RESEARCH ARTICLE | OCTOBER 28 2021

Quantum computing without quantum computers: Database search and data processing using classical wave superposition

Mykhaylo Balynsky ; Howard Chiang ; David Gutierrez; Alexander Kozhevnikov ; Yuri Filimonov ; Alexander Khitun

Journal of Applied Physics 130, 164903 (2021)
https://doi.org/10.1063/5.0068316
CHORUS

Journal of Applied Physics ARTICLE scitation.org/journal/jap

Quantum computing without quantum computers: Database search and data processing using classical wave superposition

Cite as: J. Appl. Phys. 130, 164903 (2021); doi: 10.1063/5.0068316
Submitted: 24 August 2021 · Accepted: 6 October 2021 ·
Published Online: 28 October 2021

View Online Export Citation CrossMark

Mykhaylo Balynsky,[1] Howard Chiang,[1] David Gutierrez,[1] Alexander Kozhevnikov,[2] Yuri Filimonov,[2,3] and Alexander Khitun[1,a)]

AFFILIATIONS

[1]Department of Electrical and Computer Engineering, University of California-Riverside, Riverside, California 92521, USA
[2]Kotelnikov Institute of Radioengineering and Electronics of the Russian Academy of Sciences, Saratov 410019, Russia
[3]Department of Physical and Mathematical Sciences, Saratov State Technical University, Saratov State University, Saratov 410012, Russia

[a)]Author to whom correspondence should be addressed: akhitun@engr.ucr.edu

ABSTRACT

Quantum computers are proven to be more efficient at solving a specific class of problems compared to traditional digital computers. Superposition of states and quantum entanglement are the two key ingredients that make quantum computing so powerful. However, not all quantum algorithms require quantum entanglement (e.g., search through an unsorted database). Is it possible to utilize classical wave superposition to speed up database searching as much as by using quantum computers? There were several attempts to mimic quantum computers using classical waves. It was concluded that the use of classical wave superposition comes with the cost of an exponential increase in resources. In this work, we consider the feasibility of building classical wave-based devices able to provide fundamental speedup over digital counterparts without the exponential overhead. We present experimental data on database searching through a magnetic database using spin wave superposition. The results demonstrate the same speedup as expected for quantum computers. Also, we present examples of numerical modeling demonstrating classical wave interference for period finding. This approach may not compete with quantum computers with efficiency but outperform classical digital computers. We argue that classical wave-based devices can perform some of the quantum algorithms with the same efficiency as quantum computers as long as quantum entanglement is not required.

INTRODUCTION

Quantum computing is an emerging field that will eventually lead us to new and powerful logic devices with capabilities far beyond the limits of current transistor-based technology.[1] There are certain types of problems that quantum computers can solve fundamentally faster than traditional digital computers. For example, Peter Shor developed a quantum algorithm that solves factoring and discrete logarithm problems in time $O(n^3)$ compared with the exponential time required for the best known classical algorithms.[2] Quantum computers can search an "unsorted database" (that is, for $f(x):\{0, N\} \to \{0, 1\}$, find x_0 such that $f(x_0) = 1$) in time $O(\sqrt{N})$ compared with the O(N) time that would be required classically.[3] Superposition of states and entanglement are the two key ingredients that make quantum computing so powerful. Superposition of states allows us to speed up database searching by checking the number of bits in parallel, while quantum entanglement is critically important for quantum cryptography.[4] There are quantum algorithms that require both superposition and entanglement (e.g., Shor's algorithm), but neither the Grover algorithm, nor the very first quantum algorithm due to Deutsch and Jozsa[5] need entanglement for search speedup.[6,7] Is it possible to utilize classical wave superposition to speed up database search? This interesting question was analyzed by Seth Lloyd.[7] It was concluded that classical devices that rely on wave interference may provide the same speedup over classical digital devices as quantum devices.[7] Patel came up with the same conclusion on the efficiency of using classical wave superposition for database searching.[8] The idea of using classical interferometers for quantum-like data processing has been discussed in a number of works.[9–11] There were several experimental works using

optical beam superposition for emulating Grover's algorithm.[12–14] The optical approach has several appealing advantages including ultimate speed of information transfer and low power consumption. It is orders of magnitudes more efficient compared to electronic digital devices used for classical wave simulations.[15] The main problem is associated with qubit representation. For instance, it was proposed to use multiple paths of the interferometer for state encoding,[12] which led to an exponential increase in the paths with the number of qubits. It was concluded that the use of classical wave superposition comes with the cost of exponential increase in required resources.[12] Since then, it is widely believed that the use of classical wave superposition for quantum algorithms is inevitably leading to an exponential resource overhead (e.g., the number of devices, power consumption, and precision requirements). In this work, we describe a classical Oracle machine that utilizes classical wave superposition for database searching and data processing. The novelty of this approach is in the logic state encoding, where logic 0 and 1 are assigned to phases (e.g., phase 0π corresponds to logic 0, phase π/2 corresponds to logic 1). Phase coding allows us to exploit wave superposition for parallel database searching. This approach is purely deterministic in contrast to quantum or other techniques (e.g., Simulated Annealing[16]). We present experimental data on magnetic database searching using spin wave superposition. The data show a fundamental speedup over digital computers without any exponential resource overhead. We argue that in some cases, the classical wave-based approach may provide the same speedup in database searching as quantum computers. There are also examples of numerical modeling demonstrating classical wave interference for period finding. This approach may not compete with quantum computers in efficiency but outperform classical digital computers. There is a lot of room for classical wave-based computer development that may provide a fundamental advantage over classical digital devices. These classical wave-based devices can perform some algorithms with the same efficiency as quantum computers as long as quantum entanglement is not required. The rest of the work is organized as follows. In section "Methods," we describe the structure of classical Oracle and its principle of operation. Next, we present several examples of database searching and period finding. The advantages and limitations of classical wave computing are provided in the Discussion Section.

METHODS

Let us consider a classical Oracle machine—Oracle-C as illustrated in Fig. 1. It consists of an interferometer, a non-linear detector, and a general-purpose digital computer. The interferometer is a multi-terminal device that has n inputs ports and just one output port. The input and output signals are classical sinusoidal waves (e.g., electrical, optical, sound, etc.) given by

$$y(t) = A_i \sin(\omega t + f_i), \quad (1)$$

where A_i is the amplitude, ω is the frequency, t is the time, and f_i is the phase. The subscript i defines the input number. All input waves have the same frequency ω and amplitude A_0. Information is encoded into the phases of the input waves f_i as follows:

$$\text{Input logic state (ith input)} = \begin{cases} 0, & \text{if } f_i = \pi/2 \\ 1, & \text{if } f_i = 0/\pi \end{cases} : \quad (2)$$

The output is a result of n wave interference,

$$A_{out} \sin(\omega t + f_{out}) = \sum_{i=1}^{n} \sigma_i A_i \sin(\omega t + f_i + \Delta_i), \quad (3)$$

where σ_i is the amplitude change and Δ_i is the accumulated phase shift during wave propagation from the ith input port to the output port. Hereafter, we consider a linear wave propagation (i.e., σ_i and

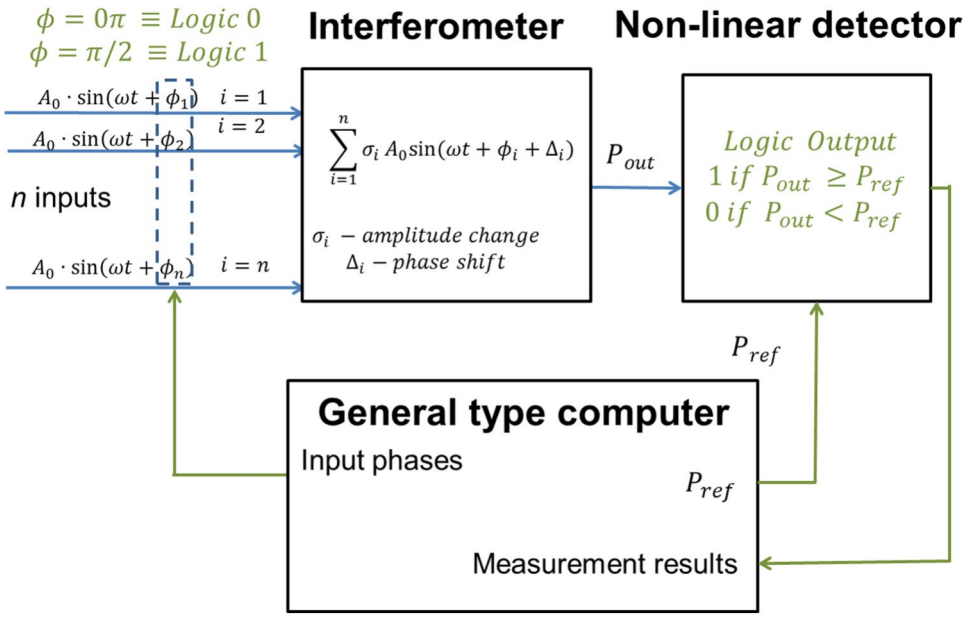


FIG. 1. Schematic view of the Classical Oracle machine—Oracle-C. It consists of a multi-input interferometer, a non-linear detector, and a general-purpose digital computer. Input information is encoded into the phases of classical sinusoidal waves f_i. The output of the interferometer is a wave—the result of interference of all input waves. The output logic state is defined by comparing the output power with a reference value P_{ref}. The digital computer controls the input phases, the reference value P_{ref}, and stores the results of measurement.

Δ_i do not depend on the wave amplitude). The non-linear detector detects the time-averaged power P_{out} and compares it to some reference value P_{ref}. The output logic state is 1, if P_{out} is equal or exceeds P_{ref}, and 0 otherwise,

$$\text{Output logic state(output)} = \begin{cases} 1, & \text{if } P_{out} \geq P_{ref}, \\ 0, & \text{if } P_{out} < P_{ref}. \end{cases} \quad (4)$$

The digital computer is aimed to control the input phases f_i, store the results of measurements, and control the reference value P_{ref}. Oracle-C provides a deterministic input–output correlation. It constitutes a database $f(x):\{0, n\} \to \{0, 1\}$, where the correlation between input and output logic states is defined by the internal structure of the interferometer (i.e., σ_i and Δ_i). Our objective is to show that using classical wave superposition it is possible to speed up database searching similar to the algorithms developed for quantum computers. In our case, the superposition of logic 0 and 1 is a wave with phase π/4,

$$\begin{aligned} &\alpha|0\rangle + \beta|1\rangle; \quad \sin(\omega t) + \cos(\omega t) \\ &\sin(\omega t) + \sin(\omega t + \pi/2) = \sqrt{2}\sin(\omega t + \pi/4). \end{aligned} \quad (5)$$

Next, we will present several examples of using Oracle-C. In example 1, we show a search through an "unsorted database," where only one input phase combination leads to logic output 1. In example 2, the search algorithm will be extended to a database with multivalued inputs. In example 3, we present experimental data on searching through a magnetic database using spin wave superposition. Finally, in example 4, we will demonstrate the results of numerical modeling showing period finding of a given function using Oracle-C.

Example 1: Searching through an "unsorted database"

Let us consider Oracle-C constructed in such a way that only one input combination results in the output logic state 1: $f(x_0) \to \{0, 1\}$, where x_0 is an input phase combination $x_0 \in \{f_1, f_2, \ldots, f_n\}$. For instance, there is only one input phase combination resulting in wave interference at the output (i.e., all waves are coming in phase to the output). All other input phase combinations result in a lower output power. The task is to find this phase combination in the minimum number of steps. This task is similar to finding a name in a phone book by giving a phone number. In our case, a phone number is a phase combination and the interferometer is the phone book. There are several assumptions we make regarding the internal interferometer structure: (i) All input waves have the same amplitude A_0. (ii) The amplitudes do not change during wave propagation within the interferometer (i.e., $\sigma = 1$ for all inputs). (iii) The phase shifters Δ_i may provide either a $\pm\pi/4$ or a π/4 phase shift. There are 2^n possible phase combinations. There are 2^n ways to choose a set of Δ_i: However, there are $2^n - 2$ shifter combinations leading to only one constructive interference output. Two combinations with all $\Delta_i = \pm\pi/4$ or π/4 should be excluded from consideration. The reference value in the detector is set up to $P_{ref} = nP_0$, where P_0 is the power provided by just one input. Thus, there is only one input phase combination that results in constructive wave interference providing logic output 1 for the given set of Δ_i. It would take 2^{n-1} queries on average to accomplish this task taking all possible input combinations one by one. It takes only 2n queries using classical wave superposition. The solving procedure is as follows:

In step 1, we find the right phase for input 1. All inputs from i = 2 to i = n are set up in a superposition of states (i.e., by exciting input waves with a phase π/4). First, we measure the output for input combination $\{0\pi, \pi/4, \pi/4, \ldots, \pi/4\}$. This input phase combination is equivalent to logic state $\left\{0, \frac{0+\pi}{2}, \frac{0+\pi}{2}, \ldots \frac{0+\pi}{2}\right\}$. The output P_{out} is memorized by the digital computer. Second, we measure the output for input combination $\{\pi/2, \pi/4, \pi/4, \ldots, \pi/4\}$. This input phase combination is equivalent to logic state $\left\{1, \frac{0+\pi}{2}, \frac{0+\pi}{2}, \ldots, \frac{0+\pi}{2}\right\}$. Then, we compare the output powers for the two measurements. The "right" phase combination always provides a larger output. The comparison procedure is as follows: The reference value P_{ref} is set up to P_{out} measured for $\left\{0, \frac{0+\pi}{2}, \frac{0+\pi}{2}, \ldots, \frac{0+\pi}{2}\right\}$. The right phase for input i = 1 is π/2 (i.e., logic 1) if $P_{out}\left\{1, \frac{0+\pi}{2}, \frac{0+\pi}{2}, \ldots, \frac{0+\pi}{2}\right\}$ is larger than $P_{out}\left\{0, \frac{0+\pi}{2}, \frac{0+\pi}{2}, \ldots, \frac{0+\pi}{2}\right\}$. The right phase is 0π (i.e., logic 0) otherwise.

In step 2, the phase shifter at input i = 1 is fixed to the value found in step 1. All inputs from i = 3 to i = n are in the superposition of states. We make two measurements with two possible phases for input 2. The one with a larger output power corresponds to the right phase. The procedure is repeated n times until the complete input phase combination is found. Finally, one can apply the obtained phase combination with $P_{ref} = nP_0$ to verify the answer (i.e., to confirm that the logic output is 1).

A simple algorithmic pseudocode is as follows:

Set all input phases to π/4.

Start with input i = 1.

Consider two possible phases: 0π and π/2.

Measure P_{ref}—output power for phase combination $\left\{0, \frac{\pi}{4}, \ldots, \frac{\pi}{4}\right\}$.

Measure P_{out}—output power for phase combination $\left\{\frac{\pi}{2}, \frac{\pi}{4}, \ldots, \frac{\pi}{4}\right\}$.

$$\text{Phase (i)} = 0 \text{ if } P_{ref} \geq P_{out}:$$

$$\text{Phase(i)} = \frac{\pi}{2}, \text{ otherwise:}$$

Check the code for errors if $P_{ref} = P_{out}$.

Continue for all remaining inputs i = 2, n.

The search procedure for Oracle-C with seven inputs n = 7 is illustrated in Fig. 2. The interferometer has the following set of phase shifters $\Delta_i\{\pi/4, \pi/4, \pi/4, \pi/4, \pi/4\}$. There is just one input phase combination $\{0\pi, \pi/2, \pi/2, 0\pi, 0\pi\}$, which results in constructive interference at the output. This phase combination was found in seven steps (14 measurements) according to the above-described procedure. The three tables in Fig. 2(a) show the

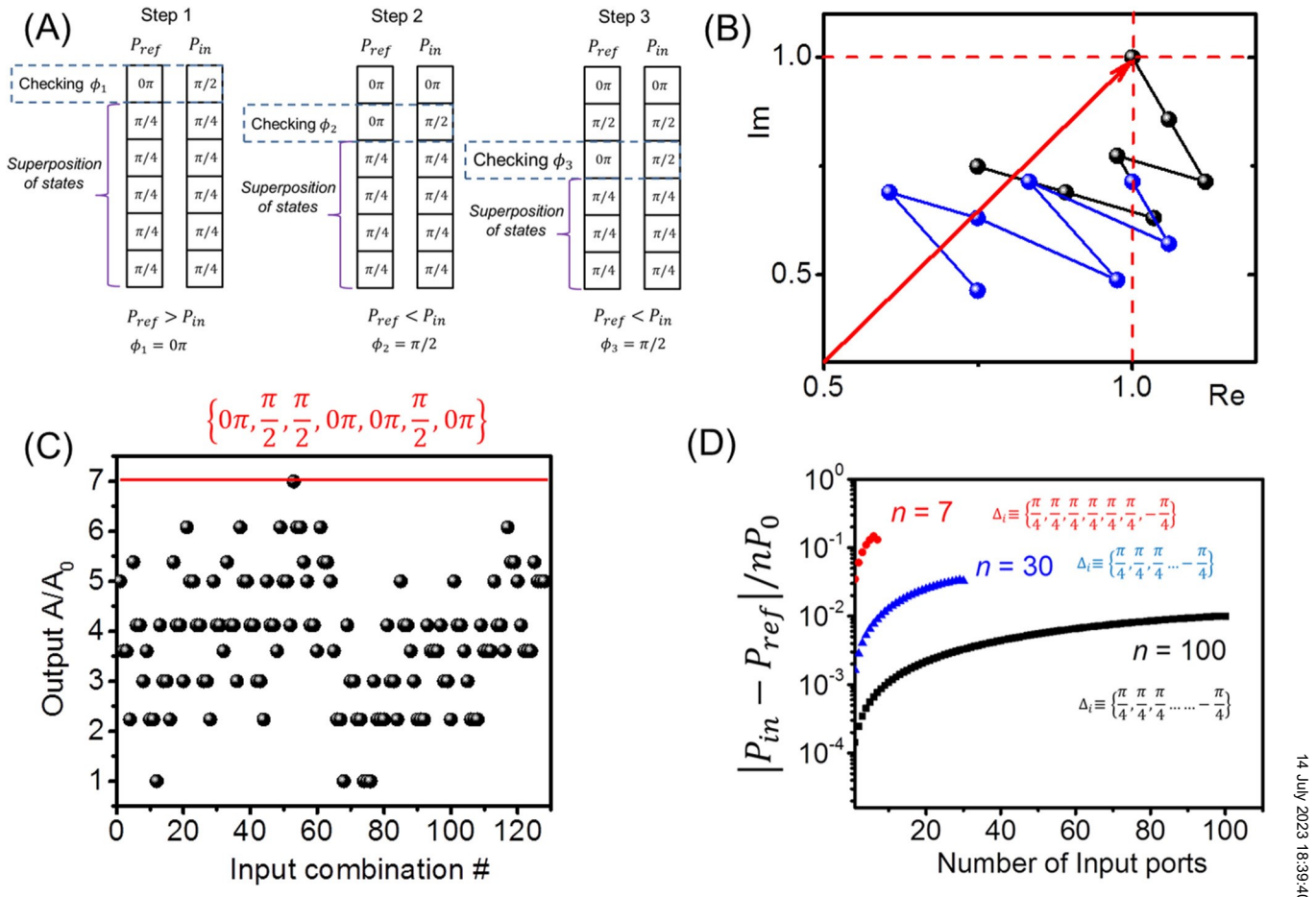


FIG. 2. (a) Illustration of the search procedure. There are shown the first three consecutive steps. At the first step, all inputs from 2 to n are put into a superposition of states. There are two measurements to determine the "true" value of input 1. One measurement is 0 > + superposition and the other is 1 > + superposition. The "true" input always provides the larger output power. It takes n steps with 2n measurements to find the one of the 2^n possible input combinations. (b) Results of numerical modeling showing the evolution of the evolution of the output signal y_{out} as a vector in a polar form for seven consequent steps. The black marker corresponds to the "true" phase input, while the blue marker corresponds to the "false" input phase. The red vector corresponds to the searched phase combination. (c) Results of numerical simulations showing the output amplitude for all possible input phase combinations. The amplitude is normalized to the amplitude of a single input A_0. (d) Results of numerical model-ing showing the minimum difference between the "true" and the "false" power output. The red, the blue, and the black curves correspond to 7-, 30-, and 100 input Oracle-C.

input phase combinations for the first three steps. The numerical values for P_{out} and P_{ref} for all steps are summarized in the supplementary material. The evolution of the output signal y_{out} for the seven consequent steps is shown in Fig. 2(b),

$$y_{out} = A_m \sin(\omega t + f_{out}), \quad (6)$$

where A_m is the maximum amplitude and f_{out} is the phase of the output signal. It is convenient to express y_{out} as a phasor in polar form, where $Re(y_{out}) = A_m \cos(f_{out})$ and $Im(y_{out}) = A_m \sin(f_{out})$. The red vector in Fig. 2(b) depicts the output signal corresponding to the searched phase combination. It corresponds to the constructive wave interference when all n input waves reach the output in-phase: $f_{out} = \pi/4$ and $A_m = nA_0$. The black and blue markers in Fig. 2(b) show the evolution of the output signal during measurements. The black marker corresponds to the "true" phase input, while the blue marker corresponds to the "false" input phase. There are seven black and blue markers corresponding to the results of 14 consequent measurements. The phase combination "true" + superposition appears closer to the correct result (i.e., the red vector) compared to the "false" + superposition combination at each step. The seventh black marker coincides with the correct phase combination. To verify the result of the search procedure for the seven-input Oracle-C, all phase combinations were checked one-by-one. The graph in Fig. 2(c) shows A_{out} normalized to A_0 for $2^7 = 128$ input phase combinations. Indeed, the phase

combination found in 14 measurements provides the maximum output corresponding to constructive wave interference.

The above procedure is based on the comparison between the "true" + superposition and "false" + superposition phase combinations. In quantum computing, "true" + superposition and "false" + superposition are usually noted as tensor product (e.g., jf_1jf_n). Hereafter, we explain results in plain language, which may be more comprehensive for non-experts in quantum computing. The "true" + superposition phase combination always provides a larger output power as it is closer to constructive interference. There is no exponential overhead in terms of number of devices or power. The required accuracy of measurements does scale with the number of inputs. The difference in the output power between the "true" + superposition and "false" + superposition combinations depend on the set of phase shifters Δ_i. Accuracy is at its minimum (i.e., the worst scenario) when n 1 shifters are the same (e.g., all $\beta\pi/4$ or all $\pi/4$) and only one phase shifter is different from the others. In Fig. 2(d), the difference is plotted as a function of the number of inputs. The red, blue, and black curves depict the normalized difference between the "true" and "false" phase combinations for n ¼ 7, n ¼ 30, and n ¼ 100. In all three cases, the first n 1 phase shifters are set to $\beta\pi/4$ and only the last one is set to $\pi/4$. Maximum accuracy is required for the first measurements. Though the difference between the "true" and "false" phase combinations does decrease with the number of inputs n, it appears acceptable even for n ¼ 100. In Fig. 3, the minimum difference between the "true" and the "false" outputs as a function of the number of input ports is shown. We model the case when only one of the phase shifters is different from the others (i.e., the worst scenario leading to the minimum power difference). The power difference decreases with the number of inputs. The blurry markers for n ⍰ 1000 are due to the limited accuracy of the regular digital computer.

The search algorithm was performed for n ¼ 100 as well. It was found that an input combination leads to the output logic 1 for the given Oracle-C structure. However, it is not practically possible to verify whether this is the only result as it would take 2^{100} operations to check all possible input phase combinations one by one.

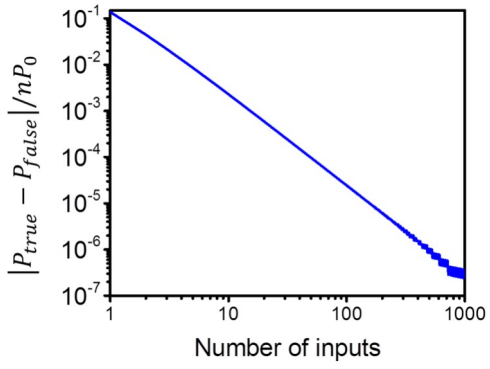


FIG. 3. Results of numerical modeling for multi-port Oracle-C showing the difference between the output power for "true" and "false" input phase combinations. The modeling is accomplished for the case when only one of the phase shifters is different from the others (i.e., the worst scenario leading to the minimum power difference).

The superposition approach can be extended to a multivalued case. In this scenario, multiple logic states (e.g., 0, 1, 2, 3, . . . , m) can be encoded into different phases (e.g., f(0), f(1), f(2), . . . , f(m)). In the next example, we demonstrate a search procedure for finding the only input (i.e., phase combination) leading to constructive wave interference out of m^n possible combinations.

Example 2: Search through Oracle-C with multivalued inputs

Let us consider a three-input n ¼ 3 Oracle-C with eight possible logic states at each input m ¼ 8. The input logic states are encoded into eight phases $0\pi, \frac{\pi}{14}, \frac{2\pi}{14}, \frac{3\pi}{14}, \frac{4\pi}{14}, \frac{5\pi}{14}, \frac{6\pi}{14}, \pi/2$. The phase shifter Δ_i in Oracle-C may be any of $0\pi, \frac{\pi}{14}, \frac{2\pi}{14}, \frac{3\pi}{14}, \frac{4\pi}{14}, \frac{5\pi}{14}, \frac{6\pi}{14}, \pi/2$ excluding only the same values for all three shifters (i.e., there are no combinations $\Delta_1 = \Delta_2 = \Delta_3$). Thus, Oracle-C is constructed in such a way that only one input phase combination results in the constructive interference (logic output = 1 at $P_{ref} = 3P_0$). The task is to find this phase combination in the minimum number of steps.

The search procedure is illustrated in Fig. 4. The whole number of possible phase combinations constitutes a cube in three-dimensional space. The x, y, and z axes correspond to the three input phases phase 1, phase 2, and phase 3, respectively. There are eight possible values for each phase. The total number of possible phase combination is $8^3 = 512$. Instead of checking all these combinations one by one, we divide the values on each axis into two halves and apply wave superposition to check phase combinations in each segment. Similar to example 1, the output of the phase segment with the true phase combination is always larger compared to the other segments.

In step 1, we divide the whole phase space into segments. There are eight phase segments for a three-dimensional space. For example, segment 1 in Fig. 3 includes all input combinations with f_1 [$0\pi, \frac{\pi}{14}, \frac{2\pi}{14}, \frac{3\pi}{14}$, f_2 [$0\pi, \frac{\pi}{14}, \frac{2\pi}{14}, \frac{3\pi}{14}$, and f_3 [$0\pi, \frac{\pi}{14}, \frac{2\pi}{14}, \frac{3\pi}{14}$. Segment 2 includes all input combinations with f_1 [$0\pi, \frac{\pi}{14}, \frac{2\pi}{14}, \frac{3\pi}{14}$, f_2 [$0\pi, \frac{\pi}{14}, \frac{2\pi}{14}, \frac{3\pi}{14}$, and f_3 [$\frac{4\pi}{14}, \frac{5\pi}{14}, \frac{6\pi}{14}, \frac{\pi}{2}$. Next, we provide eight measurements with phase combinations corresponding to the superposition of states in each segment. The phase combination for segment 1 is $\frac{\pi}{8}, \frac{\pi}{8}, \frac{\pi}{8}$. The phase combination for segment 2 is $\frac{\pi}{8}, \frac{\pi}{8}, \frac{3\pi}{8}$ and so on. The results of the measurements are summarized in the tables in Fig. 4. The first table shows the output amplitudes for the eight phase combinations. The largest amplitude of 2.846 75 A_0, where A_0 is the output of a single input, is detected for segment 2. The output amplitude is larger for the segment containing the searched phase combination. The reason is the same as illustrated in Fig. 2(b). The superposition of waves in the right segment is closer to constructive superposition.

In Step 2, we take phase segment 2 that provides the maximum output, divide it into eight sub-segments, and provide

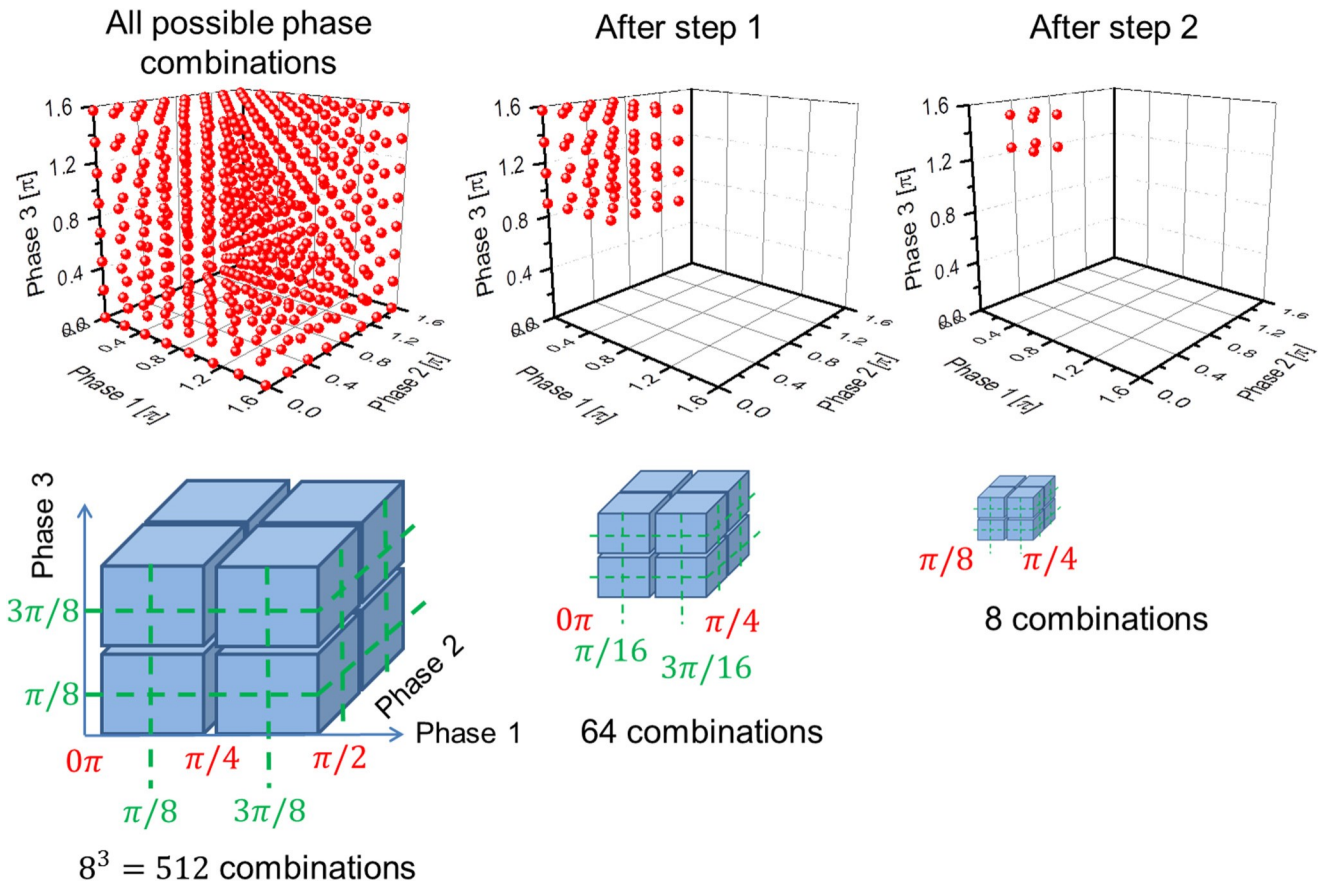


FIG. 4. Illustration of the search procedure in Oracle-C with multivalued inputs. It is considered a three-input Oracle-C with eight possible phase combinations per input. The internal structure of the Oracle is setup in such a way that only one phase combination provides output 1. The task is to find this phase combination in the minimum number of steps. In the first step, the whole space of all possible phase combinations is divided into eight segments. The output of each of the segments is obtained using wave superposition. The segment with the "right" phase combination provides the largest output. In the second step, the search is accomplished in the segment found in step 1. Finally, the last eight combinations are checked one by one. The right combination provides the maximum output (i.e., the constructive wave interference).

measurements for each sub-segment. The results of numerical simulations are summarized in Fig. 5. The maximum output amplitude corresponds to the phase segments containing the following phases: f_1 [$\frac{2\pi}{14}$, $\frac{3\pi}{14}$, f_2 [0π, $\frac{\pi}{14}$, and f_3 [$\frac{6\pi}{14}$, $\frac{7\pi}{14}$. There are eight phase combinations left out of the 512 possible. In the final step, we check all of the remaining phase combinations one by one. The maximum output amplitude of 3.0 A_0 corresponding to constructive interference occurs for the input phase combination $\frac{\pi}{7}$, 0π, $\frac{\pi}{2}$. One can check that only this input phase combinations provides the constructive output interference for $\Delta_1 = \frac{5\pi}{14}$; $\Delta_2 = \frac{\pi}{2}$; $\Delta_3 = 0\pi$.

The search was accomplished in 24 queries. It would take 256 queries on average for a digital computer. In general, the advantage of the superposition technique using phase space division over a classical digital machine is $O(\frac{p}{n}m)$, where n is the number of inputs (i.e., the dimension of phase space) and m is the number of states per each input (i.e., the number of phases per input). In example 2, we intentionally take n = 3 to have 3D space for all possible phase combinations. Without a loss of generality, m = 8 was taken as an example.

Example 3: Experimental data: Search through a magnetic database using spin wave superposition

In order to validate the practical value of the described search procedures, we present experimental data on magnetic database searching using spin wave superposition. Data centers based on magnetic storage technology have proved to be the core platforms for cloud computing and Big Data storage.[17,18] It has already stimulated a pursuit of novel and efficient methods for parallel magnetic bit read-out (e.g., multihead, multitrack magnetic memory[19]). In this part, we implement the search algorithm described in the previous examples to a magnetic database. In our preceding works,[20,21] we have developed magnonic holographic memory

(A) The results of the first search step

Phase at Input 1 [π]	Phase at Input 2 [π]	Phase at Input 3 [π]	Output amplitude [A_0]
1/8	1/8	1/8	2.38115
1/8	**1/8**	**3/8**	**2.84675**
1/8	3/8	1/8	1.764675
1/8	3/8	3/8	2.35552
3/8	1/8	1/8	2.05607
3/8	1/8	3/8	2.67766
3/8	3/8	1/8	1.65148
3/8	3/8	3/8	2.38115

(B) The results of the second search step

Phase at Input 1 [π]	Phase at Input 2 [π]	Phase at Input 3 [π]	Output amplitude [A_0]
1/16	1/16	5/16	2.84675
1/16	1/16	7/16	2.9405
1/16	3/16	5/16	2.64309
1/16	3/16	7/16	2.78277
3/16	1/16	5/16	2.81112
3/16	**1/16**	**7/16**	**2.95506**
3/16	3/16	5/16	2.65682
3/16	3/16	7/16	2.84675

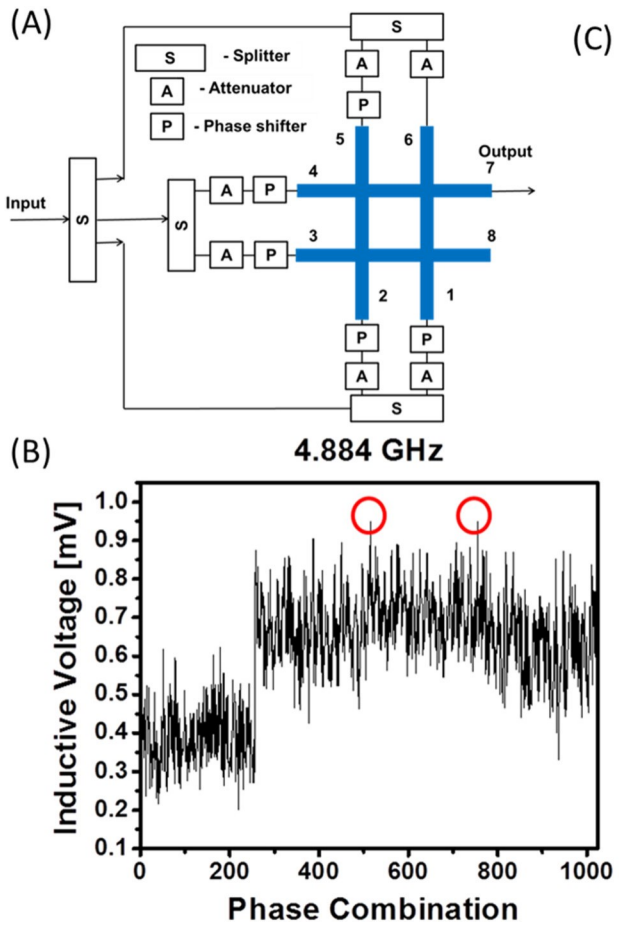
(C) The results of the third search step

Phase at Input 1 [π]	Phase at Input 2 [π]	Phase at Input 3 [π]	Output amplitude [A_0]
2/14	0	6/14	2.98324
2/14	**0**	**7/14**	**3.0**
2/14	1/14	6/14	2.94986
2/14	1/14	7/14	2.98324
3/14	0	6/14	2.94986
3/14	0	7/14	2.98324
3/14	1/14	6/14	2.93324
3/14	1/14	7/14	2.98324

FIG. 5. Results of numerical modeling on the search procedure in example 2. The first three columns show the input phases and the fourth column shows the output amplitude normalized to A_0, where A_0 is the output of a single input. Tables (A), (B), and (C) show the results for the first, second, and the third search steps, respectively.

(MHM) aimed at exploiting spin waves for parallel read-in and read-out. Spin waves—or magnons, the quanta of spin waves—represent eigen excitations of the electron spin subsystem in magnetically ordered media and are observed in ferro- and ferrimagnets as well as in antiferromagnets.[22] The interaction between the magnetic bits and propagating spin waves is the base of MHM operation. Examples of working prototypes are reported in Ref. 23. The schematic of an MHM device is shown in Fig. 6. It is a multi-input interferometer with a mesh of magnetic waveguides inside. Input spin waves are excited by the set of micro-antennas placed on top of waveguides (i.e., # 1–6). The output is the inductive voltage detected by the output antenna (i.e., # 7) produced by the interfering spin waves. The core of the MHM device is a mesh of magnetic waveguides made of material with low spin wave damping [e.g., $Y_3Fe_2(FeO_4)_3$ (YIG)]. There are magnets (e.g., Co) placed on the top of the waveguides. These are memory elements where information is encoded in the magnetization direction. The details of the structure preparation and measurement techniques can be found in Ref. 24. Spin waves propagating from different inputs to the output accumulate different phase shifts Δ_i that depend on the configuration of magnets in the mesh. The phase difference between the waves does not exceed $\pi/2$. The set of attenuators was used to equalize the amplitudes of the spin waves at the output port (i.e., σ_i are the same for all inputs). We consider a linear spin wave propagation at low input power (i.e., σ_i and Δ_i do not depend on the wave amplitude). Overall, the input–output correlation of the MHM device is well-described by Oracle-C as shown in Fig. 1. In our experiment, we use five input antennas [i.e., marked #1–5 in Fig. 6(a)] to provide input information, antenna #6 to provide a reference signal with constant phase, and antenna #7 to pick up the inductive voltage. Four distinct phases 0°, 7°, 14°, and 21° are used for each antenna. These four phases were arbitrary chosen for the test experiment with only the condition that the maximum phase difference does not exceed $\pi/2$. There are $4^5 = 1024$ possible phase combinations. The task is to apply the search algorithm as described in example 2 and the find phase combination resulting in the maximum output voltage.

The ensemble of all phase combinations constitutes a cube in 5D space. Unfortunately, it cannot be visualized as in Fig. 4. In the first step, the whole phase space is divided into $2^5 = 32$ subsegments, where phases for each antenna are grouped into two halves: (0°, 7°) and (14°, 21°). We apply wave superposition in each of the segments (e.g., 4° or 8° for each antenna) and detect the inductive voltage. The experimental data are shown in Fig. 6(c).



(C)

Phase Combination [degrees]	Output Inductive Voltage [mV]	Phase Combination [degrees]	Output Inductive Voltage [mV]
4, 4, 4, 4, 4	0.77	18, 4, 4, 4, 4	0.89
4, 4, 4, 4, 18	0.87	18, 4, 4, 4, 18	0.95
4, 4, 4, 18, 4	0.67	18, 4, 4, 18, 4	0.73
4, 4, 4, 18, 18	0.87	18, 4, 4, 18, 18	0.83
4, 4, 18, 4, 4	0.74	18, 4, 18, 4, 4	0.80
4, 4, 18, 4, 18	0.78	18, 4, 18, 4, 18	0.86
4, 4, 18, 18, 4	0.64	18, 4, 18, 18, 4	0.68
4, 4, 18, 18, 18	0.70	18, 4, 18, 18, 18	0.78
4, 18, 4, 4, 4	0.77	18, 18, 4, 4, 4	0.82
4, 18, 4, 4, 18	0.87	18, 18, 4, 4, 18	0.90
4, 18, 4, 18, 4	0.72	18, 18, 4, 18, 4	0.78
4, 18, 4, 18, 18	0.74	18, 18, 4, 18, 18	0.86
4, 18, 18, 4, 4	0.75	18, 18, 18, 4, 4	0.74
4, 18, 18, 4, 18	0.85	18, 18, 18, 4, 18	0.87
4, 18, 18, 18, 4	0.60	18, 18, 18, 18, 4	0.67
4, 18, 18, 18, 18	0.72	18, 18, 18, 18, 18	0.82

(D)

	Classical Computer	Oracle -C
Number of queries	1024	64

FIG. 6. (a) Schematics of the experimental setup. It consists of a multi-port spin wave interferometer. There are six (#1–#6) input and one output (#7) antennas. Input signals are spin waves. The spin wave excited by the same microwave source. The set of attenuators and phase shifters is to control the amplitudes and the phases of the input signals. The output is inductive voltage produced by the interfering spin waves. (b) Experimental data. The plot shows the output (i.e., the inductive voltage in mV for all 1024 input phase com-binations). The red circles depict the maximum output voltage (i.e., the phase combinations we are looking for). (c) Tables with experimental data showing the input phase combina-tion and the output inductive voltage for step 1. The maximum output voltage corresponds to the phase segment in 5D space containing the searched phase combination(s). (d) The summary table showing the number of quires required for a classical digital computer and Oracle-C using wave superposition.

In step 2, we consider only one sub-segment that provides the maximum output voltage. There are 32 possible phase combinations left, which are checked one-by-one. Two phase combinations (21°, 0°, 0°, 0°, 21°) and (21°, 7°, 7°, 0°, 21°) that provide the highest output voltage are found. In order to verify these results of the superposition-based search, we acquired test measurements taking all 1024 phase combinations one by one. The results are shown in Fig. 6(b). The inductive voltage (vertical scale) in mV is shown for all phase combinations (horizontal scale). The phase combinations are numbered as follows: (0°, 0°, 0°, 0°, 0°) = 1, (0°, 0°, 0°, 0°, 7°) = 2, ... (21°, 21°, 21°, 21°, 21°) = 1024. There are two phase combinations (21°, 0°, 0°, 0°, 21°) (#515 on the graph) and (21°, 7°, 7°, 0°, 21°) (#755 on the graph), which provide 0.9501 and 0.9507 mV inductive voltage, respectively. Thus, the results of one-by-one measurements confirmed the results of the superposition-based search. Figure 6(d) summarizes the comparison between Oracle-C and a classical digital computer. It took 64 measurements with Oracle-C using superposition instead of 1024 subsequent measurements. Overall, examples 1–3 show an intriguing possibility of exploiting classical wave superposition for database search speedup.

It is worth mentioning the capabilities of the classical wave-based approach for prime factorization. Though this approach cannot compete with true quantum algorithms in efficiency as it does not exploit quantum entanglement, it may nonetheless provide a fundamental advantage over digital computers.

Example 4: Period finding using classical wave superposition

Period finding is the key part of Shor's prime factorization algorithm.[25] The algorithm includes two parts: classical and

quantum. The classical part accomplished on a general type computer is used for calculating f(k) ¼ m^kmod(N) function, where N is the number to be factorized and m is the almost randomly chosen number. The quantum part is the period finding subroutine aimed to find the period r of the function f(k). As the period is found, the classical computer checks the greatest common divisors(gcd): gcd($m^{r/2}$ þ 1, N), and gcd($m^{r/2}$ 1, N). At least one of gcd is a nontrivial factor of N. Period finding is the most challenging part for a classical digital computer. For instance, let us consider a sequence consisting of zeros and ones:

0101101110111100100110011100011101011011
0111100100110011100011101011011101111001 00:

In the most naïve and time-consuming approach, one would have to check the period at every repeating zero or one to find r ¼ 32. Shor developed a polynomial-time quantum algorithm exploiting quantum superposition and quantum Fourier transform to speed up the period-finding part, which provides a fundamental advantage over any type of digital-type computer.[25]

Period finding can also be efficiently accomplished using classical wave superposition. Let us consider a sequence of numbers f(k) as a superposition of waves with phases [f(k)/N]/π. As an example, we consider N ¼ 3 5 17 ¼ 255. We take m ¼ 13 and calculate a sequence of f(k) ¼ 13^kmod(255). The calculated numbers are converted into waves with phases f(k) ¼ [f(k)/N] π. In Fig. 7(a), it is shown the phase of the wave superposition as a function of k. The phase converges to some value (i.e., 1/3π in the given example) as k increases. This phase 1/3π is nothing but the phase of wave superposition in one period. To find the period, one has to find the first k with this phase. The inset in Fig. 7(a) shows an enlarged part of the plot. The phase of the superposition 1/3π appears at k ¼ 4, which is the period of the given modular sequence. The rest of the prime factorization is trivial. Calculating gcd($13^{4/2}$ þ 1, 255) ¼ 17 and gcd($13^{4/2}$ 1, 255) ¼ 7 gives one nontrivial factor of 255 (i.e., 17).

In Fig. 7(b), we plotted the results of numerical simulations for N ¼ 3 5 11 17 ¼ 2805 and m ¼ 13. As in the previous example, the sequence of the modular function is converted into wave superposition with f(k) ¼ [f(k)/N] π. The phase converges to 0:416 57π at large k values. To find the period of the sequence, one takes the first k with this phase. The inset to Fig. 7(b) shows the enlarged plot where the phase of the wave superposition is 0:416 57π at k ¼ 20, which is the period of the given modular sequence. The results of numerical modeling for the first 120 steps can be found in the supplementary materials. We want to stress that the described procedure is universal and can be applied to any periodic function. The phase of the superposition does converge to the period value regardless of particular numbers (phases) in the sequences. This approach also provides a fundamental advantage over the classical one by reducing the number of calculations. More details can be found in our preceding work,[26] where MHM was used to factorize number 15.

DISCUSSION

There are several observations we want to make based on the presented examples.

(i) Classical wave-based devices may provide a fundamental speedup in database searching compared to classical digital computers. For instance, example 1 shows the possibility of finding one of the 2^n 2 input combinations in 2n steps. Examples 2 and 3 demonstrate the feasibility of p_z^m speedup, where z is the dimensionality of the phase space (i.e., the number of independent wave inputs) and m is the number of phases per input. There may be a number of other examples showing the extreme capabilities of wave superposition for parallel database searching.

(ii) The speedup does not come with an exponential resource overhead. The number of devices and the energy of operation scale linearly with the number of inputs n.

(iii) The presented examples are not universal but can be applied to a special type of databases (i.e., with one absolute maximum). A detailed discussion on other types of databases, the possibility of extending the described search procedures, etc., is beyond the scope of this work.

(iv) All of the present examples are based on phase information coding, where logic states are related to the phases of the wave signals. The use of phase possesses certain advantages for parallel data processing (i.e., state superposition).

(iv) Besides database search, classical wave-based devices can be exploited in a special type of data processing (e.g., period

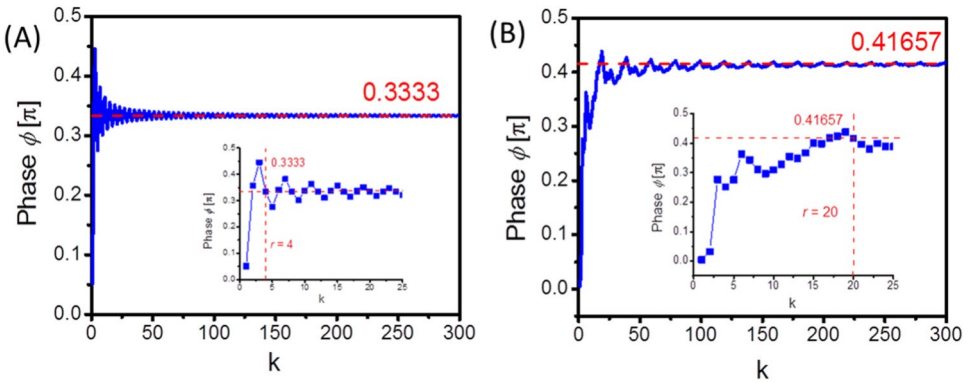


FIG. 7. Results of numerical modeling illustrating the period fining procedure from example 4. (a) The blue markers correspond to the phase of the superposition of waves with phases [f(k)/N]/π. The phase converges to some value at large k. The inset shows the enlarged part of the plot. The first k at which the phase reaches the converged value corresponds to period r. (a) N ¼ 255, m ¼ 13, r ¼ 4. (b) N ¼ 2805, m ¼ 13, r ¼ 20.

finding). These devices may not be as efficient as quantum computers (as they operate without quantum entanglement) but provide a fundamental speedup over digital computers.

This work raises a number of interesting questions regarding the theoretical and practical feasibility/efficiency of classical wave-based devices. One of the important questions to be further clarified is the trade-off between the number of waves superimposed, energy consumed, and the accuracy of measurements. The most appealing property of the phase-based approach is the ability to create state superposition without an exponential energy overhead. Input power scales linearly with the number of input ports (e.g., examples 1–4). At the same time, the output power difference between the "true" + superposition and "false" + superposition phase combinations decreases with the number of waves superposed. In the worst scenario, the maximum accuracy required for 1000 interfering waves is 10^7 (see Fig. 3). In the experimental part of example 3, the accuracy of PNA used for spin wave detection is 10^3 in the frequency range from 2 to 13.5 GHz.[27] It is practically possible to achieve 10^6 10^7 accuracy by implementing the most advanced superheterodyne receivers.[28] For instance, a sub-micro-degree phase measurement technique was demonstrated for lock-in amplifiers.[29] We want to focus that the overall functional throughput of wave-based devices is defined as

$$\text{Functional throughput} = \frac{\text{number of operations}}{\text{time} \quad \text{area}} = \frac{m^n}{[l/v_g]\ l^2}, \tag{7}$$

where m is the number of phases per input port, n is the number of input ports, l is the characteristic size of the device, and v_g is the group velocity of the information carrying wave. The number of states processed using superposition scales exponentially, while the time-area product scales proportional to n^3 assuming $l = n\ l_p$, where l_p is the characteristic dimension of one port. Even with limited accuracy of basic equipment (e.g., 10^4, $m = 2$, $n = 50$), wave-based devices will outperform any of the existing or proposed digital computers.[30]

The remarkable speedup in database searching compared to a digital classical computer has a simple explanation in terms of the database structure. The database described in all examples appears "unstructured" or "unsorted" for the digital computer. There is no other way to find the right input but checking one-by-one all possible phase combinations. The database appears "sorted" or "structured" for Oracle-C. The search method described in examples 2 and 3 is similar to binary searching.[31] This method is also known as half-interval searching or logarithmic searching in computer science.[32] This algorithm finds the position of a target value within a sorted array in logarithmic time O(logN), where N is the number of elements in the array. However, the array must be sorted first to be able to apply a binary search. In our cases, the input–output correlations of Oracle-C represent a "sorted" or "structured" database that allows us to implement superposition-based algorithms. Some of the presented examples show database search algorithms even faster than the ones proposed for quantum computers. Indeed, it takes 2n measurements for Oracle-C to find the right combination out of 2^n possible combinations in example 1 compared to $2^{n/2}$ queries required for Grover's algorithm. We want to clarify that all of the described search procedures may be implemented on quantum computers as well, leading to the same fundamental speedup compared to classical digital computers. To illustrate the computational power hierarchy, we have shown three computing machines: a classical digital computer, a classical wave-based computer, and a quantum computer in Fig. 8. All three devices have n input ports and one output port. The task is to find the only input

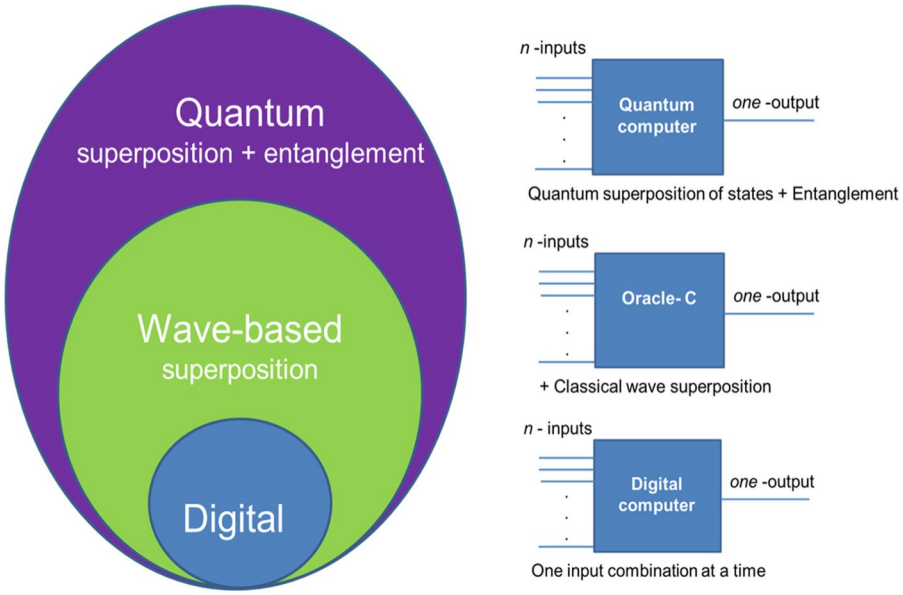


FIG. 8. Illustration of the computational power hierarchy comparing a classical digital computer, a classical wave-based computer, and a quantum computer. The three computers are used for unsorted database search (i.e., looking for one input combination leading to output one). The ability to use state superposition (classical or quantum) provides a fundamental advantage over a classical digital machine. The ability to use quantum entanglement makes quantum computers superior compared to classical wave-based computers. Classical wave-based devices may provide the same speedup as quantum computers as long as the quantum entanglement is not required.

combination leading to output 1. One-by-one checking of input combinations on the digital computer is allowed. One-by-one checking of input combinations on the classical wave-based computer and applying a superposition of states for any selected input ports are allowed. One-by-one checking of input combinations, utilizing superposition of states on any selected inputs, and utilizing quantum entanglement are allowed on the quantum computer. The ability to use state superposition (quantum or classical) provides a fundamental advantage over any classical digital machine. In turn, in order to exploit state superposition, one has to use phase in addition to amplitude for logic state representation.

To conclude this work, we would like to refer to a recent publication by Arute et al. showing Quantum supremacy using a programmable superconducting processor.[33] In this work, the team exploited both superposition and quantum entanglement of 53 qubits. It takes about 200 s for the quantum Sycamore processor to complete the task that would take approximately 10 000 years for a state-of-the-art classical supercomputer.[33] The advantages of quantum computers are undisputed. Here, we want to turn attention to classical wave-based devices, which may solve some problems with the same efficiency as quantum computers. Superposition of states, whether quantum or classical, provides a fundamental advantage over digital computers. The very first example presented in this work shows a database search over $2^{1}{}_{0}{}^{0}$ input state combinations. It will take a bit longer than the age of the universe (13.77 billion years) to check one-by-one all combinations (one combination/ns). It takes seconds to solve it using classical wave superposition. There is a lot of room for computing power enhancement by utilizing phase in addition to amplitude for information encoding. It may extend Moore's law until quantum computers are practically available.

METHODS

Device fabrication

The magnetic interferometer used in example 3 is a mesh of waveguides with four cross junctions made of a single crystal $Y_3Fe_2(FeO_4)_3$ film. The film was grown on top of a (111) gadolinium gallium garnett ($Gd_3Ga_5O_{12}$) substrate using the liquid-phase epitaxy technique. The micro-patterning was performed by laser ablation using a pulsed infrared laser ($\lambda \approx 1.03$ μm), with a pulse duration of ⍰256 ns. The YIG cross has the following dimension: the length of each waveguide is 3.65 mm; the width is 650 μm; and the YIG film thickness is 3.8 μm; and saturation magnetization of $4\pi M_0$ 1750G. There are four Π-shaped micro-antennas fabricated on the edges of the cross. Antennas were fabricated from a gold wire of thickness 24.5 μm and placed directly at the top of the YIG surface. Similar structure was described in Ref. 34.

Measurements

The antennas are connected to a programmable network analyzer (PNA) Keysight N5241A. Two of the antennas are used to generate two input spin waves. The inductive voltage is detected by the other two antennas. The set of attenuators (PE7087) and phase shifters (ARRA 9428A) is used to control the amplitudes and the phases of the interfering spin waves. The inductive voltage measurements' technique is well described in Ref. 35.

SUPPLEMENTARY MATERIAL

See the supplementary material for data that support examples 1 and 4.

ACKNOWLEDGMENTS

This work was supported in part by the INTEL CORPORATION (Award No. 008635, Spin Wave Computing) (Project director is Dr. D. E. Nikonov). This was supported in part by the Spins and Heat in Nanoscale Electronic Systems (SHINES), an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Basic Energy Sciences (BES) under Award No. SC0012670. The work of A. V. Kozhevnikov and Y. A. Filimonov was supported by the Russian Science Foundation under Grant No. 17-19-01673.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Authors' Contributions

A.K. conceived the experiment, provided numerical modeling, and wrote the manuscript; M.B. carried out the experiments; D.G. and H.C. developed the experimental setup; and A. Ko. and Y.F. provided the sample for experimental work. All authors discussed the data and the results and contributed to manuscript preparation.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1]National Academies of Sciences, Engineering, and Medicine, Quantum Computing: Progress and Prospects (The National Academies Press, 2018).
[2]P. W. Shor, Algorithms for Quantum Computation: Discrete Logarithms and Factoring, edited by E. Grumbling and M. Horowitz (IEEE, 1994).
[3]L. K. Grover, "From Schrodinger's equation to the quantum search algorithm," Am. J. Phys. 69, 769–777 (2001).
[4]I. Chuang, "Principles of quantum computation," in Quantum Entanglement and Information Processing, edited by D. Esteve, J. M. Raimond, and J. Dalibard (Elsevier Science, 2004), Vol. 79, p. 1.
[5]D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," Proc. R. Soc. London Ser. A 439, 553–558 (1992).
[6]P. Knight, "Quantum computing—Quantum information processing without entanglement," Science 287, 441–442 (2000).
[7]S. Lloyd, "Quantum search without entanglement—No. Art. 010301," Phys. Rev. A 61, 010301 (2000).
[8]A. Patel, "Optimal database search: Waves and catalysis," Int. J. Quantum Inform. 04, 815–825 (2006).
[9]A. J. P. Garner, "Interferometric computation beyond quantum theory," Foundations Phys. 48, 886–909 (2018).
[10]K. Wu, J. García de Abajo, C. Soci, P. Ping Shum, and N. I. Zheludev, "An optical fiber network oracle for NP-complete problems," Light: Sci. Appl. 3, e147 (2014).
[11]S.-H. Tan and P. P. Rohde, "The resurgence of the linear optics quantum interferometer—Recent advances and applications," Rev. Phys. 4, 100030 (2019).

[12]N. J. Cerf, C. Adami, and P. G. Kwiat, "Optical simulation of quantum logic," Phys. Rev. A 57, R1477–R1480 (1998).
[13]T. W. Hijmans, T. N. Huussen, and R. J. C. Spreeuw, "Time- and frequency-domain solutions in an optical analogue of Grover's search algorithm," J. Opt. Soc. Am. B 24, 214–220 (2007).
[14]P. G. Kwiat, J. R. Mitchell, P. D. D. Schwindt, and A. G. White, "Grover's search algorithm: An optical approach," J. Mod. Opt. 47, 257–266 (2000).
[15]K. Urban, Wavelets in Numerical Simulation Problem Adapted Construction and Applications (Springer, 2002).
[16]A. Das and B. K. Chakrabarti, "Quantum annealing and related optimization methods," Lect. Note Phys. 679, 152–157 (2005).
[17]E. E. Schadt, M. D. Linderman, J. Sorenson, L. Lee, and G. P. Nolan, "Computational solutions to large-scale data management and analysis," Nat. Rev. Gen. 11, 647–657 (2010).
[18]M. Tsiknakis, V. J. Promponas, N. Graf, M. D. Wang, S. T. C. Wong, N. Bourbakis, and C. S. Pattichis, "Guest editorial computational solutions to large-scale data management and analysis in translational and personalized medicine," IEEE J. Biomed. Health Inform. 18, 720–721 (2014).
[19]B. Fan, H. K. Thapar, and P. H. Siegel, "Multihead multitrack detection for next generation magnetic recording, part I: Weighted Sum subtract joint detection with ITI estimation," IEEE Trans. Commun. 65, 1635–1648 (2017).
[20]A. Khitun, "Magnonic holographic devices for special type data processing," J. Appl. Phys. 113, 164503 (2013).
[21]F. Gertz, A. V. Kozhevnikov, Y. A. Filimonov, D. E. Nikonov, and A. Khitun, "Magnonic holographic memory: From proposal to device," IEEE J. Exploratory Solid-State Comput. Devices Circuits 1, 67–75 (2015).
[22]C. Kittel, Introduction to Solid State Physics, 8th ed. (John Wiley and Sones, Inc., 2005).
[23]F. Gertz, A. Kozhevnikov, Y. Filimonov, and A. Khitun, "Magnonic holographic memory," IEEE Trans. Magn. 51, 1–5 (2015).
[24]A. Khitun, "An entertaining physics: On the possibility of energy storage enhancement in electrostatic capacitors using the compensational inductive electric field," Appl. Phys. Lett. 117, 153903 (2020).
[25]P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithm problems," SIAM J. Comput. 26, 1484–1509 (1997).
[26]Y. Khivintsev, M. Ranjbar, D. Gutierrez, H. Chiang, A. Kozhevnikov, Y. Filimonov, and A. Khitun, "Prime factorization using magnonic holographic devices," J. Appl. Phys. 120, 123901 (2016).
[27]See https://www.keysight.com/us/en/assets/9018-03959/technical-specifications/9018-03959.pdf for the technical characteristics of PNA.
[28]D. Song, H. Liu, L. Qi, and B. Zhou, "A general purpose adaptive fault detection and diagnosis scheme for information systems with superheterodyne receivers," Complexity 2018, 1–9.
[29]W. D. Walker, "Sub-microdegree phase measurement technique using lock-in amplifiers," in IEEE International Frequency Control Symposium (IEEE, 2008), pp. 825–828.
[30]More Moore, International Roadmap for Devices and Systems, 2017 ed. (IEEE, 2018), pp. 1–25, see https://irds.ieee.org/images/files/pdf/2017/2017IRDS_MM.pdf
[31]D. Knuth, Sorting and Searching. The Art of Computer Programming, 2nd ed. (Addison-Wesley Professional, 1998).
[32]L. F. Williams, "A modification to the half-interval search (binary search) method," in Proceedings of the 14th ACM Southeast Conference (ACM-SE 14, 1976), pp. 95–101.
[33]F. Arute, K. Arya, R. Babbush et al., "Quantum supremacy using a programmable superconducting processor," Nature 574, 505–510 (2019).
[34]A. Kozhevnikov, F. Gertz, G. Dudko, Y. Filimonov, and A. Khitun, "Pattern recognition with magnonic holographic memory device," Appl. Phys. Lett. 106, 142409 (2015).
[35]T. J. Silva, C. S. Lee, T. M. Crawford, and C. T. Rogers, "Inductive measurement of ultrafast magnetization dynamics in thin-film permalloy," J. Appl. Phys. 85, 7849–7862 (1999).