پیاده سازی الگوریتم یافتن دوره تناوب با استفاده از شبیه سازی تبدیل فوریه کوانتومی

زهره مقاره عابد[۱] - محسن عشوریان[۲] - کیومرث غوثی[۲]

(۱) کارشناس ارشد - دانشکده برق، دانشگاه آزاد اسلامی، واحد نجف آباد

(۲) استادیار - گروه برق، دانشگاه آزاد اسلامی، واحد شهر مجلسی

تاریخ دریافت: پاییز ۱۳۸۸ تاریخ پذیرش: زمستان ۱۳۸۸

خلاصه: در این مقاله، به معرفی تبدیل فوریه کوانتومی به عنوان جزء کلیدی بسیاری از الگوریتمهای پرکاربرد می‌پردازیم. الگوریتمهایی که به حل مسائلی منتهی می‌شوند که حل آنها روی یک کامپیوتر کلاسیک، سخت و گاهی غیرعملی است. تبدیل فوریه کوانتومی به عنوان کلیدی برای تخمین فاز کوانتومی مطرح می‌گردد. هدف ما در این مقاله پیاده سازی الگوریتم یافتن دوره تناوب است. یافتن دوره تناوب از جمله مسائلی است که حل آن روی یک کامپیوتر کوانتومی، به طور نمایی، سریع‌تر از حل آن روی یک کامپیوتر کلاسیک است. حال آنکه اساس الگوریتم یافتن دوره تناوب، تخمین فاز کوانتومی است. پس با شبیه سازی تبدیل فوریه کوانتومی، قادر به پیاده‌سازی الگوریتم یافتن دوره تناوب خواهیم بود. در این مقاله، شبیه سازی تبدیل فوریه کوانتومی با استفاده از نرم افزار Matlab انجام می‌شود.

کلمات کلیدی: کامپیوتر کوانتومی، محاسبه کوانتومی، تبدیل فوریه کوانتومی، تخمین فاز کوانتومی.

۱- مقدمه

اگرچه امروزه کامپیوترها به صورت کم حجم در آمده‌اند و وظیفه‌ی خود را با سرعت بیشتری انجام می‌دهند، اما کاری که انجام می‌دهند تغییر نکرده است: محاسبه روی رشته‌ای از بیت‌ها. پیشرفت و رشد سخت افزار هنوز هم با سرعت، ادامه دارد به طوری که یک قانون تجربی مشهور به قانون مور[۱] در سال ۱۹۶۵ توسط گوردن مور، بدین مضمون ارائه شد: هر هجده ماه یکبار، تعداد ترانزیستورهایی که در یک حجم معین به کار برده می‌شوند دو برابر می‌گردد، که این امر باعث افزایش سرعت پردازش و میزان حافظه‌ی کامپیوترها می‌شود [۱،۲]. قانون مور به طور تقریبی تاکنون برقرار بوده است و به عقیده‌ی بسیاری اگر این روند ادامه یابد تا سال ۲۰۲۰ می‌توان قطعات کوچکتری ساخت و به جایی رسید که هر گیت تنها تعداد کمی اتم داشته باشد. در واقع یکی از دلایل اصلی برای تلاش در کنترل سیستمهای کوانتومی مجزا، در ابعاد بسیار کوچک، همین مطلب است [۳،۴]. اما در چنین ابعادی قوانین فیزیک کلاسیک، قادر به توضیح رفتار ذرات نیستند و قوانین مکانیک کوانتومی حکم فرماست. بنابراین اگر ابعاد مدارهای الکترونیکی، همچنان کوچک‌تر شوند، فناوری کوانتومی باید جایگزین تکنولوژی امروز شود یا در کنار آن قرار گیرد و آن را تکمیل کند [۵،۶].

مطالعات اخیر نشان می‌دهد که می‌توان از مکانیک کوانتومی برای حل مسائل ریاضی اساساً مشکل، مثل تجزیه‌ی اعداد صحیح، در مراحل کمتری نسبت به روش‌های کلاسیک، استفاده کرد [۷،۸]. تحقیقات تجربی و تئوری که تا کنون در حوزه محاسبه کوانتومی انجام شده است در یک مرحله ابتدایی قرار دارد و تلاش‌های زیادی برای توضیح جنبه‌های عملی و تئوری پردازش اطلاعات، مبتنی بر پدیده‌های کوانتومی مورد نیاز است. محاسبه کوانتومی، هم اکنون بخش مهمی از تحقیقات در علم کامپیوتر را تشکیل می‌دهد. علاقه به این حوزه، به دلیل وجود الگوریتم‌های کارآمدی است که می‌توانند برخی از محاسبات را به طور قابل ملاحظه‌ای سریعتر از الگوریتم‌های کلاسیک انجام دهند. ساخت یک کامپیوتر کوانتومی، پاسخگوی بسیاری از اعمالی است که به صرف منابع بسیار زیاد نیاز دارند. شبیه سازی یک کامپیوتر کوانتومی روی یک کامپیوتر کلاسیک، مسئله‌ی سختی است.

بنابر نظر فاینمن، کامپیوترهای کلاسیک، هرگز قادر به شبیه‌سازی کامل رفتار یک سیستم کوانتومی، در زمانی از مرتبه چندجمله‌ای نیستند. در واقع، به دلیل رفتار نمایی سیستم‌های کوانتومی، شبیه سازی آن‌ها روی کامپیوترهای متداول، به طور نمایی به عملگرهای زیادی نیاز دارد [۹].

تاکنون از محیط‌هایی همچون Fortran 90، C، C++ برای پیاده سازی محاسبات کوانتومی استفاده شده است. به علاوه، نرم افزارهایی ویژه‌ی محاسبات کوانتومی طراحی شده‌اند که از جمله آنها می‌توان به QCMPI، QC_lib، QCL و Senko اشاره کرد [۹]. این نرم‌افزارها هنوز به طور کامل پاسخگوی نیاز ما در حوزه‌ی کوانتومی نیستند و قادر به شبیه سازی کلیه‌ی قابلیت‌های حوزه‌ی کوانتومی نمی‌باشند. به عنوان مثال، به کمک این نرم‌افزارها می‌توان محاسبات را روی تعداد محدودی از کوبیت[۲]ها انجام داد. همچنین از ویژگی‌هایی همچون توازی کوانتومی و برهم نهی کوانتومی به طور محدود می‌توان بهره برد. از دیگر محدودیت‌های آنها این است که اکثر این نرم افزارها قادر به اجرای محاسبات کوانتومی روی حالتهای پایه محاسباتی هستند.

۲- بیت کوانتومی

بیت، مفهوم اساسی محاسبات کلاسیک و اطلاعات کلاسیک است. محاسبات و اطلاعات کوانتومی نیز متکی بر یک مفهوم قابل مقایسه یعنی بیت کوانتومی یا به اختصار کوبیت ساخته می‌شوند.

یک بیت می‌تواند در دو حالت منطقی "صفر" یا "یک" باشد. حال آن که بیت کوانتومی، می‌تواند همزمان هم در حالت "صفر" و هم در حالت "یک" باشد [۹،۱۰]. دو حالت ممکن برای یک بیت کوانتومی عبارتند از $|0\rangle, |1\rangle$. نماد $|\ \rangle$ نماد دیراک[۳]، نامیده می‌شود که نماد استاندارد برای حالت‌ها در ماشین‌های کوانتومی است. تفاوت بین بیتها و بیتهای کوانتومی در این است که بیت‌های کوانتومی می‌توانند در یک برهم نهی[۴] از حالات نیز باشند همان طور که در رابطه‌ی (۱) نشان داده شده است:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle \qquad (۱)$$

اعداد α و β اعداد مختلط هستند. حالت یک بیت کوانتومی، یک بردار در فضای بردار مختلط دوبعدی است. حالت‌های خاص $|0\rangle, |1\rangle$ به عنوان حالتهای پایه محاسباتی، شناخته می‌شود و یک پایه‌ی متعامد و نرمال برای این فضا تشکیل می‌دهند. زمانی که یک بیت کوانتومی را اندازه گیری می‌کنیم نتیجه با احتمال $|\alpha|^2$، "صفر" و با احتمال $|\beta|^2$، "یک" خواهد بود. جمع احتمالات باید یک باشند یعنی $|\alpha|^2 + |\beta|^2 = 1$، که این در واقع، شرطی برای نرمالیزه شدن حالت بیت کوانتومی است.

۳- گیت‌های کوانتومی

یکانی بودن، تنها شرط روی گیت‌های کوانتومی است. هر ماتریس یکانی، یک گیت کوانتومی معتبر را مشخص می‌کند. گیت‌های کوانتومی، همواره معکوس پذیر هستند چون معکوس یک ماتریس یکانی، یک ماتریس یکانی است.

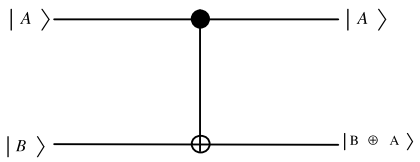
در ادامه به معرفی برخی از گیت‌های کوانتومی می‌پردازیم.

یکی از عملگرهای کوانتومی مهم که یک تبدیل تک بیتی است، عملگر هادامارد[۵] است که به صورت رابطه (۲) تعریف می‌شود:

$$|0\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0> + |1>)$$
$$|1\rangle \longrightarrow \frac{1}{\sqrt{2}}(|0> - |1>) \qquad (۲)$$
$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

وقتی به طور همزمان، عملگرهای H، روی n بیت، به کار برده می‌شوند یک برهم نهی از همه‌ی 2^n حالت ممکن، ایجاد می‌کنند. تبدیلی که H را برای n بیت به کار می‌برد تبدیل والش[۶] یا والش- هادامارد[۷] نامیده می‌شود. این عملگر، چیزی نیست جز n عملگر هادامارد که به طور موازی بر روی بیت‌های کوانتومی ورودی، اثر می‌کنند و به صورت $H^{\otimes n}$ نشان داده می‌شود [۱۰،۱۱].

یک گیت کوانتومی دیگر، گیت Controlled-Not یا CNOT است. این گیت، دو بیت کوانتومی ورودی داردکه به عنوان بیت کوانتومی کنترل و بیت کوانتومی هدف، شناخته می‌شوند. نمایش مداری CNOT در شکل (۱) نشان داده شده است. در این شکل، خط بالایی، بیت کوانتومی کنترل و خط پایین، بیت کوانتومی هدف، را نشان می‌دهد. عملکرد این گیت می‌تواند به صورت $|A, B\rangle \longrightarrow |A, B \oplus A\rangle$ خلاصه شود.

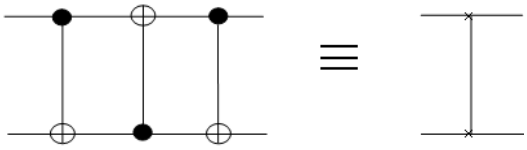


شکل (۱): نمایش مداری گیتControlled-NOT

Fig. (1): The circuit symbol of Controlled-NOT

یک گیت کوانتومی دیگر نیز وجود دارد که از ۳ گیت CNOT تشکیل شده است و عمل جابجایی را روی دو بیت کوانتومی انجام می‌دهد. توالی گیت‌ها، اثرات نشان داده شده در رابطه (۳) را روی یک حالت پایه محاسباتی $|a,b\rangle$ دارد. نمایش مداری این گیت در شکل (۲) نشان داده شده است [۱۱،۱۲].

$$\left|a,b>\rightarrow\right|a,\ a\oplus b>$$
$$\rightarrow\left|a\oplus(a\oplus b),a\oplus b>=\right|b\ ,\ a\oplus b> \qquad (۳)$$
$$\rightarrow\left|b,(a\oplus b)\oplus b>=\right|b,a>$$



شکل (۲): مدار تعویض دو بیت کوانتومی
Fig. (2): The circuit exchange for two-bit quantom

۴- تبدیل فوریه کوانتومی[8]

مهم‌ترین کشف در محاسبه کوانتومی تا کنون، این است که کامپیوترهای کوانتومی می‌توانند به طور کارآمد برخی عملیات را که توسط یک کامپیوتر کلاسیک، عملی نیستند، اجرا کنند. به عنوان مثال، پیدا کردن فاکتورهای اول یک عدد صحیح n بیتی، با استفاده از بهترین الگوریتم‌های کلاسیک شناخته شده به $\exp(\Theta(n^{\frac{1}{3}}\log^{\frac{2}{3}} n))$ عملیات، نیاز دارد. بنابراین، تجزیه به عاملهای اول، روی یک کامپیوتر کلاسیک، به عنوان یک مسئله‌ی سخت، مورد توجه قرار می‌گیرد. برعکس، یک الگوریتم کوانتومی می‌تواند همان کار را با استفاده از $O(n^2 \log n \log\log n)$ عملیات انجام دهد. یعنی یک کامپیوتر کوانتومی می‌تواند یک عدد را به صورت نمایی، سریع‌تر از بهترین الگوریتم‌های کلاسیک شناخته شده، به عامل‌های اول، تجزیه کند [۱۳،۱۴].

به کمک تبدیل فوریه کوانتومی، تخمین فاز کوانتومی[9] که تقریبی از مقادیر ویژه‌ی یک عملگر یکانی تحت شرایط معین است، امکان پذیر می‌شود. تخمین فاز کوانتومی به ما اجازه می‌دهد که مسائلی همچون مسئله‌ی یافتن درجه[10]، مسئله‌ی تجزیه به عامل‌های اول[11] و مسئله یافتن دوره تناوب[12] را حل نماییم. تخمین فاز کوانتومی همچنین می‌تواند با الگوریتم جستجوی کوانتومی ترکیب شود و مسئله‌ی تخمین راه حل‌ها برای یک مسئله‌ی جستجو را حل کند[۱۵]. تبدیل فوریه کوانتومی، دقیقاً همان تبدیل فوریه گسسته است، اگرچه نمادگذاری قراردادی برای تبدیل فوریه کوانتومی، کمی متفاوت است. تبدیل فوریه‌ی کوانتومی، روی یک پایه‌ی متعامد و نرمال تعریف می‌شود تا یک عملگر خطی روی حالت‌های پایه باشد همان طور که در رابطه‌ی (۴) نشان داده شده است.

از این پس در سایر روابط $i=\sqrt{-1}$ می‌باشد.

$$|j\rangle \longrightarrow \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1} e^{2\pi ijk/N}|k\rangle \qquad (۴)$$

این تبدیل، یک تبدیل یکانی است و بنابراین می‌تواند برای یک کامپیوتر کوانتومی پیاده‌سازی شود [۱۵،۱۶].

$N=2^n$ را در نظر می‌گیریم که n عددی صحیح است و پایه‌های $|0\rangle,\ldots,|2^n-1\rangle$، پایه‌های محاسباتی، برای یک کامپیوتر کوانتومی با n بیت کوانتومی هستند. در این صورت حالت $|j\rangle$ را می‌توان با استفاده از نمایش باینری $j=j_1j_2\ldots j_n$ یا به صورت رابطه (۵)، نشان داد.

$$j=j_1 2^{n-1}+j_2 2^{n-2}+\ldots.+j_n 2^0 \qquad (۵)$$

همچنین حالت $0.j_L j_{L+1}\ldots j_m$ را به شکل رابطه (۶)، می‌توان نشان داد.

$$j_L/2+j_{L+1}/4+\ldots+j_m/2^{m-L+1} \qquad (۶)$$

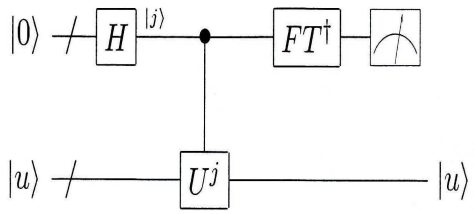
به کمک جبر، تبدیل فوریه‌ی کوانتومی می‌تواند به صورت نمایش محصول مفید نشان داده شده در رابطه (۷) ارائه شود.

$$|j_1,\ldots,j_n\rangle \longrightarrow \frac{\left(|0\rangle+e^{2\pi i 0.j_n}|1\rangle\right)\left(|0\rangle+e^{2\pi i 0.j_{n-1}j_n}|1\rangle\right)\ldots\left(|0\rangle+e^{2\pi i 0.j_1 j_2\ldots j_n}|1\rangle\right)}{\left(2^{n/2}\right)} \qquad (۷)$$

تبدیل فوریه کوانتومی با استفاده از یک الگوریتم از مرتبه $\Theta(n^2)$ قابل اجرا است این در حالی است که بهترین الگوریتم‌های کلاسیک برای محاسبه‌ی تبدیل فوریه‌ی گسسته روی 2^n المان عبارتند از الگوریتمهایی مثل تبدیل فوریه سریع که تبدیل فوریه‌ی گسسته را با استفاده از $\Theta(n2^n)$ گیت، محاسبه می‌کنند [۱۷،۱۶].

۵- تخمین فاز کوانتومی

اگر U یک عملگر یکانی باشد که یک بردار ویژه $|u\rangle$ با مقدار ویژه‌ی $e^{2\pi i\varphi}$ دارد، هدف الگوریتم تخمین فاز، تخمین φ است [۱۶]. برای اجرای تخمین، فرض می‌کنیم که جعبه‌های سیاهی در دسترس داریم که قادر به آماده کردن حالت $|u\rangle$ و اجرا کردن عملیات Controlled - U^{2^j} برای اعداد صحیح غیرمنفی مناسب j هستند. روش تخمین فاز کوانتومی از دو ثبات، استفاده می‌کند. ثبات اول، در ابتدا t بیت کوانتومی در حالت $|0\rangle$ در بردارد. t وابسته به دو چیز انتخاب می‌شود: تعداد ارقام دقت در تخمین برای φ و احتمال موفقیت روش تخمین فاز. ثبات دوم، در حالت $|u\rangle$ قرار دارد و تعداد بیت کوانتومی که برای ذخیره‌ی $|u\rangle$ لازم است، در بر دارد. مدار نشان داده شده در شکل (۳) مربوط به تخمین فاز کوانتومی است.



شکل (۳): مدار پیاده سازی کننده‌ی تخمین فاز کوانتومی
Fig. (3): The circuit implantation for quantom phase estimation

۶- الگوریتم تخمین فاز کوانتومی

- ورودیها: ۱) یک جعبه سیاه که عملیات Controlled $-U^j$ را برای عدد صحیح j، اجرا می‌کند
۲) یک حالت ویژه $|u\rangle$ از U با مقدار ویژه‌ی $e^{2\pi i\varphi_u}$
۳) t بیت کوانتومی که با $|0\rangle$ شروع می‌شود
- خروجی‌ها: یک تقریب n بیتی $\tilde{\varphi}_u$ از φ_u
- بار محاسباتی: $O(t^2)$ عملگر و یک جعبه سیاه احضار Controlled $-U^j$

- مراحل الگوریتم:

حالت اولیه $1.\rightarrow |0\rangle|u\rangle$

اعمال گیت هادامارد $2.\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle|u\rangle$

اعمال جعبه سیاه $3.\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1}|j\rangle U^j|u\rangle$

نتیجه جعبه سیاه $4.\rightarrow \frac{1}{\sqrt{2^t}}\sum_{j=0}^{2^t-1} e^{2\pi i j\varphi_u}|j\rangle|u\rangle$

اعمال تبدیل فوریه معکوس $5.\rightarrow |\varphi_u\rangle|u\rangle$

اندازه گیری ثبات اول $6.\rightarrow \tilde{\varphi}_u$

۷- الگوریتم شور

پیتر شور توانست از توازی موجود در کامپیوترهای کوانتومی به منظور عامل یابی اعداد با استفاده از یک قانون در نظریه اعداد استفاده کند. این الگوریتم، عملاً بسیار ساده بود. یک عدد برای عامل یابی دریافت می‌شود (N). مقداری کمتر از N برای a انتخاب می‌شود. با فرض اینکه N حاصلضرب دو عدد اول است، a باید نسبت به N نیز اول باشد. به ازای مقادیر مختلف x، تابع $f(x) = (a^x \bmod N)$ محاسبه می‌شود. تمام این عملیات با توان محاسبات کوانتومی می‌تواند در واحد زمان انجام شود. یک الگوی تکرار، در نتایج تابع وجود دارد که دوره این تکرار را باید پیدا کرد. دوره با r نشان داده می‌شود. سپس مقادیر $\gcd(a^{r/2}+1,N)$ و $\gcd(a^{r/2}-1,N)$ محاسبه می‌شوند. حداقل یکی از این مقادیر باید عاملی از N باشد که این فرضیه به دلیل برقراری تساوی $a^r = (1 \bmod N)$ امکان پذیر بوده و طی رابطه‌ی (۸) نشان داده شده است.

$$a^r - 1 = (a^{r/2}-1).(a^{r/2}+1) = 0 \bmod N \qquad (۸)$$

به این صورت $(a^{r/2}-1).(a^{r/2}+1)$ مضرب صحیحی از N است.

۸- مراحل الگوریتم شور برای تجزیه اعداد صحیح

در الگوریتم شور از خاصیت بر هم نهی کوانتومی استفاده می‌شود، که اجازه می‌دهد n کیلوبیت در یک لحظه، تمام 2^n حالت ممکن را داشته باشند.

الگوریتم شور، جهت پیدا کردن عوامل اول عدد صحیح N می‌باشد. شور نشان داد که کامپیوترهای کوانتومی، قادر به محاسبه عوامل اول اعداد خیلی بزرگ، در یک زمان کوتاه هستند. این الگوریتم، وابسته به توازی کوانتومی و تبدیل فوریه کوانتومی می‌باشد. مدارات کوانتومی، برای این الگوریتم طراحی شده‌اند [۱۸،۱۷]. با انتخاب عدد صحیح N، مراحل پیاده سازی الگوریتم شور، جهت پیدا کردن عوامل اول آن به شرح زیر می‌باشند:

مرحله اول: پیدا کردن عدد صحیح Q که توانی از 2 بوده و $N^2 < Q < 2N^2$ باشد. این مرحله توسط کامپیوترهای کلاسیک انجام می‌شود.

مرحله دوم: انتخاب عدد صحیح a که کوچکتر از N بوده و نسبت به آن اول باشد $(\gcd(a,N)=1)$. روشهای کلاسیکی زیادی برای پیدا کردن a وجود دارند. این مرحله نیز توسط کامپیوترهای کلاسیک انجام می‌شود.

مرحله سوم: ایجاد دو رجیستر کوانتومی به نام رجیستر ورودی و رجیستر خروجی. رجیستر ورودی باید دارای تعداد کافی کوبیت، جهت نگهداری اعدادی به بزرگی Q-1 باشد و رجیستر خروجی باید دارای تعداد کافی کوبیت جهت نگهداری اعدادی به بزرگی N-1 باشد.

مرحله چهارم: بارگذاری رجیستر ورودی با مقادیر صحیح 0 تا Q-1 و بارگذاری رجیستر خروجی با مقدار 0. مجموع حالت‌های اولیه رجیسترهای کوانتومی سیستم در این نقطه در رابطه‌ی (۹) بیان شده است.

$$\underbrace{\frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x\rangle}_{\substack{\text{Input}\\ \text{register}}}\ \underbrace{|0\rangle}_{\substack{\text{Output}\\ \text{register}}} \qquad (۹)$$

مرحله پنجم: اجرای تابع $a^x \bmod N$ برای هر عدد ذخیره شده در رجیستر ورودی و ذخیره سازی نتایج آن در رجیستر خروجی. به دلیل توازی کوانتومی، این مرحله در یک گام اجرا می‌شود. کامپیوتر کوانتومی فقط تابع $a^{|x\rangle} \bmod N$ را محاسبه می‌کند که $|x\rangle$، بر هم نهی حالتهای به دست آمده در مرحله چهارم می‌باشد.

این مرحله، روی کامپیوتر کوانتومی اجرا می‌شود. وضعیت رجیسترهای کوانتومی در این مرحله در رابطه (۱۰) نشان داده شده است.

$$\frac{1}{\sqrt{Q}}\sum_{x=0}^{Q-1}|x\rangle\left|a^x \bmod N\right\rangle \qquad (۱۰)$$

مرحله ششم: به دست آوردن رجیستر خروجی با مقادیر K، مطابق رابطه (۱۱):

$$a^x \bmod N = K \qquad (۱۱)$$

این عمل، به وسیله کامپیوتر کوانتومی اجرا می‌شود. وضعیت رجیسترهای کوانتومی بعد از این مرحله به صورت رابطه (۱۲) می‌باشد:

$$\frac{1}{\sqrt{\|A\|}}\sum_{x'\in A}|x'\rangle|k\rangle \qquad (۱۲)$$

که A مجموعه‌ای از x' است به طوری که $a^x \bmod N = K$ و $\|A\|$ تعداد عناصر این مجموعه است.

مرحله هفتم: اعمال تبدیل فوریه کوانتومی، روی رجیستر ورودی. تبدیل فوریه کوانتومی باعث تغییر حالت $|x\rangle$ می‌شود همان طور که در رابطه (۱۳) نشان داده شده است:

$$|x\rangle = \frac{1}{\sqrt{Q}}\sum_{c=0}^{Q-1}|C\rangle\, e^{2\pi i x c/Q} \qquad (۱۳)$$

این مرحله توسط کامپیوتر کوانتومی در یک گام، به وسیله توازی کوانتومی اجرا می‌شود. بعد از تبدیل فوریه کوانتومی، وضعیت رجیسترها به صورت رابطه (۱۴) است:

$$\frac{1}{\sqrt{\|A\|}}\sum_{x'\in A}\frac{1}{\sqrt{Q}}\sum_{c=0}^{Q-1}|C\rangle|k\rangle e^{2\pi i x'c/Q} \qquad (۱۴)$$

مرحله هشتم: اندازه گیری رجیستر ورودی. این مقدار m نامیده می‌شود. این مرحله توسط کامپیوتر کوانتومی اجرا می‌شود.

مرحله نهم: با به کار بردن m روی کامپیوترهای کلاسیک، مقدار دوره تکرار r توسط روش‌های مختلف به دست می‌آید.

مرحله دهم: با داشتن مقدار r، عوامل اول عدد N توسط بزرگترین مقسوم علیه مشترک، مطابق رابطه‌ی (۱۵) به دست می‌آیند. این مرحله توسط کامپیوترهای کلاسیک به دست می‌آید.

$$\left.\begin{aligned}\gcd(a^{r/2}+1,N)=p\\ \gcd(a^{r/2}-1,N)=q\end{aligned}\right\}\Rightarrow N=p.q \qquad (۱۵)$$

نتیجه‌ای که شور بدان دست یافت این بود که یک کامپیوتر کوانتومی، در یک زمان از مرتبه‌ی چند جمله‌ای عمل تجزیه را انجام می‌دهد.

۹- یافتن دوره تناوب

فرض کنید f یک تابع متناوب است که یک بیت، به عنوان خروجی، تولید می‌کند به طوری که f(x+r)=f(x). به ازای $0<r<2^L$ که L عبارت است از تعداد بیت مورد نیاز برای نمایش r و. $x,r\in\{0,1,2,..\}$ U یک جعبه سیاه کوانتومی است که تبدیل یکانی $U|x\rangle|y\rangle\rightarrow|x\rangle|y\oplus f(x)\rangle$ را اجرا می‌کند. U روی یک دامنه محدود که ابعاد آن توسط دقت مطلوب برای r تعیین می‌شود، عمل می‌کند [۱۵،۱۶].

۱۰- الگوریتم کوانتومی پیدا کردن دوره تناوب

- ورودیها: ۱) یک جعبه سیاه که عملیات $U|x\rangle|y\rangle=|x\rangle|y\oplus f(x)\rangle$ را اجرا می‌کند.

۲) یک حالت برای ذخیره کردن ارزیابی تابع که با $|0\rangle$ شروع می‌شود.

۳) t بیت کوانتومی که با $|0\rangle$ شروع می‌شود.

- خروجیها: کوچکترین عدد صحیح r>0 به طوری که $f(x+r)=f(x)$

- بار محاسباتی: یک استفاده از U و $O(L^2)$ عملیات دیگر.

- مراحل الگوریتم:

حالت اولیه $1.|0\rangle|0\rangle$

ایجاد برهم نهی $2.\rightarrow\frac{1}{\sqrt{2^t}}\sum_{x=0}^{2^t-1}|x\rangle|0\rangle$

اعمال U $3.\rightarrow\frac{1}{\sqrt{2^t}}\sum_{x=0}^{2^t-1}|x\rangle|f(x)\rangle$

$$\approx\frac{1}{\sqrt{r2^t}}\sum_{l=0}^{r-1}\sum_{x=0}^{2^t-1}\exp\left[\frac{2\pi i l x}{r}\right]|x\rangle|\hat{f}(l)\rangle$$

اعمال تبدیل فوریه معکوس به رجیستر اول

$$4.\rightarrow\frac{1}{\sqrt{r}}\sum_{l=0}^{r-1}|l/r\rangle|\hat{f}(l)\rangle$$

اندازه گیری رجیستر اول $5.\rightarrow l/r$

اعمال الگوریتم کسرهای متوالی $6.\rightarrow r$

در الگوریتم فوق $\varphi=l/r$ است یعنی همان فاز. کلید فهمیدن این الگوریتم که مبتنی بر تخمین فاز است، مرحله ۳ می‌باشد که در آن، حالت نشان داده شده در رابطه (۱۶) که تبدیل فوریه $|f(x)\rangle$ است، وارد شده است.

$$|\hat{f}(l)\rangle\equiv\frac{1}{\sqrt{r}}\sum_{x=0}^{r-1}\exp\left[\frac{-2\pi i l x}{r}\right]|f(x)\rangle \qquad (۱۶)$$

آنچه در مرحله ۳ استفاده شده است مبتنی بر رابطه (۱۷) است.

$$|f(x)\rangle=\frac{1}{\sqrt{r}}\sum_{L=0}^{r-1}\exp\left[\frac{2\pi i l x}{r}\right] \qquad (۱۷)$$

الگوریتم کسرهای متوالی که در مرحله آخر از آن استفاده شده است عبارت است از توصیف اعداد حقیقی بر حسب اعداد صحیح، به صورتی که در رابطه (۱۸) نشان داده شده است.

$$[a_0,\ldots,a_M]=a_0+\cfrac{1}{a_1+\cfrac{1}{a_2+\cfrac{1}{\ldots+\cfrac{1}{a_M}}}} \qquad (۱۸)$$

که $a_0,\ldots a_M$ اعداد صحیح مثبت هستند. واضح است که این الگوریتم بعد از یک تعداد محدود از مراحل شکستن و معکوس کردن، برای هر عدد گویا، به پایان می‌رسد.

الگوریتم کوانتومی پیدا کردن دوره تناوب، مبتنی بر تخمین فاز کوانتومی می‌باشد چون نخست $\varphi=\frac{1}{r}$ یعنی فاز به دست می‌آید و در مرحله آخر، با اعمال الگوریتم کسرهای متوالی، r یعنی دوره تناوب به دست می‌آید. پس برای یافتن دوره تناوب باید تخمین فاز کوانتومی، پیاده سازی شود، حال آن‌که کلید تخمین فاز کوانتومی، تبدیل فوریه کوانتومی است. از طرف دیگر، مرحله (۴) در الگوریتم یافتن دوره تناوب، اعمال تبدیل فوریه معکوس می‌باشد، پس برای حل مسئله یافتن دوره تناوب، اولین گام شبیه‌سازی تبدیل فوریه کوانتومی است [۱۵].

۱۱- مقایسه‌ای بین الگوریتم کلاسیک یافتن دوره تناوب و الگوریتم کوانتومی نظیر

الگوریتم کلاسیک یافتن دوره تناوب از نظر اصول کار، مشابه الگوریتم کوانتومی نظیر است. اما آنچه باعث برتری الگوریتم کوانتومی بر الگوریتم کلاسیک می‌شود مزایا و قابلیت‌های حوزه کوانتوم است که سبب افزایش سرعت اجرای الگوریتم و کاهش بار محاسباتی می‌گردد.

به عنوان مثال، در حوزه کوانتوم، خاصیتی تحت عنوان توازی کوانتومی وجود دارد که باعث افزایش سرعت محاسبات، به طور نمایی نسبت به حالت کلاسیک می‌گردد. در مرحله سوم الگوریتم کوانتومی یافتن دوره تناوب از این خاصیت استفاده شده است. این خاصیت، این امکان را فراهم می‌کند که مقدار تابع f به ازای 2^t عدد (t، تعداد بیت‌های کوانتومی است) به طور همزمان، محاسبه شود. در حالیکه در حالت کلاسیک باید 2^t بار عملیات محاسبه صورت گیرد تا f به ازای 2^t عدد مورد نظر محاسبه شود. بنابراین، طی مرحله سوم از الگوریتم مذکور، به طور نمایی یک صرفه‌جویی در زمان و منابع محاسباتی صورت گرفته است. در مرحله چهارم این الگوریتم از تبدیل فوریه کوانتومی استفاده شده است. تبدیل فوریه کوانتومی نسبت به تبدیل فوریه کلاسیک، به طور نمایی به عملگرهای کمتری نیاز دارد. به بیان دقیقتر، تبدیل فوریه کوانتومی روی 2^n عنصر، با استفاده از یک الگوریتم از مرتبه $\Theta(n^2)$ قابل اجراست. این در حالی است که بهترین الگوریتم‌های کلاسیک برای محاسبه تبدیل فوریه گسسته روی 2^n عنصر، عبارتند از الگوریتم‌هایی مثل تبدیل فوریه سریع که تبدیل فوریه گسسته را با استفاده $\Theta(n2^n)$ گیت، محاسبه می‌کند. بنابراین در این مرحله نیز یک تسریع نمایی صورت گرفته و به طور نمایی تعداد گیت کمتری مورد نیاز است. بنابراین، به طور کلی این الگوریتم نسبت به معادل کلاسیک خود بسیار سریعتر عمل می‌کند همچنین بار محاسباتی و تعداد گیت‌های مورد نیاز کاهش می‌یابد. از آنجا که ظرف چند سال آینده، کامپیوترهای کوانتومی جایگزین کامپیوترهای کلاسیک خواهند شد مسائلی چون یافتن دوره تناوب یک تابع متناوب روی این کامپیوترها باید با الگوریتم‌های کوانتومی حل شوند. از این رو بایستی برای حل مسائل مختلف روی این کامپیوترها از روش‌های کوانتومی استفاده کرد [۱۹،۱۸].

۱۲- شبیه سازی تبدیل فوریه کوانتومی

یکی از مزایای شبیه‌سازی ارائه شده در این مقاله، این است که به کمک آن می‌توان تبدیل فوریه کوانتومی را روی هر تعداد بیت کوانتومی اجرا کرد. از طرف دیگر، کوبیت‌ها لزوما در حالت‌های پایه محاسباتی نیستند و می‌توانند در یک برهم نهی دلخواه از حالت‌ها باشند. بنابراین کوبیت‌ها در هر حالتی که باشند، می‌توان تبدیل فوریه‌ی کوانتومی آن‌ها را یافت. به کمک این شبیه‌سازی از یک تسریع نمایی در محاسبه‌ی تبدیل فوریه بهره‌مند خواهیم شد. به علاوه، در عمل تعداد عملگرهای مورد نیاز، به طور نمایی کاهش می‌یابد که این امر مرهون خاصیت توازی کوانتومی است. نتایج اعمال تبدیل فوریه کوانتومی، روی 1، 2، 3 و 4 کوبیت که در برهم نهی دلخواهی از حالات قرار داشته‌اند، در جدول (۱) بیان شده است.

ابتدا سعی شد کوبیت‌ها در حالت‌های مختلف، تعریف شوند و سپس در برهم نهی‌های دلخواه قرار گیرند. سپس برنامه‌ای ترتیب داده شد به گونه‌ای که ماتریسی از ضرایبی که باید در مقدار ورودی، ضرب شوند (مطابق رابطه (۴)) تولید شود. به این ترتیب برای محاسبه تبدیل فوریه کوانتومی کافیست ماتریس ضرایب محاسبه گردد. سپس با ضرب کردن ماتریس حاصل در کوبیت‌ها (که در برهم نهی‌های دلخواه از حالات پایه محاسباتی قرار دارند) تبدیل فوریه آن‌ها محاسبه می‌گردد. در پیوست (۱)، مراحل پیاده سازی یکی از توابعی که تعریف نموده‌ایم و در شبیه‌سازی مورد استفاده قرار داده‌ایم، ارائه شده است. ویژگی این تابع این است که حالت خاصی از شبیه‌سازی ما است. در واقع با استفاده از این برنامه، می‌توان مستقیما تبدیل فوریه کوانتومی را به ازای یک ورودی دو حالته (یک کوبیت) محاسبه کرد.

۱۳- نتیجه گیری

استفاده از ابزارهای مکانیک کوانتومی در کار محاسبات، علاوه بر آن که موجب کوچکتر شدن سخت افزار می‌گردد، این امکان را نیز فراهم می‌آورد که برخی از مسائل محاسباتی را در زمان بسیار کمتری در مقایسه با کامپیوترهای کلاسیک انجام دهیم.

اهمیت پیاده‌سازی تبدیل فوریه کوانتومی از این جهت است که این تبدیل، پایه و اساس الگوریتم تخمین فاز کوانتومی است و با استفاده از تخمین فاز کوانتومی می‌توان مسائلی چون یافتن دوره تناوب، یافتن درجه و تجزیه به عامل‌های اول را حل نمود. در این مقاله الگوریتم یافتن دوره تناوب، مبتنی بر الگوریتم تخمین فاز کوانتومی، پیاده سازی شد. یافتن دوره تناوب از جمله مسائلی است که حل آن روی یک کامپیوتر کلاسیک سخت بوده و نسبت به حل آن روی یک کامپیوتر کوانتومی، به طور نمایی به عملگرهای بیشتری نیاز دارد. بنابراین با این پیاده سازی می‌توان با تعداد عملگر کمتر و صرف زمان کمتر، مسئله فوق الذکر را به صورت بهینه حل نمود.

پی‌نوشت:

1- Moore's law
2- Qubit
3- Dirac notation
4- Superposition
5- Hadamard
6- Walsh
7- Walsh-Hadamard
8- Quantum Fourier Transform
9- Quantum phase estimation
10- Order-finding problem
11- Factoring problem
12- Period-finding problem

Table (1): The quantomised Fourier transform of states super position for 1,2,3 and 4 qbits

جدول (۱): تبدیل فوریه کوانتومی برهم نهی حالات، به ازای 1، 2، 3و 4 کوبیت

تعدادکوبیت	حالت ورودی	تبدیل فوریه ورودی
1	0.57735|0> + 0.70711|1>	0.90825|0> + -0.091752+6.1232e-017i|1>
2	0.40825|00> + 0.44721|01> + 0.70711|10> + 0.57735|11>	1.07|00> + -0.14943-0.065068i|01> + 0.045396+4.6846e-017i|10> + -0.14943+0.065068i|11>
3	0.44721|010> + 0.33333|101> + 0.35355|111>	0.40097|000> + 0.005055-0.013608i|001> + -0.15811-0.0071489i|010> + -0.005055-0.32984i|011> + -0.084737+5.0983e-016i|100> + -0.005055+0.32984i|101> + -0.15811+0.0071489i|110> + 0.005055+0.013608i|111>
4	0.70711|1010> + 0.57735|1100> + 0.44721|1111>	0.43292|0000> + -0.021707-0.31212i|0001> + -0.065281+0.09772i|0010> + 0.16779-0.083955i|0011> + -0.032439-0.1118i|0100> + 0.082215-0.12263i|0101> + -0.22339-0.25583i|0110> + -0.22829+0.22655i|0111> + 0.20931+8.0947e-016i|1000> + -0.22829-0.22655i|1001> + -0.22339+0.25583i|1010> + 0.082215+0.12263i|1011> + -0.032439+0.1118i|1100> + 0.16779+0.083955i|1101> + -0.065281-0.09772i|1110> + -0.021707+0.31212i|1111>

مراجع

[1] M.A. Nielsen, I.L. Chuang, "Quantum computation and quantum information", Cambridge Univ. Press, 2000.

[2] D.C. Marinescu, M. Marinescu Gabriela, "From classical to quantum information", Jan. 19, 2010.

[3] P. Kaye, R. Laflamme, M. Mosca, "An introduction to quantum computing", 1st ed, Oxford University Press, 2007.

[4]G. Cincotti, "Prospects on planar quantum computing", Jou. of Lig. Tech., Vol.27, No.24, Dec. 15, 2009.

[5] V.V. Shende, I.L. Markov, S.S. Bullock, "Finding small two-qubit circuits", SPIE, Vol.5436, pp.348-359, Apr. 2004.

[6] L. Marchildon, "Does quantum mechanics need interpretation?", IEEE / ICQNMT, pp.11-16, Cancun, Feb. 2009.

[7] I.G. Karafyllidis, "Quantum computer simulator based on the circuit model of quantum computation", IEEE Trans. Circ. and Sys., Vol.52, No.8, pp.1590 - 1596, Aug. 2005.

[8] M.A. Aoki, "Qubits structure and coherence in a one-way quantum computer", IEEE/ICEEE, pp.150-152, MexicoCity, Sep. 2007.

[9] S. Caraiman, A. Archip, V. Manta, "A grid enabled quantum computer simulator", IEEE/ISSNASC , pp.189-196, Timisoara, Sep. 2009.

[10] H. De Raedt, K. Michielsen, "Handbook of theoretical and computational nanotechnology", Edited by M. Rieth and W. Schommers, American Scientific Publishers, Los Angeles, 2005.

[11] D. Marinescu, G. Marinescu, "Approaching quantum computing", 1st ed, Pearson/Prentice Hall, 2005.

[12] G.F. Viamontes, I.L. Markov, J.P. Hayes, "Graph-based simulation of quantum computation in the state-vector and density-matrix representation", Qua. Info. and Com., Vol.5, No.2, pp.113-130, 2005.

[13] S.C. Ding, Z. Jin, Q. Yang, "Evolving quantum circuits at the gate level with a hybrid quantum-inspired evolutionary algorithm", Soft Computing, Vol.12, pp.1059-1072, Authorized, 2008.

[14] R. Jozsa, "Quantum algorithms and the Fourier transform", Los Alamos preprint archive, http://xxx.lanl.gov–/archive/quant-ph/9707033, 1997.

[15] B. Djordjevic Ivan, "On the photonic implementation of universal quantum gates, Bell states preparation circuit, quantum relay and quantum LDPC encoders and decoders", Vol.2, No.1, Feb. 2010.

[16] W. Jiajia, C. Hanwu, Li. Zhiqiang, "The study of simulation technique of quantum compute and quantum Fourier transform", IEEE/ICCSCWD, pp.1111-1115, Melbourne, April 2007.

[17] D. Coppersmith, "An approximate Fourier transform useful in quantum computing", IBM Tech. Rep. RC 19642 (1994), quant-ph/0201067.

[18] FU. Xiangqun, W. Bao, C. Zhou, "Design and implementation of quantum factorization algorithm", IEEE/ISIITSI, pp.748-752, 2010.

[19] R.C. Vidya, H.D. Phaneendra, M.S. Shivakumar, "Quantum algorithms and hard problems", IEEE/ICCI, pp.783-787, Beijing, July 2006.

پیوست:

```
function P = prewalsh(n)
%P = prewalsh(n)
%
%Create n-qubit prewalsh matrix.

if n==1
   P = [ 1 1; 1 -1]/sqrt(2);
else
   P1 = prewalsh(1);
   P=1;
   for j=1:n
      P=kron(P,P1);
   end
end
```