HW 5
ECE 592 / CSC 591
Alexander Allen, Keith Mellendorf

Our goal for this week was to see some amount of code written in Qiskit be viewed in a DAX format.

To begin we forked the repository created by AQT for their Ion Trap Device. We removed a lot of things from it e.g. access tokens and connections to a server, mostly keeping some infrastructure to allow us to act as a provider / backend quickly. Keith worked on the side of system integration and removal of obsolete systems. Alex worked on the parsing of the qobj into individual qasm type instructions, which then have been turned into a list of DAX format instructions.

Repo changed:
Because we worked on code that was forked from a non-ncsu github repository, we had to manually do so into a new empty repo. Alex is the one that did that, so here is the link to our new project repository.
https://github.ncsu.edu/arallen4/qc-ion-trap

All told, we now have the ability to view the DAX version of a qiskit program. The only changes to make to the qiskit script are setting the provider and backend. The dax code is read from the job.result() method .print_dax(), as shown below.

Next steps:

- Instead of just printing the code to the terminal, printing it as a file.
- There were other requirements indicated in the conversation we had with Duke a few weeks ago, such as the interaction with the ‘with parallel:’ keyword. Before we implement that we need to have another meeting with Dr. Mueller, and or Leon Riesebos.
- There is some infrastructural work to do as well, such as renaming code called aqt to something that makes sense.

Sample code and results shown on page 2.

A test qiskit script, followed by the terminal output it generated.

```
from qiskit import *
from qiskit.providers.aqt import AQT

aqt = AQT.get_provider() # aqt is a provider

backend = aqt.get_backend('aqt_qasm_simulator') # backend is a job

qc = QuantumCircuit(2, 2)
qc.h(0)
qc.cx(0, 1)
qc.measure([0,1], [0,1])
result = execute(qc, backend).print_dax()
```

```
(venv) arallen@Alexander-Ubuntu:~/NCSU/QC/Project/qiskit-aqt-provider$ python
test_script.py
--- DAX Code ---from dax.experiment import *
class ConstructedExperiment(EnvExperiment):
        def build(self):
                self.setattr_device('core')
        @kernel
        def run(self):
                self.load_ions(2)
                self.initialize_all()
                self.h(0)
                self.cnot(0, 1)
                self.detect_all()
                r = self.measure_all()
```