ME 201/APh 250, Homework 4:

Assigned: Friday, May 1, 2019
Due: Friday, May 15, 2019

1:
(a) From equation (2.10 to 2.12) from Lecture 7, we know that $a_p^\dagger a_p = \frac{1}{2}(I_o - Z_p)$. We now consider $a_i^\dagger a_j^\dagger a_j a_i$ where $j > i$.

$$\begin{aligned} a_i^\dagger a_j^\dagger a_j a_i &= a_i^\dagger (a_j^\dagger a_j) a_i \\ &= a_i^\dagger (\frac{1}{2}(I_j - Z_j)) a_i \\ &= \frac{1}{2}(a_i^\dagger a_i) + a_i^\dagger a_i Z_j \\ &= \frac{1}{4}(I_i - Z_i) - \frac{1}{4}(I_i - Z_i) Z_j \\ &= \frac{1}{4}(I - Z_i - Z_j + Z_i Z_j) \end{aligned}$$

Then using the coefficients from the table, we are done. Let us use PyQuil to verify this is indeed true. It is possible to implement Jordan-Wigner transform in PyQuil. I pasted the code snippet here, and we can see this is indeed the case.

```
from grove.alpha.fermion_transforms import jwtransform
jw = jwtransform.JWTransform()
h = h00*jw.create(0)*jw.kill(0) + h11*jw.create(1)*jw.kill(1) + h22*jw.create(2)*jw.kill(2) \
   + h33*jw.create(3)*jw.kill(3) \
   + h0110*jw.create(0)*jw.create(1)*jw.kill(1)*jw.kill(0)\
   + h2332*jw.create(2)*jw.create(3)*jw.kill(3)*jw.kill(2)\
   + h0330*jw.create(0)*jw.create(3)*jw.kill(3)*jw.kill(0)\
   + h1221*jw.create(1)*jw.create(2)*jw.kill(2)*jw.kill(1)\
   + (h0220-h0202)*jw.create(0)*jw.create(2)*jw.kill(2)*jw.kill(0)\
   + (h1331-h1313)*jw.create(1)*jw.create(3)*jw.kill(3)*jw.kill(1)\

print(h.__str__())
```

```
(-0.8126100000000005+0j)*I + (0.17120100000000002+0j)*Z0 + (0.17120100000000002+0j)*Z1 + (-0.
22279649999999998+0j)*Z2 + (-0.22279649999999998+0j)*Z3 + (0.16862325+0j)*Z0*Z1 + (0.17434925
+0j)*Z2*Z3 + (0.165868+0j)*Z0*Z3 + (0.165868+0j)*Z1*Z2 + (0.12054625+0j)*Z0*Z2 + (0.12054625+
0j)*Z1*Z3
```

Figure 1: Code snippet to evaluate the jordan wigner string on PyQuil. Note one has to install grove to import the necessary functions.

(b) We use PyQuil to evaluate this.

```
from grove.alpha.fermion_transforms import jwtransform
jw = jwtransform.JWTransform()

h = h0312*jw.create(0)*jw.create(3)*jw.kill(1)*jw.kill(2)\
  + h0312*jw.create(2)*jw.create(1)*jw.kill(3)*jw.kill(0)\
  + h0132*jw.create(0)*jw.create(1)*jw.kill(3)*jw.kill(2)\
  + h0132*jw.create(2)*jw.create(3)*jw.kill(1)*jw.kill(0)

print(h.__str__())
```

```
(-0.04532175+0j)*Y0*Y1*X2*X3 + (0.04532175+0j)*Y0*X1*X2*Y3 + (0.04532175+0j)*X0*Y1*Y2*X3 + (-0.04532175+0j)*X0*X1*Y2*
Y3
```

Figure 2: Code snippet to evaluate the jordan wigner string on PyQuil. Note one has to install grove to import the necessary functions.

(c) We just need to implement three types of gates. They are σ_p^z, $\sigma_i^z\sigma_j^z$ and a mixture of σ^x and σ^y. We will use (i) σ_0^z , (ii) $\sigma_1^z\sigma_0^z$ and (iii) $\sigma_3^x\sigma_2^x\sigma_1^y\sigma_0^y$ as examples. The rest will have similar circuit implementations.

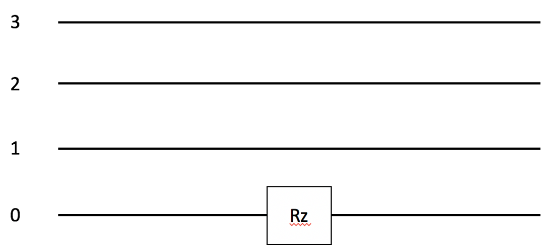

Figure 3: Circuit implementation for σ_0^z.

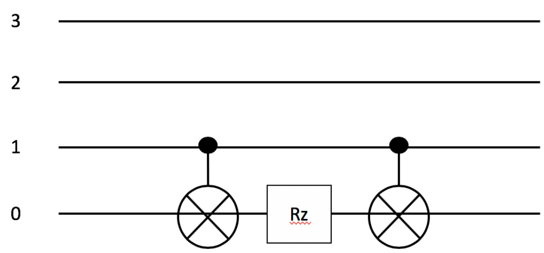

Figure 4: Circuit implementation for $\sigma_1^z\sigma_0^z$.

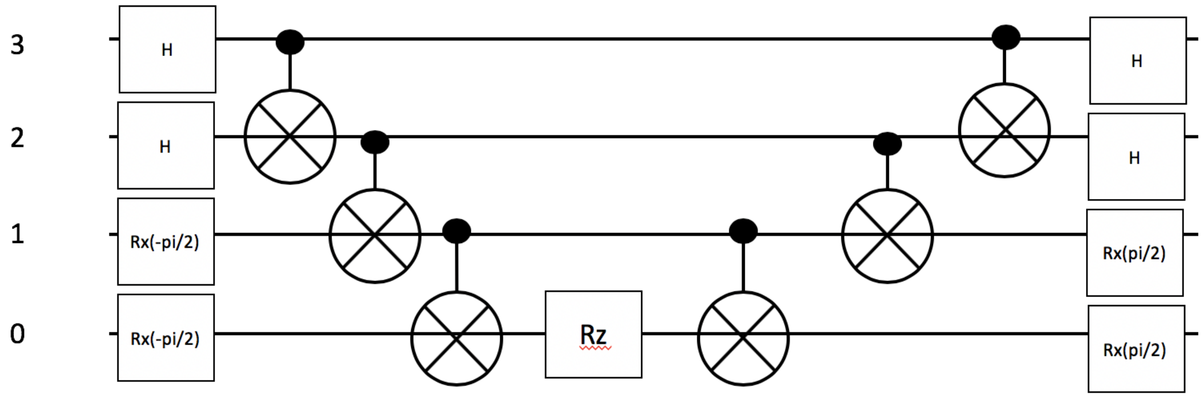

Figure 5: Circuit implementation for $\sigma_3^x \sigma_2^x \sigma_1^y \sigma_0^y$.

2
Check out the notebook on the website.