**העבודה כוללת הערות המנחה במצב סקירה**

**מחשוב קוונטי**

**מגיש:** XXX

תוכן עניינים

מחשוב קוונטי – למה צריך את זה?3

Qubit5

Quantum logic gates7

Quantum algorithm8

מחשבים קוונטיים כיום9

צפי לעתיד10

מחשוב קוונטי – למה צריך את זה?

בעשרות השנים האחרונות גדל כוח המחשוב באופן אקספוננציאלי בעוד גודלם של הטרנזיסטורים, מהם מורכבים המחשבים, הולך וקטן במהירות. תופעה זו תוארה בשנת 1965 ע"י גורדון מור מייסדה של אינטל וידועה **כחוק מור** כפי שניתן לראות באיור הבא:



באיור ניתן לראות את מספר הטרנזיסטורים המרכיבים מעבד כפונקציה של השנה בה יצא המעבד. הגרף מוצג בגרף חצי-לוגריתמי, מה שמעיד על גדילה אקספוננציאלית של מספר הטרזיסטורים: מספר הטרנזיסטורים המרכיבים מעבד מוכפל בערך כל שנתיים וגודלם של הטרנזיסטורים קטן במהירות.

בשנים האחרונות מסתמן כי חוק מור יגיע לסופו כאשר גודלם של הטרנזיסטורים יקטן לננומטרים בודדים, דהיינו קרוב לסדר גודל של אטום. עבור טרנזיסטורים בגודל כזה, תופעות קוונטיות מתחילות להשפיע כאשר אלקטרונים מבצעים מנהור קוונטי ולמעשה הטרנזיסטורים שלנו כבר לא עובדים כמו שצריך (מקור).

כאן נכנסים המחשבים הקוונטים לתמונה. מחשבים קוונטים, כפי שנראה בהמשך, מנצלים את אותם התופעות הקוונטיות שעוצרות את התקדמותם של מחשבים רגילים ומשתמשים בהן כדי לקבל ביצועים מהירים בהרבה מאשר מחשבים רגילים.

אבל לפני שניכנס לעובי הקורה, יש השואלים למה בכלל צריך את זה? האם המחשבים שלנו אינם מהירים מספיק? כמובן שתמיד רוצים שהמחשבים יהיו מהירים יותר אבל למעשה, קיימות בעיות שפשוט לא יכולות להיפתר בצורה יעילה באמצעות מחשבים רגילים, אחת מהן הינה בעיית הסוכן הנוסע. בעיה זו מוגדרת כך: "בהינתן רשימת ערים והמרחק בין כל שתי ערים, מהו המסלול הקצר ביותר, אשר יעבור בכל עיר פעם אחת, ויחזור לעיר ממנה התחיל?" נראה פשוט לא? ובכן, אם נתבונן באיור הבא נראה שזה לא פשוט כפי שזה נראה, באיור מופיע מספר המסלולים האפשריים כפונקציה של מספר הערים:



אם ניקח מחשב רגיל הפועל בתחום הג'יגהרצים ומבצע $10^{9}$ פעולות בשניה ונתבונן ב-14 ערים, נקבל $10^{11}$ *מסלולים אפשריים והמחשב ימצא את המסלול הקצר תוך 100 שניות. פשוט. אבל, אם נעלה ל-22 ערים נקבל* $10^{19}$ *מסלולים אפשריים ולמחשב ייקח 1600 שנים! ואם נתבונן ב-28 ערים צריך יותר זמן מאורך חיי היקום!!! (צטט סימוכין לכך ולהסביר, ולו באופן איכותי, את החשבון הזה)*

*הצפי הוא שמחשבים קוונטיים באמצעות ביצועיהם המהירים יתנו מענה לבעיות כאלה בזמן סביר. אבל לפני שנתבונן בפתרון בעיות באמצעות מחשבים קוונטיים נתבונן ביחידה הבסיסית ביותר במחשב הקוונטי, ה-*Qubit.

**Qubit:**

אם כן, במה שונים מחשבים קוונטיים ממחשבים רגילים? בניגוד למחשבים רגילים המשתמשים בביטים כדי לשמור מידע, מחשבים קוונטיים משתמשים בביטים קוונטיים הנקראים Qubits, כדי לשמור מידע?.

Qubits משתמשים בעיקרון הסופרפוזיציה, וכך, בניגוד לביטים רגילים היכולים להימצא באחד משני מצבים 0 או 1, Qubits יכולים להימצא בסופרפוזיציה של שני המצבים. ניתן לבטא את המצב של ה-Qubit כ:

$$\left|ψ> =α\right|0> +β|1>$$

תחת האילוץ:

$$\left|α\right|^{2}+ \left|β\right|^{2}=1$$

וכך, בזמן ש-n ביטים רגילים יכולים ליצור $2^{n}$ קונפיגורציות אפשריות אבל נאלצים להימצא באחת מהן, n Qubits יכולים להימצא בסופרפוזיציה בכל המצבים הללו בו זמנית. כאשר בכל פעם שמוסיפים Qubit נוסף מכפילים את כוח המחשוב. למעשה, לפי המודל שתארנו, מחשב קוונטי בעל 30 Qubits בלבד יהיה חזק יותר מהמחשב הכי חזק שקיים כיום (מקור, על סמך מה?), ומחשב קוונטי בעל 300 Qubits יהיה חזק יותר מכל המחשבים בעולם מחוברים יחד! זהו כוחו של המחשוב הקוונטי (אתה רוצה שיאמינו לך? עדיף שתבסס קביעות כאלה).

מבחינה פיזית ביטים רגילים מיוצגים ע"י זרם, יש/אין זרם, דהיינו 0 או 1. אבל פיזית

Qubits ניתנים לייצוג באמצעות כל מערכת קוונטית המורכבת משתי רמות. קיימות שתי מערכות מוכרות, הראשונה, פוטון והקיטוב שלו, סופרפוזיציה של אופקי ומאונך, והשנייה, אלקטרון והספין שלו בכיוון ציר z, up או down.

ייצוג מקובל של מצבי הספין של אלקטרון ידוע כספרת בלוך (Bloch Sphere) כמתואר באיור הבא:



והמצב של ה-Qubit יתואר כ:

$$|ψ> =cos⁡(\frac{θ}{2})|α> + e^{iϕ}sin⁡(\frac{θ}{2})|β>$$

*כאשר:*

$$0\leq θ\leq π ו 0\leq ϕ\leq 2π$$

שטח הפנים של הספרה מייצג את מרחב המצבים, וביט רגיל ייוצג ע"י שני הקטבים המתאימים ל- $θ=0, π $*.*

*ייצוג מקביל עבור הקיטובים של הפוטון ידוע כספרת פונקרה(Poincare's Sphere).*

אחרי כל מה שתיארנו צריך להיזהר, n Qubits מחזיקים $2^{n}$ מצבים בו זמנית אבל בסופו של דבר כאשר נבצע מדידה נקרוס לאחד המצבים הללו נקבל n ביטים של מידע בדיוק כמו מחשב קלאסי רגיל. הבעיה הזאת היא בעיה מאוד מהותית, ברגע שנמדוד את הערך של Qubit הוא כבר לא יהיה בסופרפוזיציה וכל מה שהרווחנו מכך שהשתמשנו ב-Qubits ייעלם! נקבל מחשב רגיל! אם כן, כיצד נוכל לדעת את הערך של Qubit מסוים בלי למדוד אותו? אחת הפתרונות המוצעים כדי לפתור בעיה זו היא **שזירה קוונטית** (Entanglement). ניתן לייצר שני חלקיקים כך שסכום הספינים הכולל שלהם יהיה אפס (כיצד?), וכך, אם מודדים אחד מהם ורואים שיש לו ספין חצי אזי לשני יהיה ספין מינוס חצי, במקרה כזה נגיד ששני החלקיקים **שזורים**. באמצעות תכונה קוונטית זו תינתן האפשרות לדעת מה הערך של ה-Qubit גם בלי למדוד אותו. אך יש חוקרים הטוענים שאין צורך בשיטה זו כדי לענות על הבעיה ושזו טעות לחשוב ששזירה קוונטית היא המפתח למחשוב הקוונטי[[1]](#footnote-1).

**Quantum logic gates:**

מחשבים רגילים פועלים באמצעות שערים לוגיים המקבלים ביטים בודדים, מבצעים עליהם פעולות לוגיות ומוצאים ביטים חדשים. המקבילה שלהם במחשבים קוונטיים הם שערים לוגיים קוונטיים. שערים לוגיים קוונטיים מקבלים n Qubits במצב מסויים של סופרפוזיציה ומוצאים אותם שוב במצב של סופרפוזיציה אבל עם סיכויים שונים להימצא בכל מצב. שערים לוגיים קוונטיים מיוצגים באמצעות מטריצות אוניטריות בגודל $2^{n}X2^{n}$, ורצף של n Qubits יהיה מיוצג באמצעות וקטור בגודל $2^{n}$. השערים הלוגיים הקוונטיים הנפוצים מקבלים Qubit בודד או שני Qubits. בניגוד לשערים לוגיים קלאסיים שערים לוגיים קוונטיים הינם **הפיכים**, כלומר, שער לוגי שמופעל פעמיים יחזיר אותנו למצב המקורי. נביא דוגמא אחת להמחשה:

**Hadamard gate:**

שער זה הינו שער לוגי הפועל על Qubit יחיד ומוגדר ע"י המטריצה:

$$H=\frac{|0>+|1>}{\sqrt{2}}<0|+\frac{|0>-|1>}{\sqrt{2}}<1|=\frac{1}{\sqrt{2}}\left(\begin{matrix}1&1\\1&-1\end{matrix}\right)$$

Hadamard gate לוקח מצבים טהורים והופך אותם לסופרפוזיציה של שני המצבים בסיכויים שווים, כלומר:

$H(|0>)= \frac{1}{\sqrt{2}}|0>+\frac{1}{\sqrt{2}}|1>$, $H(|1>)= \frac{1}{\sqrt{2}}|0>-\frac{1}{\sqrt{2}}|1>$

ואם נפעיל את השער שוב נקבל:

$H(\frac{1}{\sqrt{2}}|0>+\frac{1}{\sqrt{2}}|1>)=|0>$, $H(\frac{1}{\sqrt{2}}|0>-\frac{1}{\sqrt{2}}|1>)=|1>$

כך שהשער אכן הפיך.

כמו כן, H הינה מטריצה אוניטרית ומתקיים: $HH^{\*}=I$

הרבה אלגוריתמים קוונטיים משתמשים בשער זה בתור צעד התחלתי בגלל שהוא ממפה n Qubits המאותחלים במצב $|0>$ לסופרפוזיציה של כל $2^{n}$ המצבים בסיכויים שווים.

כאשר שער זה פועל על n Qubits הוא פשוט מופעל על כל Qubit בנפרד. מזה נובע כי ה- quantum Hadamard transform(הפעולה המבוצעת ע"י Hadamard gate) הפועל על n Qubits הינו בעל סיבוכיות O(n) כאשר ה- Hadamard transform הקלאסי[[2]](#footnote-2) (שאינו מובא כאן עקב מורכבות החורגת ממטרתה של עבודה זו) הינו בעל סיבוכיות של O(nlogn). כבר ניתן לראות כי אפילו שערים קוונטיים פשוטים יעילים יותר מהשערים הקלאסיים.

שערים לוגיים קוונטים נוספים הינם:

* Pauli-X gate
* Pauli-Y gate
* Pauli-Z gate
* Phase shift gate
* Swap gate

אך קיימים עוד רבים.

**Quantum algorithm:**

אלגוריתמים קוונטיים משתמשים במעגלים קוונטיים המורכבים משערים לוגיים קוונטים (כפי שתיארנו לעיל) בכדי לפתור בעיות שמחשבים רגילים מתקשים להתמודד עימם. למעשה, אלגוריתמים קוונטיים פותרים בעיות הרבה יותר מהר מאשר מחשבים רגילים בזכות היכולת שלהם לעבור על הרבה אפשרויות במקביל ולא בצורה ליניארית כמו מחשבים רגילים. חשוב להדגיש שכיוון שהאלגוריתמים עובדים עם הסתברויות בגלל המצב של הסופרפוזיציה של ה-Qubits,לא תמיד האלגוריתמים נותנים את התוצאה הנכונה ולפעמים צריך להריץ אותם מספר פעמים עד לקבלה של התשובה המבוקשת, ואף על פי כן יתרונם של המחשבים הקוונטיים הוא עצום.

אלגוריתמים ידועים הינם אלגוריתם שור[[3]](#footnote-3) המשמש לפירוק לגורמים של מספרים גדולים ומשמש לקידוד והצפנה, ואלגוריתם גרובר[[4]](#footnote-4) המשמש לחיפוש במבנה נתונים שאינו ממוין. אלגוריתם שור רץ אקספוננציאלית יותר מהר מהאלגוריתם הקלאסי המקביל אליו ואלגוריתם גרובר רץ באופן ריבועי יותר מהר מהאלגוריתם הקלאסי המקביל.

אלגוריתמים אלו מדגימים את יעילותם של המחשבים הקוונטיים יחסית למחשבים הרגילים.

אלגוריתמים קוונטיים נוספים הינם:

* Simon's algorithm[[5]](#footnote-5)
* Quantum phase estimation algorithm
* Boson sampling problem[[6]](#footnote-6)
* Deutsch–Jozsa algorithm[[7]](#footnote-7)

**מחשבים קוונטיים כיום:**

נכון לשנת 2017 מחשבים קוונטים אכן קיימים. חברה קנדית בשם ""D-wave[[8]](#footnote-8) החלה בשנת 1999 לייצר מחשב קוונטי. ואכן, בשנת 2007 הם הציגו את המעבד הקוונטי הראשון שהורכב מ-16 Qubits. בשנת 2011 החברה הוציאה לראשונה לשיווק מחשב קוונטי הנקרא "D-wave one" שהורכב מ-128 Qubits. החברה שיפרה את הדגמים שלה כך שבשנת 2017 הם שיווקו את הדגם החדש ביותר, ה-"D-wave 2000Q" המורכב מ-**2000 Qubits!**

מספר נתונים על ה-"D-wave 2000Q":

* מקורר לטמפרטורה של $0.0015K^{°}$!!!
* המעבד בנוי ממוליך על ולכן לא מייצר חום.
* חשוף לשדה מגנטי אשר עוצמתו קטנה פי 50000 מזה של כדוה"א.
* פועל בוואקום גבוה, $10^{-9}$ לחץ אטמוספרי.
* המערכת צורכת פחות מ-25KW! (מחשב העל 'Titan' צורך 8.2MW)
* בעל API המאפשר לתכנת עליו ב-MATLAB, C++, python.

כך נראה ה-"D-wave 2000Q":



קבוצת חוקרים מהרווארד השתמשו ב-"D-wave one" כדי לפתור את "protein folding problem"[[9]](#footnote-9) אך זה לא עבד בצורה מושלמת. לפי החוקרים, תוך שימוש ב-81 Qubits ו-10000 מדידות רק 13 מדידות היו נכונות, דבר שרק מדגיש יותר את הבעייתיות הקיימת באלגוריתמים הקוונטיים מבחינת תוצאות מדויקות אך מצד שני מראה כי מחשבים קוונטים אכן עובדים גם אם לא בצורה מדויקת.

**צפי לעתיד:**

כפי שראינו, למחשבים קוונטיים יש פוטנציאל עצום וכל מה שנישאר לעשות הוא להמתין לראות כיצד תתפתח תעשיה זו. השאלה העיקרית היא האם מחשבים קוונטיים הם המהפכה הבאה בעולם התעשייה? המצאת המחשב הובילה למהפכה, מחשב הוא לא סתם חשבונייה מיוחדת המסוגלת לחשב יותר מהר מחשבונייה רגילה, המחשבים הפכו את הצורה שבה העולם פועל ושינו את צורת הפעולה וההסתכלות שלנו בכל תחומי החיים. מעניין יהיה לראות האם המחשבים הקוונטיים הינם כמו מחשבים רגילים רק מהירים יותר? או, כמו המצאת המחשב הנוכחי, ישנה המחשב הקוונטי גם את הצורה בה אנו מסתכלים על העולם.

לעניות דעתי, אם אכן התעשייה מבוססת -קוונטים תגרום לכך שניתן יהיה לעבוד עם מחשב קוונטי באותה הצורה והנוחות בה אנו עובדים עם מחשבים רגילים כיום, השפעתה תהא עצומה במידה שתשנה את צורת החיים שלנו לבלי היכר. דוגמא/ות?

1. Jozsa, Richard, and Noah Linden. "On the role of entanglement in quantum-computational speed-up." *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 459. No. 2036. The Royal Society, 2003.‏ [↑](#footnote-ref-1)
2. Ahmed, Nasir, and Kamisetty Ramamohan Rao. "Walsh-hadamard transform." *Orthogonal Transforms for Digital Signal Processing*. Springer Berlin Heidelberg, 1975. 99-152.‏ [↑](#footnote-ref-2)
3. Shor, Peter W. "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." *SIAM review* 41.2 (1999): 303-332.‏ [↑](#footnote-ref-3)
4. Grover, Lov K. "A fast quantum mechanical algorithm for database search." *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM, 1996.‏ [↑](#footnote-ref-4)
5. Simon, Daniel R. "On the power of quantum computation." *SIAM journal on computing* 26.5 (1997): 1474-1483.‏ [↑](#footnote-ref-5)
6. Aaronson, Scott, and Alex Arkhipov. "The computational complexity of linear optics." *Proceedings of the forty-third annual ACM symposium on Theory of computing*. ACM, 2011.‏ [↑](#footnote-ref-6)
7. Deutsch, David, and Richard Jozsa. "Rapid solution of problems by quantum computation." *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 439. No. 1907. The Royal Society, 1992.‏ [↑](#footnote-ref-7)
8. https://www.dwavesys.com/ [↑](#footnote-ref-8)
9. Perdomo-Ortiz, Alejandro, et al. "Finding low-energy conformations of lattice protein models by quantum annealing." *arXiv preprint arXiv:1204.5485* (2012).‏ [↑](#footnote-ref-9)