Quantum NV Sieve 개선

https://youtu.be/NSHziEblrwY

개선 요소

1. 가장 큰 복잡도를 갖는 곱셈 과정을 위해 덧셈기 변경 + AND gate 사용

- 이전 구현은 알고리즘 완성이 목표여서 최적화를 크게 신경 안 씀
- 이번 개선에서는 **depth 감소에 집중하기 위해 out of place 방식의 Draper adder 사용**
- **Toffoli depth 1**인 **AND gate** 적용

2. Fixed-point 연산 적용

- **더 정밀한 격자 감소 및 연산 (sieve factor인 γ 및 모든 벡터에 대해)**
  - 격자 감소 범위를 조금 더 정밀하게 잡을 수 있음
    - **클래식 구현에서 사용되는 sieve factor는 0.97이며 1에 가까울수록 좋음**
    - **그러나 기존 구현에서는 소수점 단위로 설정할 수 없었음 → 개선**
- **Quantum에서 Floating point 연산보다 효율적이라고 함**

Draper adder (out of place)

- **In place 구현은 out of place에 비해 상대적으로 Toffoli depth가 높음**
  - 이번 개선에서는 depth 최적화를 위해 **Cuccaro CDKM 대신 out of place + Draper adder 사용**

Adder	Year	Toffoli Count	Toffoli Depth	Qubit Count
VBE RCA [14]	1995	$4n-2$	$4n-2$	$3n+1$
Cuccaro RCA [2]	2004	$2n-1$	$2n-1$	$2n+2$
Draper In-place CLA [3]	2004	$10n-3\omega(n)-3\omega(n-1)$ $-3\lfloor\log n\rfloor-3\lfloor\log(n-1)\rfloor-7$	$8+\lfloor\log n\rfloor+\lfloor\log(n-1)\rfloor$ $+\lfloor\log\frac{n}{3}\rfloor+\lfloor\log\frac{n-1}{3}\rfloor$	$4n-\omega(n)-\lfloor\log n\rfloor$
Draper Out-of-place CLA [3]	2004	$5n-3\omega(n)-3\lfloor\log n\rfloor-1$	$4+\lfloor\log n\rfloor+\lfloor\log\frac{n}{3}\rfloor$	$4n+1-\omega(n)-\lfloor\log n\rfloor$

- **Quantum NV Sieve에서 Toffoli gate가 사용되는 함수 (아래 함수들에서 Toffoli, 덧셈기 사용됨)**
  - **Rank번 호출되는 함수**
    - COMPLEMENT_pos
    - outDraper_adder
    - COMPLEMENT_neg
    - Muliplication
  - **1번 호출되는 함수 (Rank 관련 x)**
    - CSA_Draper
    - COMPLEMENT_rR_c
    - outDraper_adder

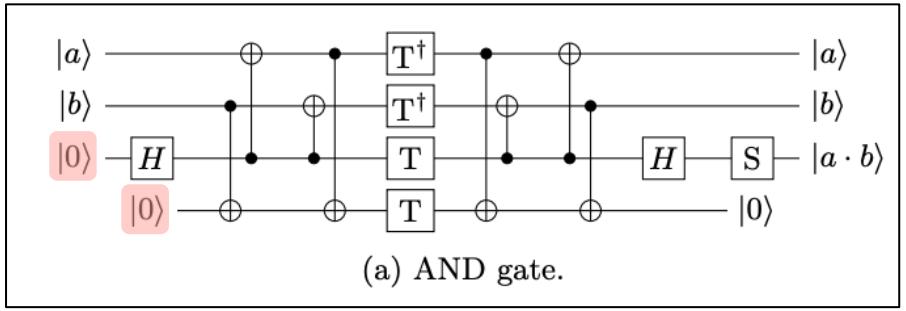
각 함수마다 연산 대상이 다르며 Toffoli gate에 따라 Td 결정, AND gate 적용 시의 실제 Td는 아래와 같이 계산

$$Td = (3\times(Rank\times bitsize)) + Rank\times(bit\_size^2\times sqr\_bit\_size) + sqr\_bit\_size + sqr\_bit\_size + sqr\_bit\_size$$

AND gate 적용

- **AND gate는 대상 큐비트가 0이라면, Toffoli gate와 동일하게 동작**

c
ancilla



(a) AND gate.

```
def quantum_and(eng, a, b, c):
    ancilla = eng.allocate_qubit()
    H | c
    CNOT | (b, ancilla)
    CNOT | (c, a)
    CNOT | (c, b)
    CNOT | (a, ancilla)
    Tdag | a
    Tdag | b
    T | c
    T | ancilla
    CNOT | (a, ancilla)
    CNOT | (c, b)
    CNOT | (c, a)
    CNOT | (b, ancilla)
    H | c
    S | c
```

- **개선 구현에서는 Out of Place 방식의 덧셈기를 사용했으며, 타겟 큐비트가 새로 할당되었으므로 0의 상태**

  → 따라서 큰 구조적 변경 없이 ancilla qubit 추가하여 AND gate가 적용된 덧셈기 사용

자원 추정

- **현재 QIP 심사 중인 버전에 자원 추정이 누락된 부분 발견..** 😒
  - 리뷰 받을 때 수정 예정
  - 개선 버전에서는 T-depth 정확하게 계산하도록...

자원 추정

- **Takahashi adder (In place) + Toffoli gate (T-depth 4) 적용 vs Draper adder (Out of place) + AND gate (T-depth 1) 적용**
  - **#CNOT, #1qCliff:** 크게 변동 없음
  - **#T gate:** 모든 경우에서 약 2배 감소
  - ***FD*:** 최소 2배 ~ 최대 16배 감소
  - ***Td*:** 이전 버전이 잘못 계산 되었음 (Toffoli가 쓰이는 모든 곳에 대한 depth가 아니라, Mul에 대한 Td만 계산함..)

Table 2: Resource Estimation of quantum NV Sieve oracle (R10D10 means the rank and the dimension of the lattice are 10).

Case	#CNOT	#1qCliff	#T	T-depth (Td)	Full depth (FD)	Qubit (M)	Td-M	FD-M
R10D10	$2^{16.1767}$	$2^{13.9067}$	$2^{15.7118}$	$2^{7.6037}$	$2^{11.5264}$	$2^{12.5454}$	$2^{20.1491}$	$2^{24.0718}$
R20D20	$2^{18.9097}$	$2^{16.6143}$	$2^{18.4212}$	$2^{8.4470}$	$2^{14.2190}$	$2^{15.2640}$	$2^{23.7110}$	$2^{29.4830}$
R30D30	$2^{20.5672}$	$2^{18.2624}$	$2^{20.0695}$	$2^{8.9454}$	$2^{15.2810}$	$2^{16.9202}$	$2^{25.8656}$	$2^{32.2012}$
R40D40	$2^{21.7859}$	$2^{19.4808}$	$2^{21.2880}$	$2^{9.3531}$	$2^{16.0566}$	$2^{18.1329}$	$2^{27.4860}$	$2^{34.1896}$
R50D50	$2^{22.6998}$	$2^{20.3885}$	$2^{22.1958}$	$2^{9.6165}$	$2^{16.6674}$	$2^{19.0527}$	$2^{28.6692}$	$2^{35.7201}$
R60D60	$2^{23.4836}$	$2^{21.1729}$	$2^{22.9802}$	$2^{9.8595}$	$2^{17.1715}$	$2^{19.8329}$	$2^{29.6924}$	$2^{37.0044}$
R70D70	$2^{24.1348}$	$2^{21.8228}$	$2^{23.6301}$	$2^{10.0660}$	$2^{17.6005}$	$2^{20.4848}$	$2^{30.5508}$	$2^{38.0853}$

무시

Table 3: Quantum cost for Grover's search on quantum NV Sieve.

Case	#Total gates	T-depth (Td)	Full depth (FD)	Qubit (M)	Quantum cost	Td-M	FD-M
R10D10	$2^{18.1267}$	$2^{8.6073}$	$2^{12.5264}$	$2^{12.5456}$	$2^{30.6532} \cdot r$	$2^{21.1529}$	$2^{25.0720}$
R20D20	$2^{20.8481}$	$2^{9.4470}$	$2^{15.2190}$	$2^{15.2640}$	$2^{35.0664} \cdot r$	$2^{24.7110}$	$2^{32.4830}$
R30D30	$2^{22.5012}$	$2^{9.9454}$	$2^{16.2810}$	$2^{16.9202}$	$2^{37.7823} \cdot r$	$2^{26.8656}$	$2^{33.2012}$
R40D40	$2^{23.7199}$	$2^{10.3531}$	$2^{17.0566}$	$2^{18.1329}$	$2^{39.7765} \cdot r$	$2^{28.4860}$	$2^{35.1895}$
R50D50	$2^{24.6308}$	$2^{10.6165}$	$2^{17.6674}$	$2^{19.0527}$	$2^{41.2938} \cdot r$	$2^{29.6692}$	$2^{36.7201}$
R60D60	$2^{25.4150}$	$2^{10.8595}$	$2^{18.1715}$	$2^{19.8329}$	$2^{42.5865} \cdot r$	$2^{30.6924}$	$2^{38.0044}$
R70D70	$2^{26.0655}$	$2^{11.0660}$	$2^{18.6005}$	$2^{20.4848}$	$2^{43.6661} \cdot r$	$2^{31.5509}$	$2^{39.0853}$

무시

Table 2: Resource Estimation of quantum NV Sieve oracle (R10D10 means the rank and the dimension of the lattice are 10).

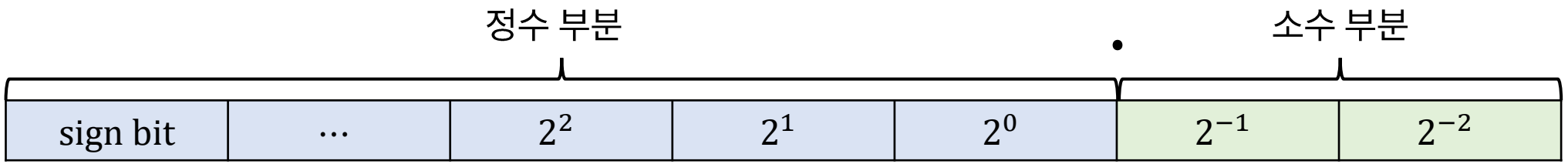
Case	#CNOT	#1qCliff	#T	T-depth (Td)	Full depth (FD)	Qubit (M)	Td-M	FD-M
R10D10	$2^{16.0685}$	$2^{14.2921}$	$2^{14.7054}$	$2^{11.0443}$	$2^{10.5565}$	$2^{10.6329}$	$2^{21.6772}$	$2^{21.2458}$
R20D20	$2^{18.6200}$	$2^{16.7937}$	$2^{17.2083}$	$2^{13.5521}$	$2^{11.6821}$	$2^{12.4424}$	$2^{25.9945}$	$2^{24.1245}$
R30D30	$2^{20.2122}$	$2^{18.3524}$	$2^{18.7801}$	$2^{15.1241}$	$2^{12.4795}$	$2^{13.5543}$	$2^{28.6784}$	$2^{26.0238}$
R40D40	$2^{21.3774}$	$2^{19.5197}$	$2^{19.9337}$	$2^{16.2734}$	$2^{12.8570}$	$2^{14.3445}$	$2^{30.6179}$	$2^{27.2015}$
R50D50	$2^{22.2717}$	$2^{20.4010}$	$2^{20.8160}$	$2^{17.1802}$	$2^{13.1656}$	$2^{14.9614}$	$2^{32.1416}$	$2^{28.1271}$
R60D60	$2^{23.0307}$	$2^{21.1551}$	$2^{21.5701}$	$2^{17.9293}$	$2^{13.4146}$	$2^{15.4768}$	$2^{33.4061}$	$2^{28.8914}$
R70D70	$2^{23.6705}$	$2^{21.7907}$	$2^{22.2057}$	$2^{18.5675}$	$2^{13.8495}$	$2^{15.9188}$	$2^{34.4863}$	$2^{29.7603}$

Table 3: Quantum cost for Grover's search on quantum NV Sieve.

Case	#Total gates	T-depth (Td)	Full depth (FD)	Qubit (M)	Quantum cost	Td-M	FD-M
R10D10	$2^{16.8175}$	$2^{11.0443}$	$2^{10.5565}$	$2^{10.6329}$	$2^{27.3740} \cdot r$	$2^{21.6772}$	$2^{21.2458}$
R20D20	$2^{19.3494}$	$2^{13.5521}$	$2^{11.6821}$	$2^{12.4424}$	$2^{31.0315} \cdot r$	$2^{25.9945}$	$2^{24.1245}$
R30D30	$2^{20.9313}$	$2^{15.1241}$	$2^{12.4795}$	$2^{13.5543}$	$2^{33.4108} \cdot r$	$2^{28.6784}$	$2^{26.0238}$
R40D40	$2^{22.0942}$	$2^{16.2734}$	$2^{12.8570}$	$2^{14.3445}$	$2^{34.9512} \cdot r$	$2^{30.6179}$	$2^{27.2015}$
R50D50	$2^{22.9837}$	$2^{17.1802}$	$2^{13.1656}$	$2^{14.9614}$	$2^{36.1493} \cdot r$	$2^{32.1416}$	$2^{28.1271}$
R60D60	$2^{23.7407}$	$2^{17.9293}$	$2^{13.4146}$	$2^{15.4768}$	$2^{37.1553} \cdot r$	$2^{33.4061}$	$2^{28.8914}$
R70D70	$2^{24.0570}$	$2^{18.5675}$	$2^{13.8495}$	$2^{15.9188}$	$2^{37.9065} \cdot r$	$2^{34.4863}$	$2^{29.7603}$

Fixed-point

- **고정 소수점 연산**
  - 고정된 자릿수를 사용 (정수, 소수)
  - **사칙연산에 있어 효율적** (정수 연산과 동일하게 수행 가능)
  - 적은 수의 비트를 사용함
- **격자의 Dimension과 별개로 정밀도를 위해 소수파트 추가**
  - 즉, R50D50인 경우, 정수 부분의 길이가 50-bit, **소수 부분은 별개로 큐비트 할당**



Fixed-point arithmetic in quantum

- **Addition**
  - 기존 덧셈 방식과 동일
- **Multiplication**
  - 기존 곱셈 방식과 동일하게 Toffoli gate 사용
  - **소수 부분 ($m\ qubit$)이 존재하기 때문에 곱셈 후 해당 부분을 위한 큐비트도 2배로 증가 ($2m\ qubit$)**
    - 정수 부분: $n\ qubit \rightarrow 2n\ qubit$, 소수 부분: $m\ qubit \rightarrow 2m\ qubit$
- **2's Complement**

1. 기존과 동일하게 최상위 비트 기준으로 양/음수 판단하여 **조건에 맞게 모든 비트 반전**
2. 소수 부분을 제외한 **정수 부분의 LSB에만 1을 더해줌** (for 2의 보수)

알고리즘 구현 정확성 테스트

- **소수자리 사용하지 않는 경우 in 고정 소수점 버전**
  - 이전 구현: 1011000 → -40

    → 심사 중인 논문에 단계별 결과 값이 있으며, **해당 벡터와 동일한 벡터 사용**
  - 고정 소수점 구현 결과 값: 1011000.0000 → -40.0000

    → 정상 동작
- **소수자리 사용하는 경우 in 고정 소수점 버전**
  - 입력 벡터: $v = (5.5, 1.0, 1.0, 3.0, 1.0)$, $c = (6.0, 4.0, 4.0, 1.0, 8.0), sqr\ \gamma R = 32.0$ (나머지 벡터는 0)
  - 고정 소수점 구현 결과 값: → 1010111.0100 → -39.25

    1010111.0100 → 0101000.1011 → 0101000.1100 → -40+0.75→ -39.25
  - 실제 계산 값도 -39.25로 구현 정확성 확보

```
X | v_arr[0][1]
X | v_arr[0][2]
X | v_arr[0][4]

X | v_arr[1][2]

X | v_arr[2][2]

X | v_arr[3][2]
X | v_arr[3][3]

X | v_arr[4][2]

X | c_arr[0][3]
X | c_arr[0][4]

X | c_arr[1][4]

X | c_arr[2][4]

X | c_arr[3][2]

X | c_arr[4][5]

X | sqr_rR[9]
```

자원 추정

- **기존 R10D10 ~ R70D70과 더불어 정밀도에 따른 자원도 추정하기 위해 소수부분 2자리, 4자리 추가하여 실험 중**
  - 현재 R70D70의 소수 2자리, 4자리만 남은 상태 (실험 2개)

- **기존 구현에서 디멘션이 소수 자리만큼 늘어나는 것과 유사하므로 이에 해당하는 추가 자원이 사용될 것으로 보임**

- **고정소수점 연산이 헷갈려서 다시 보긴 할 예정**

- **실험 완료 후 개선 논문으로 우선 작성할 계획**

감사합니다