitly, for any function κ:

$$\sum_{x\in\{0,1\}^N}\kappa(x\oplus s)|x\oplus s\rangle = \sum_{\substack{x\oplus s\\ x\in\{0,1\}^N}}\kappa(x\oplus s\oplus s)|x\oplus s\oplus s\rangle = \sum_{x\in\{0,1\}^N}\kappa(x)|x\rangle$$

This means that our state can be written simply as

$$\frac{1}{\sqrt{2}^N}\sum_{x\in\{0,1\}^N}(-1)^{f(x)}|x\rangle$$

We apply a shift s set of gates again (recall that this is not an operation on the phases) to arrive at $\frac{1}{\sqrt{2}^N}\sum_{x\in 0,1^N}(-1)^{f(x)}|x\oplus s\rangle$. At this point we apply another full set of Hadamard gates:

$$\frac{1}{\sqrt{2}^{2N}}\sum_{x\in 0,1^N}(-1)^{f(x)}\sum_{y\in\{0,1\}^N}(-1)^{x\oplus s\cdot y}|y\rangle$$

rearranging the sum we get:

$$\frac{1}{\sqrt{2}^{2N}}\sum_{x,y\in 0,1^N}(-1)^{f(x)+x\cdot y}(-1)^{y\cdot s}|y\rangle$$

we make now the following definition:

$$\hat{F}(y)\equiv\frac{1}{\sqrt{2}^N}\sum_{x\in\{0,1\}^N}(-1)^{f(x)+x\cdot y} \tag{2}$$

Applying this to our state, we obtain:

$$\frac{1}{\sqrt{2}^{2N}}\sum_{x,y\in 0,1^N}(-1)^{f(x)+x\cdot y}(-1)^{y\cdot s}|y\rangle = \frac{1}{\sqrt{2}^N}\sum_{x,y\in 0,1^N}\hat{F}(y)(-1)^{y\cdot s}|y\rangle$$

Then we have the following result:

Lemma 1. *Let F be defined with (1) as $F(y)=(-1)^{f(y)}$, and $\hat{F}$ as in (2). Then:*

$$F(y)\hat{F}(y)=1 \tag{3}$$

Proof. Notice that N should be even to make f well defined. First consider the case where $N=2$:

$$\begin{aligned} F(y)\hat{F}(y) &= \tfrac{1}{\sqrt{2}^2}\sum_{x\in\{00,01,10,11\}}(-1)^{y_1y_2}(-1)^{x_1x_2}(-1)^{x_1y_1+x_2y_2} \\ &= \tfrac{(-1)^{y_1y_2}}{2}\Big(1+(-1)^{y_1}+(-1)^{y_2}-(-1)^{y_1+y_2}\Big) \end{aligned}$$

we see from here that for $y \in \{00, 01, 10, 11\}$, the right hand side gives 1. Now, the idea is to decompose the general sum two qubits at a time:

$$\begin{aligned}
F(y)\hat{F}(y) &= \tfrac{1}{\sqrt{2}^N} \sum_{x \in \{0,1\}^N} (-1)^{\sum_i y_{2i-1}y_{2i}} (-1)^{\sum_i x_{2i-1}x_{2i}} (-1)^{\sum_i x_i y_i} \\
&= \tfrac{1}{\sqrt{2}^N} \sum_{x \in \{0,1\}^N} \prod_i (-1)^{y_{2i-1}y_{2i}} (-1)^{x_{2i-1}x_{2i}} (-1)^{x_i y_i} \\
&= \prod_{i=1}^{N} \left\{ \frac{1}{\sqrt{2}^2} \sum_{x_i \in \{0,1\}^2} (-1)^{y_{2i-1}y_{2i}} (-1)^{x_{2i-1}x_{2i}} (-1)^{x_i y_i} \right\}
\end{aligned}$$

The last line is a product of terms that we have seen give each 1.

□

By this lemma we see that an application of the function f via Controlled-Z gates transforms the state into:

$$\frac{1}{\sqrt{2}^N} \sum_{x,y \in 0,1^N} F(y)\hat{F}(y)(-1)^{y \cdot s} |y\rangle = \frac{1}{\sqrt{2}^N} \sum_{y \in 0,1^N} (-1)^{y \cdot s} |y\rangle = H^N |s\rangle$$

Finally, we apply once more a full set of Hadamards to return to the desired s shift state (since the inverse of a Hadamard is another Hadamard).

References

[1] Wim van Dam, Sean Hallgreen, Lawrence Ip Quantum Algorithms for some Hidden Shift Problems.

[2] K Wrigth, et. a. Benchmarking an 11-qubit quantum computer. *Nature Communications*, 107(28):12446–12450, 2010. doi:10.1038/s41467-019-13534-2.