Matematikai és Informatikai Intézet

Tanulás segítő program és applikáció egy kvantumbites logikai kapukhoz

Készítette
Bakos Rózsa Ajándék
Programtervező informatikus BSc

Témavezető
Biró Csaba
Egyetemi docens

Eger, 2024

Tartalomjegyzék

Bevezetés

A kvantuminformatika térhódítása egyre nagyobb figyelemnek örvend, valamint új lehetőségei hatalmas potenciállal bírnak a számítástechnika terén. A kvantummechanika alapelveinek felhasználása új típusú megoldásokat eredményez, amelyek képesek áthidalni a jelenleg is használt számítógépek korlátait. Ennek köszönhetően ezek a kvanumszámítógépek olyan problémák megoldásában ígérkeznek hatékonyabbnak, amelyek a hagyományos számítógépek számára nehezen, vagy egyáltalán nem megoldhatók.

A kvantumszámítógépek potenciális alkalmazási területei közé tartozik a mesterséges intelligencia, kriptográfia, gyógyszerkutatás és a számításelmélet. Azonban az ilyen rendszerek működése rendkívül érzékeny a környezeti tényezőkre, például gyakran használnak extrém alacsony hőmérsékletet követelő szupravezetőket. A jelenleg létező kvantumszámítógépek egyelőre kezdeti fázisban járnak, de a különböző cégek, kutatócsoportok között kialakult verseny ezen a helyzeten bármikor változtathat.

A kvantum-számítástechnika egyik kulcsfontosságú területe a kvantum logikai kapuk koncepciója, amelyek a kvantum bitek (más néven qubit) manipulációját teszik lehetővé. Ehhez a már említett kvantummechanika alapelveire támaszkodnak. Értelmezésük, valamint a hozzá tartozó összetett matematikai háttér sok esetben nehézséget okozhat.

Szakdolgozatom célja, hogy bemutasson ezen problémák áthidalására egy olyan tanulás segítő programot, mely közelebb hozza az érdeklődőkhöz a kvantumkapuk koncepcióját. Ezt egyqubites kapuk bemutatásával teszem meg. Az alkalmazás tervezése és implementálása során figyelembe veszem a felhasználók igényeit és a pedagógiai célokat, miközben kihasználom a kvantumtechnológia által nyújtott lehetőségeket.

A továbbiakban bemutatom a kvantuminformatika alapjait, a kvantumlogikai kapuk működését és jellemzőit, valamint részletesen ismertetem a fejlesztett tanulás segítő programot, beleértve annak tervezési alapelveit, implementációját és tesztelését. Végül összefoglalom az elért eredményeket és felvázolom a jövőbeli kutatási irányokat ezen a területen.

1. fejezet

Kvantuminformatika

1.1. Előzmények

A kvantummechanika gyökerei az 1800-as évek elejére vezethetők vissza, ahol a fizika többek között olyan problémákkal is küzdött, mint a hőmérsékleti sugárzás.

Az első matematikai modellt, amely erre magyarázatot adott, Max Planck német fizikushoz kötjük. A ma már Planck-féle kvantumhipotézisnek ismert levezetés 1900-ban történt előterjesztésre. Ezt a dátumot tekintjük a "régi" kvantumelméleti korszak kezdetének, valamint a kvantumfizika születésének. A jelenséggel Albert Einstein is foglalkozott, aki 1905-ben megjelent, fényelektromos jelenségről szóló cikkében a már említett hőmérsékleti sugárzástól független is alkalmazta a Planck-féle kvantumhipotézist.

Ezután rohamos növekedésnek kezdett a tudományág. A "régi" kvantumelméleti korszak végét 1924-re datáljuk, amikor De Brogile publikálta anyaghullámokról szóló elméletét. A modern kvantummechanika születése 1925-ben történt, jelenleg is ebben a korszakban vagyunk.

Természetesen az informatikát sem kerülte el ez a hullám. 1981-ben Richard Feynman előadásában felvetette egy kvantumelveken működő számítógép ötletét, ami képes kvantumrendszereket szimulálni. Az ő nevéhez fűződik a kvantumszámítógép kifejezés. A korszak hasonlóan fontos neve még Paul Benioff, aki 1982-ben publikált, tanulmányában lefektette a kvantumszámítási modellek alapjait. Feynman és Benioff gondolatait felhasználva 1985-ben David Deutsch bemutatta tanulmányában egy kvantumszámítógép elméleti működését, ami képes elvégezni a klasszikus számítógép számításait, de kihasználva a kvantummechanika előnyeit. Fontos személy volt még ebben az időszakban Peter Shor, aki 1994-ben megalkotott egy olyan kvantumalgoritmust, ami a napjainkban használt titkosítási eljárásokat veszélybe sodorhatja. Az 1994-2000-es időszakban számos hasonlóan fontos kvantumalgoritmusok született. Ilyen például az 1996-ban készült Grover algoritmus, ami a rendezetlen adatokban való keresést segítette.

Ahogy ezek a felfedezések egyre tovább bővítették a kvantumszámítás fogalmát, úgy felmerült az igény a kvantumlogikai kapukra is. Az 1980-as évek végétől kezdve számos kaput állítottak elő.

1.2. Jelenlegi helyzet

A 21. században kialakult egy verseny a kutatóintézetek és tech óriások közt, cél egy kvantumszámítógép megépítése volt. Ehhez a korszakhoz fűződik a kvantumfölény kifejezés is. Ez arra utal, hogy egy kvantumszámítógép képes megoldani egy olyan problémát, melyre klasszikus társai nem képesek lehetséges időn belül.

2011-ben a D-Wave Systems cég volt az első, akik azt állították, hogy megépítették az első kereskedelmi forgalomban kapható kvantumszámítógépet, a D-Wave One-t, ami 128 qubites lapkakészleten működött.

Fontos esemény volt még a 2019-ben a Google által bejelentett kvantumfölény. 53 qubites Sycamore processzorával rendelkező kvantumszámítógépük 200 másodperc alatt megoldott egy konkrét problémát, amelyhez egy klasszikus szuperszámítógépnek körülbelül 10 000 év alatt végzett volna el. Ugyanebben az évben jelent meg az IBM első kvantumszámítógépe is, az IMB Quantum System One.

Ebben az időszakban elért jelentős előrelépés ellenére számos kihívás még mindig fennáll, ilyen például a kvantumbitek minősége és stabilitása. Az olyan cégek, mint az IBM, Google és a Microsoft továbbra is jelentős összegeket fektet a kvantumszámítógépek fejlesztésébe. Ezen kívül számos ország és nagyhatalom is foglalkozik ezzel a tudományággal.

Hazai viszonylatban is folyamatos kutatások történnek, ezeket különböző egyetemek mellett a Kvantuminformatikai Nemzeti Laboratórium és a HunQuTech konzorcium vezeti. Céljuk elérni a nemzetközi szintet.

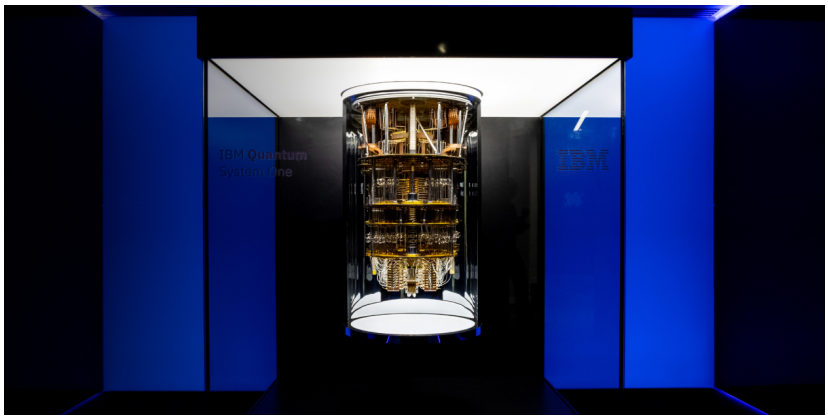
Ebben az időszakban több megközelítés is született a kvantumszámítógépekre, ezáltal több típusról is beszélhetünk, például:

1. Ioncsapdás: A kvantumbitek a csapdában lévő részecske állapotai alapján definiálódnak.

2. Szupravezetős: Szupravezető áramköröket használnak a qubitek létrehozására és manipulálására. Rendkívül alacsony hőmérsékleten, az abszolút nulla közelében működnek, hogy fenntartsák állapotukat. Például az IBM Q System One is ilyen.

3. Fotonikus: Fotonokat használnak kvantumbitként. Nagy sebességű kommunikációra és információfeldolgozásra képesek.

4. Quantum Annealing: Speciális kvantumszámítógép, Optimalizálási problémák megoldására tervezték.



1.1. ábra. IBM Q System One

1.3. Nehézségek

Mint láttuk, a kvantumszámítógépek jelenleg is kísérletezés alatt állnak, valamint számos kihívások továbbra is feltáratlanok maradtak. Egy kvantumszámítógép építése és karbantartása, azok speciális igényük és építőelemeik miatt mai napig nagy költségeket ölelnek fel.

A kvantumbitek minősége és stabilitása szintén egy jelentős probléma. A jelenlegi kvantumszámítógépek kevés qubittel rendelkeznek, skálázhatóságuk limitált, emiatt szűk körben használhatóak, számos komplex feladat megoldására továbbra is képtelenek.

Környezetre való érzékenysége miatt, extrém környezetekben tárolják őket. Az olyan tényezők, mint például a zaj, számítási problémákat okozhat, amik kijavítása rendkívül időigényes.

1.4. Előnyök és veszélyek

Nehézségei ellenére a kvantumszámítógépek számos lehetőséget is hordoznak magukban. Sokszor viszont ezek az előnyök rossz kezekben képesek veszélyeket is okozni.

Legfontosabb potenciáljuk a gyorsaságuk. Egy kvantumszámítógép sokkal gyorsabban képes elvégezni számításokat, mint a klasszikus számítógépek. Ez annak köszönhető, hogy egy klasszikus számítógép szekvenciálisan végzi feladatait, ezzel szemben a kvantumszámítógépek párhuzamosságra képesek.

Bár számos komplex feladat megoldására jelenleg is képtelenek, léteznek olyan összetettebb problémák, amiben a kvantumszámítógépek megfelelőnek bizonyultak, míg a klasszikus számítógépek számára megoldhatatlanok voltak. Ilyenek például különböző optimalizálási problémák, illetve faktoriális számítás. Ezek mögött számos kvantumalgoritmus jelenik meg, amely hasonlóan egy új ága a kvantuminformatikának.

A gyorsasági és számítási előnyei miatt több más iparág segítésére is szolgálhatnak a kvantumszámítógépek, többek között az anyag- és a gyógyszerkutatás is ide tartozik.

Viszont ezek az előnyök veszélyeket is hordozhatnak magukban. A már említett Shor algoritmus egy szám prímtényezős felbontását végzi el hatékonyan. A problémát az okozza, hogy a jelenlegi titkosítási algoritmusok (például az RSA) nagy része emiatt védtelen lesz, mivel ezek prímszámokat használnak titkosításra. Természetesen ezekre a problémákra is léteznek kutatások, így született a kvantumkriptográfia területe, ami fejlettebb biztonságot ígér.

A sok új lehetőség és a felsorolt előnyök nem léteznének a kvantum logikai kapuk nélkül, amik az alapkövei a kvantumszámítógépeknek.

2. fejezet

Klasszikus- és kvantum logikai kapuk

2.1. Klasszikus logikai kapuk és áramkörök

Elektromos impulzusokhoz értékeket társítunk, az alapján hogy küldtünk-e vagy sem. Ha érzékelünk, akkor ezt I logikai értéknek vagy 1 bitnek tekintjük. Ellenkező esetben H logikai értéknek, vagy 0 bitnek feleltetjük meg.

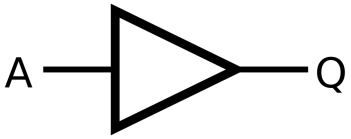
Ezekhez az impulzusokhoz többnyire logikai kapukat társítunk, melyek bináris operátorokat foglalnak magukban. A logikai kapuk alapvető építőkövei az elektronikának és számos célra használják őket. Összekapcsolásukkal áramköröket alakíthatunk ki, amik lineárisak és balról jobbra értelmezzük őket. A bal oldali vezetékek jelentik a bemenetet, míg a jobb oldaliak a kimenetet. Az ismertebb kapukhoz speciális ábrák és igazságtáblák tartoznak.

Az igazságtáblák a klasszikus logika alapvető eszközei, amelyek segítségével értelmezhetjük az adott műveleteket, valamint ellenőrizhetjük áramköreinket. Megmutatják az összes lehetséges bemeneti kombinációt, illetve a műveletek alkalmazása után a várható kimenetet is. Ezek a logikai műveleteket, kifejezéseket Boole-algebrának nevezzük, amely egy 19. századi matematikus, George Boole nevét viseli

2.1.1. Logikai kapuk

Buffer

A buffer kapuk kimenete megegyezik a kimenetükkel, egy biten értelmezzük. Jele: A



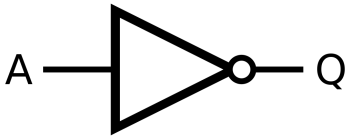
2.1. ábra. Buffer kapu rajza

A	A
0	0
1	1

2.1. táblázat. Buffer kapu igazságtáblája

NOT, vagy Negáció

A negáció kapu (vagy NOT) hasonlóan egy bites, a bemeneti jel logikai értékét megfordítja a kimeneten. Jele:$\neg A$



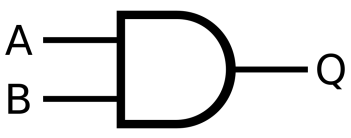
2.2. ábra. NOT kapu rajza

A	$\neg A$
0	1
1	0

2.2. táblázat. Negáció kapu igazságtáblája

AND, vagy Konjukció

Más néven logikai és. Kétbites művelet, kimenete akkor igaz, ha mind a két operandusa igaz. Jele: $A \wedge B$



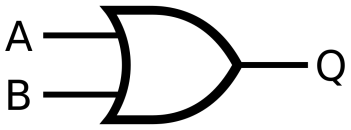
2.3. ábra. AND kapu rajza

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

2.3. táblázat. A konjukció igazságtáblája

OR, vagy Diszjunkció

Más néven logikai vagy. Szintén kétbites művelet, értéke csak akkor hamis, ha mind a két operandusa hamis. Jele: $A \vee B$



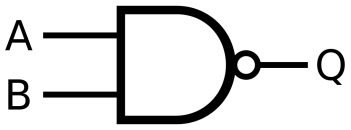
2.4. ábra. OR kapu rajza

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

2.4. táblázat. A diszjunkció igazságtáblája

NAND, vagy Negált konjukció

A konjukció negált változata, értéke akkor hamis, ha mindkét operandusa igaz. Jele: $\neg(A \wedge B)$



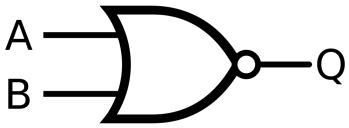
2.5. ábra. NAND kapu rajza

A	B	$\neg(A \wedge B)$
0	0	1
0	1	1
1	0	1
1	1	0

2.5. táblázat. A negált konjukció igazságtáblája

NOR, vagy Negált diszjunkció

A diszjunkció negált változata, értéke akkor igaz, ha mindkét operandusa hamis. Jele: $\neg(A \vee B)$



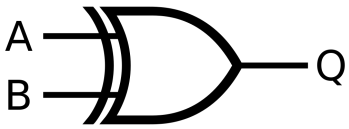
2.6. ábra. NOR kapu rajza

A	B	$\neg(A \vee B)$
0	0	1
0	1	0
1	0	0
1	1	0

2.6. táblázat. A negált diszjunkció igazságtáblája

XOR, vagy Exclusive OR

Más néven kizáró vagy. Értéke akkor hamis, ha a bemenetek megegyeznek. Jele: $A \oplus B$



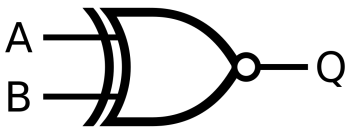
2.7. ábra. XOR kapu rajza

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

2.7. táblázat. A negált konjukció igazságtáblája

XNOR, vagy Exclusive NOR

A kizáró vagy negáltja, értéke akkor hamis, ha a bemenetek különböznek Jele: $\neg(A \oplus B)$



2.8. ábra. XNOR kapu rajza

A	B	$\neg(A \oplus B)$
0	0	1
0	1	0
1	0	0
1	1	1

2.8. táblázat. A negált konjukció igazságtáblája

2.1.2. Univerzális kapuk

Léteznek olyan logikai kapuk, amelyek képesek reprodukálni bármely másik működését. Ezek az univerzális kapuk lehetővé teszik akármelyik logikai függvény leképezését, ami azt jelenti, hogy akár tetszőleges áramkör is megvalósítható velük.

Bizonyítható, hogy bármilyen logikai függvény összeállítható csak NOT és AND, vagy NOT és OR függvények kombinációival. Mint láttuk, ezek a kapuk ötvözhetőek, erre szolgálnak a NAND és NOR kapuk.

Ebből következtethető, hogy bármilyen Boole-függvény megvalósítható olyan áramkörrel, amely csak NAND vagy NOR kapukat alkalmaz. Gyakran emiatt előnyt élveznek, mivel más kapuknak nincs ilyen tulajdonsága.

2.2. Kvantum logikai kapuk és áramkörök

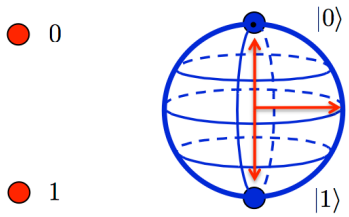
A kvantumkapuk jelentőségét a kvantummechanika alapelveire támaszkodva értelmezhetjük, amelyek olyan jelenségeket és viselkedéseket írnak le, melyek a klasszikus fizika keretein belül nem megfigyelhetők vagy magyarázhatók. A kvantummechanika alapján működő kapuk és áramkörök lehetővé teszik olyan számítások végrehajtását, amelyek rendkívül nagy számítási kapacitással és párhuzamosítással rendelkeznek, ami jelentős előnyöket kínál a hagyományos, klasszikus számítógépekkel szemben.

Bár vannak hasonlóságok a klasszikus változatokhoz képest, a kvantum logikai kapuk egy sokkal komplexebb matematikai hátteret és fizikai jelenségeket hordoznak magukban.

Jelen alfejezet csak pár, ismertebb logikai kaput tartalmaz, de fontos megemlíteni, hogy a kvantumkapuk száma végtelen. Mivel szakdolgozatom az egyqubites kapukra irányul, ezért a több kvantumbites változatok csak megemlítő jelleggel szerepelnek.

2.2.1. Qubit

A kvantumbit, röviden qubit, a kvantuminformatika alapvető építőeleme, és a klasszikus számítógépek működésétől eltérő, kvantummechanikai alapokon nyugvó adatábrázolási egység. Míg a hagyományos bit csak két állapotban (0 vagy 1) lehet jelen, a qubit képes egyszerre számos állapotban lenni, ami egyike annak a tulajdonságának, amely a kvantuminformatikát annyira különlegessé teszi.



2.9. ábra. A bit és qubit reprezentációja

Ahogy az a 2.9. ábrán látható, a qubit számos állapotát egy úgynevezett Bloch gömbön ábrázoljuk, északi pólusán $|0\rangle$, déli pólusán pedig $|1\rangle$ helyezkedik el. A $|\rangle$ és $\langle|$ jelöléseket braket-nek nevezzük, de Dirac jelölésként is ismert, és a kvantumállapotok jelölésében segít. A ket ($|\Psi\rangle$) egy oszlopvektort, a bra ($\langle\Psi|$) pedig egy sorvektort jelöl.

A kvantummechanika alapelvei és a qubitok sajátos tulajdonságai lehetővé teszik olyan számítási feladatok elvégzését, amelyek a hagyományos számítógépek számára gyakorlatilag lehetetlenek lennének.

2.2.2. Szuperpozíció és összefonódás

Egy qubit állapotát a következőképpen adhatjuk meg:

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{2.1}$$

ahol α és β komplex számok, valamint teljesül, hogy $|\alpha|^2+|\beta|^2 = 1$. Ez azt jelenti, hogy a $|\Psi\rangle$ kvantumbit $|\alpha|^2$ eséllyel 0, $|\beta|^2$-el pedig 1 lesz. Ezt szuperpozíciónak nevezzük. A két szélső érték, azaz $|0\rangle$ vagy $|1\rangle$ akkor áll elő, ha α 0 vagy 1, β pedig ennek ellentettje. A szuperpozíció legjobban a Schrödinger macskája gondolatkísérlettel prezentálható, melyet Erwin Schrödinger fogalmazott meg 1935-ben.

A qubit másik fontos jelensége a kvantum-összefonódás. Ez egy olyan jelenség a kvantummechanikában, amelyben két vagy több részecske állapota olyan összekapcsolt módon van, hogy az egyiken végzett mérések azonnal hatnak a másikra, függetlenül attól, hogy a részecskék milyen fizikai távolságra vannak egymástól.

2.2.3. Qubit mérése

Méréskor eredményként egyetlen klasszikus bitet kapunk. A qubit mérése során az eredmény kvantumállapota "összeomlik" az adott értékre, amelyet a mérés során megfigyeltünk. Azaz elveszti a szuperpozícióját és az összefonódottságát, ami azt jelenti, hogy a mérés után a rendszer elveszíti a kvantumos jellegét, és klasszikus állapotba kerül. Ez az összeomlás a kvantummechanika alapvető jelensége, amely lehetővé teszi a kvantumrendszer állapotának rögzítését a mérés pillanatában. Emiatt a mérések kritikus szerepet játszanak.

2.10. ábra. Mérés rajza

2.2.4. Kvantum logikai kapuk

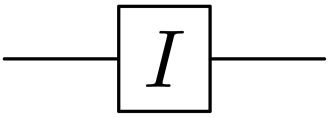
I, vagy Identity

A buffer kapuhoz hasonlóan nem végez műveletet, helyben hagyja a kvantumbiteket. Egy qubiten használjuk. Azonosság transzformációként működik, alakja:

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.2}$$

Ez egy tetszőleges kvantumbiten:

$$|\Psi\rangle = I\,|\varphi\rangle \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = a\,|0\rangle + b\,|1\rangle \tag{2.3}$$



2.11. ábra. I kapu rajza

X, vagy Pauli-X

A klasszikus NOT kapunak felel meg, bit-flipnek is hívják. A Bloch gömb X tengelyére tükröz. Egy qubiten működik. Alakja:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{2.4}$$

Ez egy tetszőleges kvantumbiten:

$$|\Psi\rangle = X\,|\varphi\rangle \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = b\,|0\rangle + a\,|1\rangle \tag{2.5}$$

Az X kapu hatására a valószínűségi amplitúdók cserélődnek. Inverze saját maga, tehát például:

$$|\Psi\rangle = XX\,|0\rangle = |0\rangle \tag{2.6}$$

2.12. ábra. X kapu rajza

Y, vagy Pauli-Y

Az X kapuhoz hasonlóan felcseréli a $|0\rangle$-t és az $|1\rangle$-t, viszont az Y kapu a relatív fázist is megváltoztatja. A Bloch gömb Y tengelyére tükröz. Egy qubiten működik. Alakja:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \tag{2.7}$$

Ez egy tetszőleges kvantumbiten:

$$|\Psi\rangle = Y\,|\varphi\rangle \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = -ib\,|0\rangle + ia\,|1\rangle\,, \tag{2.8}$$

ahol i komplex szám. Inverze saját maga, tehát például:

$$|\Psi\rangle = YY\,|0\rangle = |0\rangle \tag{2.9}$$

2.13. ábra. Y kapu rajza

Z, vagy Pauli-Z

Phase-flip kapunak is hívják, ebből adódóan a fázist cseréli meg. A Bloch gömb Z tengelyére tükröz. Egy qubiten működik. Alakja:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \tag{2.10}$$

Ez egy tetszőleges kvantumbiten:

$$|\Psi\rangle = Z\,|\varphi\rangle \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = a\,|0\rangle - b\,|1\rangle \tag{2.11}$$

Látható, hogy a Z kapu csak az $|1\rangle$ valószínűségi amplitúdót változtatja. Inverze saját maga, tehát például:

$$|\Psi\rangle = ZZ\,|0\rangle = |0\rangle \tag{2.12}$$

2.14. ábra. Z kapu rajza

H, vagy Hadamard

A Hadamard kapu segítségével a már említett szuperpozíciós állapotokat lehet előállítani. Egy qubiten működik. Alakja:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2.13}$$

Ez egy tetszőleges kvantumbiten:

$$|\Psi\rangle = H\,|\varphi\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \frac{a+b}{\sqrt{2}}\,|0\rangle + \frac{a-b}{\sqrt{2}}\,|1\rangle \tag{2.14}$$

Mivel a kaput gyakran használják a standard bázisvektorokon, emiatt külön számon tartják őket:

$$|\Psi\rangle = H\,|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \tag{2.15}$$

$$|\Psi\rangle = H\,|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{2.16}$$

A Hadamard kapu inverze saját maga, tehát például:

$$|\Psi\rangle = HH\,|0\rangle = |0\rangle \tag{2.17}$$

2.15. ábra. H kapu rajza

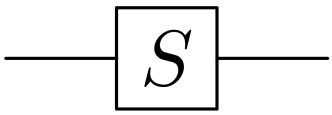
S, vagy Phase

Más néven fáziseltolási kapu. $\pi/2$ fáziseltolást alkalmaz a qubit állapotára. A Bloch gömb körül az óramutató járásával megegyező irányban forgat $\pi/2$ radiánnal. Egy qubiten működik. Alakja:

$$S = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix} \tag{2.18}$$

Ez egy tetszőleges kvantumbiten:

$$|\Psi\rangle = S\,|\varphi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = a\,|0\rangle - ib\,|1\rangle \tag{2.19}$$



2.16. ábra. S kapu rajza

T, vagy $\pi/8$

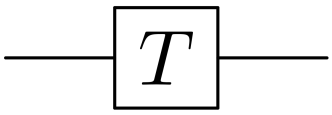
A T kapu, hasonlóan az S kapuhoz, fáziseltolást alkalmaz egy qubit állapotára. Az állapotvektort $\pi/4$ radiánnal forgat a Bloch gömb körül, óramutató járással megegyező irányban. Alakja:

$$T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \tag{2.20}$$

Ez egy tetszőleges kvantumbiten:

$$|\Psi\rangle = T\,|\varphi\rangle = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = a\,|0\rangle + e^{i\pi/4} b\,|1\rangle \tag{2.21}$$

Az S kapuval való azonosságok miatt elmondható, hogy $S = T^2$

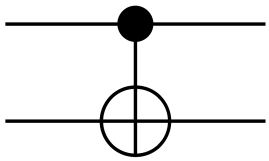


2.17. ábra. T kapu rajza

CNOT, vagy Controlled Not

A CNOT kapu két qubites művelet, ahol az első qubit vezérlő-, a második pedig cél kvantumbit néven ismert. Ha a vezérlő $|1\rangle$, akkor a cél qubiten X kaput használ, a többi változatlan. Amennyiben a vezérlő qubit $|0\rangle$, a cél változatlan marad. Klasszikus logikai kapuként is alkalmazható. Inverze saját maga. Alakja:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.22}$$

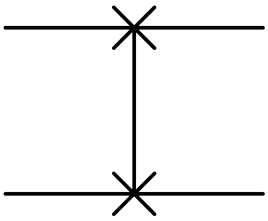


2.18. ábra. CNOT kapu rajza

SWAP

Két qubites kvantumkapu, mely felcseréli a bemenetek állapotát. Mivel három CNOT kapuval valósítható meg, így más jelölést is alkalmaznak rá, az egyszerűbb változat a 2.19. ábrán látható. Inverze saját maga. Alakja:

$$SWAP = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.23}$$

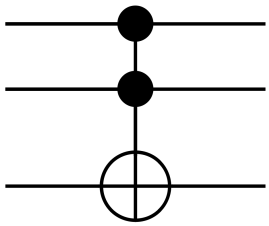


2.19. ábra. SWAP kapu rajza

Toffoli

Ahogy az a 2.20. ábrán látható, a Toffoli kapu CNOT műveletet használ, ezért szokás CCNOT kapunak is nevezni. Három qubiten működik, az első kettő vezérlő, az utolsó pedig a cél qubit. Klasszikus logikai kapuként is alkalmazható, ez esetben univerzális klasszikus kapunak tekintik. Inverze saját maga. Alakja:

$$Toffoli = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.24}$$



2.20. ábra. Toffoli kapu rajza

2.2.5. Univerzális kvantumkapuk

Klasszikus esetben tapasztaltuk, hogy léteznek univerzális kapuk. Ez amiatt lehetséges, mivel csak végesen sok Boole-függvény van egy adott számú változó esetén. Kvantumkapuknál elmondható, hogy a lehetséges kapuk száma végtelen sok. Emiatt a fontos különbség miatt univerzális kvantumkapuk halmaza sem létezik. Ellenben, ha a kapuknak vesszük egy véges számú gyűjteményét, akkor beszélhetünk univerzális megoldásokról, viszont ezek sem használhatóak minden lehetséges kvantumáramkörre, csak hozzávetőlegesen.

2.3. Reverzibilis kapuk

Megfordítható kapuknak is nevezik őket. Két alapvető tulajdonsága van:

1. Az eredményükből egyértelműen kikövetkeztethetőek a bemenetek, azaz egy bemeneti kombinációhoz pontosan egy kimeneti kombináció tartozik, és fordítva

2. Lehetővé teszik a bemenetek helyreállítását a kimenetekből anélkül, hogy információt veszítenének.

Vegyük például az OR kaput. Mint láttuk, értéke csak akkor hamis, ha bemenetei hamisak. Viszont minden más esetben nem tudjuk megállapítani, hogy melyik volt az aktuális bemenet a háromból, hacsak nincs további információnk. Tehát az OR kapuról elmondható, hogy irreverzibilis (nem reverzibilis) kapu. A legtöbb klasszikus logikai kapu szintén ebbe a csoportba tartozik, a már említett kapuk közül ez alól a CNOT és a Toffoli kivétel.

Lehetőség van klasszikus logikai művelet végrehajtására csak reverzibilis kapuk használatával, de ilyen esetekben gyakran szükségessé válik a segédbitek használata.

Alapvetően bármilyen reverzibilis klasszikus kapu megvalósítható kvantumszámítógépeken, így emiatt létezik a Toffoli és CNOT kapunak kvantum változata is.

A klasszikus logikai kapukkal ellentétben, a kvantum kapuk közül mindegyik reverzibilis.

3. fejezet

Tanulás segítő program és applikáció

3.1. Technológiai körültekintés

A különböző technológiák és eszközök közötti választás nem mindig egyértelmű. Ezért szükséges részletesen mérlegelni és összehasonlítani az elérhető lehetőségeket. A nyelv, keretrendszer vagy platform kiválasztása előtt érdemes megfontolni a különböző szempontokat, mint például a teljesítmény, a fejlesztési idő és a támogatottság. Szakdolgozatom írásának kezdetekor, a nyelv és keretrendszer választásnál figyelembe vettem az alábbi elképzeléseket:

1. Két részből álljon: egy telefonos applikációból és egy asztali alkalmazásból, mindkettő angol nyelven.

2. Az egy kvantumbites kapuk egy csoportját mutassa be, ezeket interaktív, tanulást segítő formában.

3. A telefonos applikáció képes legyen csatlakozni az asztali alkalmazásra, majd szenzor adatokat küldeni. Fontos volt továbbá, hogy rövid ismertetőket tudjon a felhasználó olvasni az adott kapukról.

4. A program egy letisztult felületet nyújtson, ahol a felhasználó válogathat az adott egy kvantumbites kapuk közül. Ezekről kapja meg azt a leírást, ami a telefonon is elérhető, valamint animációkat is tudjon nézni róluk.

5. A program középpontja egy Bloch gömb legyen, egy állapotvektorral. Ezt a felhasználó tudja állítani, valamint amennyiben csatlakoztatott telefont, ki is tudja próbálni az adott kapukat. A telefon forgatásával képes legyen a cél állapotba igazítani a vektort, ez egy minimális hibatűréssel történjen.

A program lehetőségeit szűkítette a tény, hogy minél pontosabb ábrákat, reprezentációkat kellett keresnem. A gömb forgatását vettem a legmeghatározóbb tényezőnek választásnál. Egy egyszerű 3D-s objektumot több nyelv és keretrendszer is támogat. Kipróbáltam, hogy ez Java Fx-ben valamint PythonQt-ban hogy jelenik meg. Viszont, mint ahogy azt a 2.9. ábrán is látható, a Bloch gömb tartalmaz egy vektort, aminek pozíciója változhat, ami miatt a klasszikus megjelenítés nem elegendő, hiszen nem az objektumot forgatjuk. Emiatt fontos volt áttekintenem, hogy milyen kvantumáramkörökkel foglalkozó könyvtárakkal szolgálnak a keretrendszerek.

A telefonos applikáció kapcsán fontos volt, hogy a választott operációs rendszer minél több felhasználót lefedjen, valamint közel pontos szenzor adatot legyen képes küldeni.

3.2. Választott technológiák

3.2.1. Asztali alkalmazás

Python

A Python egy magas szintű, interpretált programozási nyelv, amely egyszerűségéről és könnyen olvashatóságáról ismert. Manapság a legnépszerűbb nyelv.

Szintaxisát úgy tervezték, hogy intuitív és olvasható legyen, így kezdők és tapasztalt programozók számára egyaránt nagyszerű választás. Hangsúlyozza a kód olvashatóságát, és lehetővé teszi a fejlesztők számára, hogy más nyelvekhez képest kevesebb kódsorban oldjanak meg problémákat.

A Python emellett egy interpreteres nyelv, ami azt jelenti, hogy a Pythonban írt kód soronként hajtódik végre, nem pedig előzetesen gépi kódba fordítva. Ez megkönnyíti a fejlesztést és a hibakeresést, mivel a kód azonnal végrehajtható külön fordítási lépés nélkül.

Dinamikus típusrendszerének köszönhetően, változótípusokra a rendszer futás közben következtet. Ez nagyobb rugalmasságot és gyorsabb fejlesztést tesz lehetővé.

A Python támogatja az objektum-orientált programozás (OOP) elveit, lehetővé téve a fejlesztők számára, hogy osztályokat és objektumokat hozzanak létre az adatok és a viselkedések egységbe zárásához.

Egy átfogó könyvtárral is rendelkezik, amely modulokat és funkciókat biztosít különféle feladatok végrehajtásához, például hálózatkezelés, matematikai kifejezések stb.

Mindezek miatt a Python nyelv rendkívül sokoldalú. Számos alkalmazáshoz használják, beleértve a webfejlesztést, adatelemzést, gépi tanulást, mesterséges intelligenciát, tudományos számítástechnikát, automatizálást stb.

PySide modul

A PySide lehetővé teszi a fejlesztők számára, hogy Qt alapú alkalmazásokat írjanak Python használatával, hozzáférést biztosítva a Qt keretrendszer összes funkciójához és szolgáltatásához. Előnyei:

1. Cross-platform: Hasonlóan a Qt-hoz, a Pyside is cross-platform, tehát az ezzel fejlesztett alkalmazások számos operációs rendszeren futhatnak, beleértve a Windowst, macOS-t és a Linuxot, anélkül, hogy a kódon jelentős változtatásokat kellene végrehajtani.

2. GUI fejlesztés: A PySide leegyszerűsíti a grafikus felhasználói felületek (GUI) fejlesztését widgetek létrehozására, elrendezésekre és események kezelésére, a Qt hatékony eszközkészletének segítségével.

Az alkalmazás Pyside6-ot használ, ami Qt6-ot támogat

Qiskit könyvtár

A legjobban elterjedt nyílt forráskódú Python könyvtár ami kvantumáramkörökkel foglalkozik, az IBM fejleszti. Segítségével különböző áramköröket alakíthatunk ki, és szimulálhatunk. Lehetőség van az adott qubit állapotokat Bloch gömbön is reprezentálni. Hasonló könyvtárakhoz képest sokkal kvantumáramkör és kvantumszámítás központúbb, ami az asztali alkalmazás magját adja.

Matplotlib könyvtár

A Qiskit-ben kialakított áramköröket és Bloch gömböket a Matplotlib könyvtárral könnyen meg lehet jeleníteni. A könyvtár széles körben biztosít megjelenítést különböző diagramokhoz, 2D és 3D objektumokhoz, és még sok máshoz. A Matplotlib egy népszerű Python-könyvtár, amelyet statikus, interaktív és animált vizualizációk létrehozására használnak Pythonban. Széles körben használják különféle területeken.

A Matplotlib könyvtárral előállított objektumokat egyszerűen hozzá lehet adni a PySide projektekhez.

Numpy könyvtár

A NumPy egy alapvető csomag a Python segítségével végzett komplex számításokhoz. Támogatja a többdimenziós tömböket és mátrixokat, valamint matematikai függvények gyűjteményét biztosítja az ezeken a tömbökön való műveletekhez.

Ahogy azt a 2.2.4. fejezetben láttuk, a komplex mátrix számítások elengedhetetlenek a kvantumlogikai kapuk prezentációjában.

Socket könyvtár

Pythonban a socket könyvtár alacsony szintű hálózati interfészt biztosít, amely lehetővé teszi a hálózati socketek létrehozását és az azokkal való interakciót. Megkönnyíti mind a kliens, mind a szerver hálózati alkalmazások létrehozását. Támogatja a TCP és UDP protokollokat.

Ez könyvtár a telefon csatlakoztatása miatt szükséges.

3.2.2. Applikáció

Android Studio

Android-alkalmazások fejlesztésének hivatalos integrált fejlesztői környezete (IDE). A Google fejlesztette, és a JetBrains IntelliJ IDEA szoftverén alapul.

Felhasználóbarát felülettel rendelkezik, amely különböző paneleket tartalmaz a kód szerkesztéséhez, az elrendezések megtervezéséhez, a projektfájlok kezeléséhez, valamint a naplók és hibák megtekintéséhez.

A felületet egy XML Layout Editor is segíti. Ez egy vizuális szerkesztőt tartalmaz az Android-alkalmazások felületeinek megtervezéséhez, XML-fájlok használatával. A fejlesztők behúzhatják a felhasználói felület összetevőit egy vászonra, és testre szabhatják tulajdonságaikat a Properties panelen.

Az Android Studio legfontosabb eleme a beépített emulátor, amely lehetővé teszi a fejlesztők számára, hogy teszteljék alkalmazásaikat virtuális Android-eszközökön, különböző képernyőmérettel, felbontással és Android-verziókkal. Támogatja továbbá a hibakeresést fejlesztői számítógéphez USB-n keresztül csatlakoztatott fizikai Android-eszközökön. A fejlesztők beállíthatnak töréspontokat, ellenőrizhetik a változókat, és valós időben elemezhetik az alkalmazások teljesítményét.

Az Andorid Studio-hoz tartozik továbbá egy Gradle rendszer, ami lehetővé teszi a fejlesztők számára, hogy egyedi build konfigurációkat és függőségeket határozzanak meg projektjeikhez.

Mivel általánosságban elmondható, hogy a világon jelenleg az Android felhasználók vannak többségben, ezért ragaszkodtam ehhez a megoldáshoz, más operációs rendszerek helyett.

Java

A Java egy széles körben használt, magas szintű objektum-orientált programozási nyelv, amelyet a Sun Microsystems fejlesztett ki az 1990-es évek közepén.

A Java szintaxis hasonló a C nyelvekhez, így az ezeket ismerő fejlesztők viszonylag könnyen megtanulhatják.

A Java egy átfogó könyvtárral rendelkezik. Java Development Kit (JDK) néven ismert, amely számos segítséget kínál különféle feladatokhoz. A könyvtár segít a fejlesztőknek robusztus és funkciókban gazdag alkalmazások hatékonyabb felépítésében.

Ezek mellett széles közössége és támogatottsága miatt a telefonos applikáció fejlesztésénél számomra egyértelmű volt a Java használata.

3.3. Program bemutatása

3.4. Applikáció bemutatása

3.5. Tesztelések

3.6. Továbbfejlesztési lehetőségek

Összegzés

Irodalomjegyzék

Nyilatkozat

Alulírott ....., büntetőjogi felelősségem tudatában kijelentem, hogy az általam benyújtott, ... című szakdolgozat önálló szellemi termékem. Amennyiben mások munkáját felhasználtam, azokra megfelelően hivatkozom, beleértve a nyomtatott és az internetes forrásokat is.

Aláírásommal igazolom, hogy az elektronikusan feltöltött és a papíralapú szakdolgozatom formai és tartalmi szempontból mindenben megegyezik.

Eger, 2021. szeptember 25.

aláírás

A *Nyilatkozatot* kitöltve nyomtassa ki, írja alá,
majd szkennelve tegye ennek a helyére!