RUN

Run simple circuits using IBM Quantum (5 points)

- Prepare circuits X, SX, $SX \rightarrow RZ(\pi/4) \rightarrow SX$, with measurement
- Run circuit SX, SX, with two measurements
- Run circuit SX, measurement, again SX, next measurement
- Run conditional circuit: SX then measurement, if 0 again SX, if 1 do X, finally SX again and measurement
- Draw the pulse sequences
- Run on the AER simulator
- Plot the results – check with theoretical predictions

Have fun!

Presentation: run with IQ kernel discriminator on REAL quantum machine.

Helpful commands 1

- `service = QiskitRuntimeService(channel="ibm_cloud", instance=CRN, token=TOKEN)`
- `backend = AerSimulator()` or REAL `backend = service.backend("ibm_fez")`
- `qc=QuantumCircuit(Q,C)` circuit with Q qubits and C cbits
- `qc.sx(q)` gate SX on qubit q
- `qc.rz(a,q)` gate RZ_a on qubit q
- `qc.measure(q,c)` measure q to c
- `sampler = Sampler(backend)` prepare to run
- `job=sampler.run([qc1,qc2], shots=N)` run!
- `result=job.result()` read out

Helpful commands 2

- `result[i].data.c.get_counts()` get counts
- `qc.draw("mpl")`
- `from qiskit.visualization.timeline import draw`
- `draw(qc3,target=backend.target,idle_wires=False)`
- `qc=QuantumCircuit(1,2)` circuit with Q qubits and C cbits
- `qc.rz(a,q)` gate RZ_a on qubit q
- `qc.measure(q,c)` measure q to c
- `with qc.if_test((c,v)) as other:`
  `.... #if c=v`
  `with other:`
  `.... #otherwise`
- order of outcomes "BA": A – result of cbit 0, B – result of cbit 1 (right to left).