CS 1190: Fundamentals of Quantum Information Science

From Qubits to Quantum Circuits

Guided IBM Quantum Lecture and Live Experiments — November 2, 2026

Mohammad Saidur Rahman, Ph.D.

Department of Computer Science, University of Texas at El Paso

Course website: qis.rahmanmsaidur.com

Today's question: Can we use simple quantum gates to produce, predict, and measure superposition, interference, and entanglement? We will build circuits one at a time, run them using Qiskit, and compare predictions with the measured outcomes. This lecture connects our earlier photon, double-slit, and entanglement lessons to IBM Quantum.

How to follow the class. Keep this handout beside the instructor's notebook `CS1190_FINAL_01_INSTRUCTOR_LIVE_DEMO.ipynb`. Both use exactly the same **Step 0–8** labels. Each green box tells you what to **Predict, Run, Observe, and Explain**. The separate student homework notebook uses the same concepts but different problems. You do not need to write Python code from memory.

By the end, you should be able to: read a circuit; use X, H, Z, and CNOT; interpret a measurement histogram; explain interference and Bell-pair correlations; and describe why the measurement basis matters for BB84.

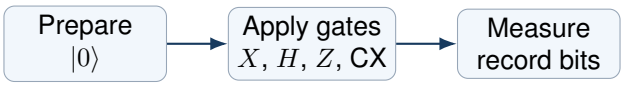
Step	What we learn	Notebook cells
0–1	Setup, qubits, measurements, and bit flips	0A–0B; 1A–1B
2–3	Superposition and interference	2; 3A–3B
4	Measure in Z or X basis	4A–4B; *4C optional*
5	CNOT and entanglement	5A–5B; *5C optional*
6	BB84 and sifting	6A; *6B optional*
7	Real IBM hardware (instructor only)	7A–7D; *optional*
8	GHZ and the no-cloning principle	8A–8B; *optional*

Where are we running this? The *Jupyter notebook* runs in Google Colab or JupyterLab. Steps 0–6 use an *ideal local simulation* and need no IBM login. Only Step 7 connects to physical IBM quantum hardware. *IBM Quantum Composer* is a separate, visual drag-and-drop circuit editor; it is not the Jupyter notebook. Students do not need your IBM API key or CRN.

Step 0. What happens when a quantum circuit runs?

A **bit** is either 0 or 1. A **qubit** is a quantum system with two reference states, written $|0\rangle$ and $|1\rangle$ (pronounce these "ket zero" and "ket one"). For example, our earlier lessons used two photon polarizations as qubit states. IBM's quantum processors instead use *superconducting circuits*. The devices differ physically, but the same circuit language helps describe both.

A **quantum circuit** tells us which state to prepare, which **gates** to apply, and when to **measure**. A gate is a controlled change in the state. A measurement turns that quantum state into a classical answer such as 0 or 1. A circuit diagram is read *from left to right*.



A **shot** is one full run of the procedure: prepare a fresh state, apply gates, and measure. A 1,024-shot experiment repeats that procedure 1,024 times. Its **counts** describe how often each classical result appeared. A **histogram** is a bar chart of those counts.

Notebook: Step 0: setup cells 0A and 0B

Predict. Does installing Qiskit use a quantum processor?
Run. Open Google Colab, upload the instructor notebook, run 0A once, and run 0B to define the plotting function.
Observe. The notebook prints its Qiskit version and reports that the first experiments run locally.
Explain. The local simulator lets us test ideal circuit predictions without queuing a real hardware job.

Step 1. Known states: first measure, then flip with X

A fresh qubit starts in $|0\rangle$ in these demonstrations. With no gate, an ideal measurement gives 0 every time. The X gate exchanges the reference states, much like the NOT operation on a known classical bit:

$$X|0\rangle = |1\rangle, \qquad X|1\rangle = |0\rangle, \qquad X(X|0\rangle) = |0\rangle.$$



Read aloud: “Start in ket zero, apply X, and measure. The expected answer is one.” Two X gates undo one another because the bit flips twice.

Notebook: Demo 1A (initial state), then Demo 1B (X gate)

Predict. What will we observe for $|0\rangle$, $X|0\rangle$, and $XX|0\rangle$?
Run. Execute 1A and 1B in order. The two-X extension is in Homework 2.
Observe. 1A shows only 0; 1B shows only 1 on an ideal simulator.
Explain. An X gate changes a known basis state; it is not the same operation as measurement.

Home practice: Homework 1–2.

Step 2. Superposition: what does the Hadamard gate do?

The Hadamard gate, abbreviated H, changes a known reference state into a **superposition**. Starting from $|0\rangle$, it produces the **plus state**:

$$\boxed{H|0\rangle = |+\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}.}$$

The fractions in this expression are **probability amplitudes**, not measured probabilities. To predict the result of a 0/1 measurement, we square the amplitude’s magnitude:

$$P(0) = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}, \qquad P(1) = \left|\frac{1}{\sqrt{2}}\right|^2 = \frac{1}{2}.$$



What “50/50” means. One shot gives one ordinary bit, never “half of each bit.” If we prepare and measure the same state 1,024 times, we expect about 512 zeros and 512 ones. We should not demand exactly 512 each.

Notebook: Demo 2: $H|0\rangle$

Predict. Does the histogram show only 0, only 1, or both?
Run. Execute Demo 2, which applies H and then measures 1,024 shots.
Observe. Both 0 and 1 occur; the fractions should be near 50% each.
Explain. The squared amplitudes predict the probabilities; the finite histogram shows measured samples.

Common confusion: "A superposition is just a hidden random classical bit." A classical 50/50 mixture gives the same immediate 0/1 histogram, but the next experiment shows that *relative phase* can change what a later gate reveals.

Home practice: Homework 3 and 9.

Step 3. Interference: why do HH and HZH differ?

This is the key experiment in the lecture. Introduce the **minus state**:

$$|-\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$

Both $|+\rangle$ and $|-\rangle$ give 50/50 outcomes if measured immediately in the usual 0/1 basis. The *minus sign* is not a negative probability: it represents a different **relative phase**, a relationship between the amplitudes. The Z gate changes that relationship:

$$Z|0\rangle = |0\rangle, \qquad Z|1\rangle = -|1\rangle, \qquad Z|+\rangle = |-\rangle.$$

When we subsequently apply H, the two states behave differently:

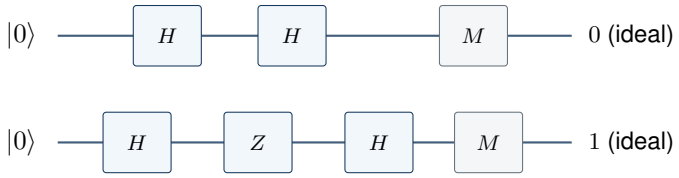
$$\boxed{H|+\rangle = |0\rangle}, \qquad \boxed{H|-\rangle = |1\rangle}.$$

Worked example. Use $H|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and $H|1\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$:

$$\begin{aligned} H|+\rangle &= \frac{H|0\rangle + H|1\rangle}{\sqrt{2}} \\ &= \frac{(|0\rangle + |1\rangle) + (|0\rangle - |1\rangle)}{2} \\ &= |0\rangle. \end{aligned}$$

The two $|1\rangle$ contributions cancel and the two $|0\rangle$ contributions reinforce. This is **interference**: quantum amplitudes combine before we calculate probabilities. We encountered the same principle in the double-slit experiment.

Compare the two circuits.



Notebook: Demo 3A (HH) and Demo 3B (HZH)

Predict. Does adding a second H necessarily produce another 50/50 distribution? What if we insert Z in between?
Run. Execute 3A and then 3B. Compare the resulting histograms.
Observe. 3A measures 0 in every ideal shot; 3B measures 1 in every ideal shot.
Explain. The middle Z changes $|+\rangle$ to $|-\rangle$, and the last H turns this phase difference into a visible bit.

Home practice: Homework 4 and 5.

Step 4. Measurement basis: how do we choose what to ask?

A measurement **basis** specifies which two states the measurement can distinguish. Our ordinary computational-basis measurement is called the Z **basis**: it distinguishes $|0\rangle$ and $|1\rangle$. Another common choice, called the X **basis**, distinguishes $|+\rangle$ and $|-\rangle$.

To measure in the X basis with the usual computational-basis readout, apply H *immediately before* the measurement. The bit 0 then represents the $|+\rangle$ outcome and the bit 1 represents the $|-\rangle$ outcome.

State prepared	Direct Z measurement	X measurement (H then readout)
$\|0\rangle$	always 0	50% 0, 50% 1
$\|1\rangle$	always 1	50% 0, 50% 1
$\|+\rangle$	50% 0, 50% 1	always 0
$\|-\rangle$	50% 0, 50% 1	always 1

Notebook: Demos 4A–4B; optional 4C1–4C2

Predict. Will $|+\rangle$ measured in Z and X have the same histogram? What about $|-\rangle$?
Run. Run 4A (prepare $|+\rangle$, measure Z), then 4B (prepare $|+\rangle$, measure X). For more practice, run 4C1 and 4C2 with $|-\rangle$ prepared by X then H.
Observe. $|+\rangle$: 50/50 in Z, always 0 in X. $|-\rangle$: 50/50 in Z, always 1 in X.
Explain. The two states have the same Z probabilities but different relative phases, revealed by the X measurement.

Z gate $\neq$ Z measurement. The Z *gate* changes the state phase; a Z-basis *measurement* produces a classical 0 or 1. The same letter refers to two different operations.

Home practice: Homework 5 and 8.

Step 5. CNOT and entanglement: what changes with two qubits?

With two qubits, the possible computational-basis results are `00`, `01`, `10`, and `11`. If we apply H independently to both qubits, we get four equally likely outcomes:

$$|+\rangle \otimes |+\rangle = \frac{|00\rangle + |01\rangle + |10\rangle + |11\rangle}{2}.$$

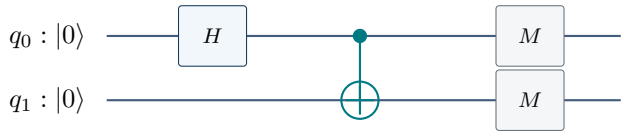
The symbol $\otimes$ means that we are describing two different qubits together. The two states are *independent*: no gate has linked their outcomes.

The **CNOT** (controlled-X, written CX) has a *control* and a *target*. It flips the target only when the control is 1. In our notebook `qc.cx(0, 1)` uses q_0 as control and q_1 as target.

Bit order in Qiskit: When using `measure_all()`, the reported two-bit string is displayed as `q1q0`, *not* `q0q1`. For example, `01` means $q_1 = 0$, $q_0 = 1$; if we apply CNOT$(0, 1)$, the output becomes `11`.

Input (q1q0)	Does q0 control a flip?	Output (q1q0)
00	No	00
01	Yes	11
10	No	10
11	Yes	01

Prepare a Bell state. Start from $|00\rangle$, apply H to q_0, and then CNOT with control q_0 and target q_1.



$$|00\rangle \xrightarrow{H \text{ on } q_0} \frac{|00\rangle + |01\rangle}{\sqrt{2}} \xrightarrow{\text{CX}(0,1)} \boxed{|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}}.$$

This is an **entangled state**: its quantum description cannot be separated into one state for q_0 and an independent state for q_1. In an ideal Z-basis measurement, we get only `00` and `11`, each 50% of the time.

Circuit	00	01	10	11
H on both qubits	25%	25%	25%	25%
H on q_0, then CX$(0, 1)$	50%	0%	0%	50%

Notebook: Demos 5A–5B; optional 5C (CNOT truth table)

Predict. Which outcomes will disappear when we replace two independent superpositions with a Bell pair? When does the CNOT target flip?
Run. Run 5A, then 5B. For the full truth table, run 5C four times while changing `INPUT_NUMBER` from 0 through 3.
Observe. 5A has four equally likely outputs; 5B has only 00/11 ideally. Demo 5C reproduces the four CNOT rows above.
Explain. The H–CNOT sequence prepares a Bell state. Its correlations differ from independent qubits.

Does a histogram alone prove entanglement? No. Seeing only 00 and 11 in the Z basis is also compatible with a *classical mixture* of 00 and 11. We know the *ideal circuit we constructed* creates an entangled state, but experimentally certifying entanglement requires additional information, such as measurements in other bases.

Home practice: Homework 6–7.

Step 6. BB84: why measurement bases matter for secure communication

Alice wishes to share a secret key with Bob using quantum states. The **BB84 quantum key distribution protocol**, introduced by Charles H. Bennett and Gilles Brassard in 1984, is built around a simple choice: for each signal, Alice chooses a random bit and one of two preparation bases; Bob chooses a basis independently.

Alice's basis	Bit 0 state	Bit 1 state
Z	$\|0\rangle$	$\|1\rangle$
X	$\|+\rangle$	$\|-\rangle$

When their bases *match*, Bob gets the intended bit in an ideal, noiseless experiment. When they *differ*, the result is random (50/50). After all measurements, Alice and Bob publicly announce their *basis choices*, not the secret bit values. They discard mismatched positions. This is called **sifting**.

Notebook: Demo 6A (basis choices), optional Demo 6B (sifting)

Predict. Which combinations of Alice's and Bob's bases should produce predictable outcomes? Which positions survive a sifting table?
Run. In 6A, set `CASE_NUMBER` to 0, 1, 2, and 3 one at a time. In 6B, run the separate six-position sifting example.
Observe. Matching bases reproduce Alice's bit; mismatched bases give about 50/50. In Demo 6B, positions 1, 3, 5, and 6 are kept.
Explain. The bases determine whether Bob reads the encoded bit reliably. Sifting removes measurements taken in incompatible bases.

Worked sifting example (the homework has different data):

Position	1	2	3	4	5	6
Alice bit	0	1	1	0	1	0
Alice basis	Z	X	X	Z	Z	X
Bob basis	Z	Z	X	X	Z	X
Keep?	Yes	No	Yes	No	Yes	Yes

Sifted Alice bits: 0110. These are *candidate sifted bits*, not a complete secure key.

Security qualification. Real BB84 needs an authenticated classical channel, error estimation, reconciliation, privacy amplification, and protection against imperfect devices. Merely discarding mismatched bases does not guarantee secrecy.

Home practice: Homework 8 and 10 (a new, eight-position example).

Step 7. Optional: test a Bell state on a real IBM quantum computer

All previous experiments used `StatevectorSampler`, which samples from *ideal* quantum predictions on your Colab/Jupyter computer. A physical IBM quantum processing unit (QPU) has finite coherence, imperfect gates, and imperfect measurements. These can cause extra errors that do not appear in the ideal simulator.

Cell	What you do	What happens
7A	Enter your API key and CRN in the *hidden prompts*	Connect to your own IBM Quantum instance
7B	Display accessible operational QPUs	No job submitted
7C	Deliberately enable `SUBMIT_REAL_JOB=True`	Submits one 512-shot Bell-state job
7D	Inspect the result *after the job finishes*	Plot the actual measured counts

The **transpiler** converts a circuit into gates and connections supported by the selected hardware. This preserves the intended ideal computation, although the real implementation is not perfect.

Notebook: Instructor-only hardware Steps 7A–7D

Predict. Will a real device necessarily return exactly 256 counts of 00 and 256 of 11 in 512 shots?
Run. Use 7A and 7B to verify access. If desired, submit one small job using 7C. In 7D, check its status and plot the counts when complete.
Observe. 00 and 11 should dominate if the Bell preparation works well. Errors may introduce 01 and 10. Counts need not be exactly equal.
Explain. Unequal counts can result from finite sampling; unexpected bitstrings can also result from physical errors.

Instructor security and billing. Never type an API key into a visible code cell, never share your executed private notebook, and never give students your credentials. QPU jobs can queue and may consume allocation or incur charges depending on the chosen IBM instance. **The class and homework work without any hardware job.**

Optional visual alternative: IBM Quantum Composer can show circuits and simulated probabilities in a browser. A displayed ideal probability chart is not the same thing as completed real-hardware measurement counts.

Step 8. Extensions: three-qubit GHZ states and no-cloning

Bonus 8A: three qubits. Start with $|000\rangle$, apply H to q_0, then CX$(0,1)$ and CX$(1,2)$. This creates a **GHZ state**:

$$|\mathrm{GHZ}\rangle = \frac{|000\rangle + |111\rangle}{\sqrt{2}}.$$

In the ideal computational basis, only 000 and 111 appear. This is different from independently applying H to three qubits, which makes eight outcomes equally probable.

Bonus 8B: why can't CNOT copy an unknown qubit? For a known computational-basis input, CNOT with a $|0\rangle$ target resembles copying: $|0\rangle|0\rangle \mapsto |00\rangle$ and $|1\rangle|0\rangle \mapsto |11\rangle$. But for $|+\rangle$,

$$|+\rangle|0\rangle \xrightarrow{\mathrm{CX}} \frac{|00\rangle + |11\rangle}{\sqrt{2}} \neq |+\rangle|+\rangle.$$

The output is entangled, *not* two independent copies. If both qubits really were $|+\rangle$, measuring each in the X basis would give 00 every time. The Bell state instead gives 00 or 11. This illustrates the **no-cloning theorem**: no universal physical operation perfectly copies every arbitrary unknown quantum state.

Notebook: Optional Demos 8A and 8B

Predict. How many outcomes should a three-qubit GHZ state have? Could CNOT copy $|+\rangle$ into another independent $|+\rangle$?
Run. Execute 8A, then 8B when additional time is available.
Observe. GHZ: 000 and 111. No-cloning example: 00 and 11 after measuring both in the X basis.
Explain. Multi-qubit correlations are not the same as preparing independent copies.

Home practice: Bonus A and Bonus B.

How to read the Qiskit code you see on screen

The meaning of each line matters more than memorizing its Python syntax. This is the local, ideal Bell-state demonstration; it needs *no IBM Cloud credentials*:

```
from qiskit import QuantumCircuit
from qiskit.primitives import StatevectorSampler

qc = QuantumCircuit(2)  # Two qubits begin at |00>
qc.h(0)                 # H on q0
qc.cx(0, 1)             # CNOT: control q0, target q1
qc.measure_all()        # Read both qubits in Z basis

sampler = StatevectorSampler(seed=2026)
result = sampler.run([qc], shots=1024).result()
counts = result[0].data.meas.get_counts()
print(counts)           # Near-half 00 and near-half 11
```

Some useful translations: `qc.x(0)` means apply X to qubit 0; `qc.h(0)` means H; `qc.z(0)` means Z; `qc.cx(0,1)` means CNOT; `qc.measure_all()` means standard-basis measurement. The code does not return the entire quantum state, but measured classical counts.

How many shots are enough to see a 50/50 result?

If outcome 1 appears n_1 times out of N shots, the measured proportion is $\widehat{P}(1) = n_1/N$. For example, 509 ones in 1,024 shots gives

$$\widehat{P}(1) = \frac{509}{1024} \approx 0.497 \quad \text{(49.7\%)}.$$

This is consistent with a theoretical probability of 50%. The typical statistical spread in the proportion for independent shots of a two-outcome experiment is

$$\sigma_{\widehat{p}} = \sqrt{\frac{p(1-p)}{N}}.$$

For $p = 1/2$, this is $1/(2\sqrt{N})$. With 16, 128, and 2,048 shots, the typical spreads are about 12.5, 4.4, and 1.1 percentage points, respectively. More shots generally stabilize the *proportion*; they do not guarantee a closer result in every individual experiment.

Short exit ticket: check your understanding

Q1. Why do $H|0\rangle$ and $HH|0\rangle$ produce different measurement distributions?

Q2. Why does inserting Z between two Hadamard gates change the expected result?

Q3. How can $|+\rangle$ and $|-\rangle$ have identical Z-basis histograms but different X-basis results?

Q4. For `qc.cx(0,1)`, what does the output bitstring `01` mean about q_1 and q_0?

Q5. What distinguishes two independently prepared $|+\rangle$ qubits from a Bell pair? Does one histogram prove entanglement?

Q6. Why do Alice and Bob discard mismatched-basis positions in BB84?

Q7. What is the difference between an ideal simulator, a visual Composer probability chart, and a real QPU result?

Homework: where to practice each idea

The student notebook `CS1190_FINAL_02_STUDENT_HOMEWORK.ipynb` contains **10 tasks totaling 100 points** and two optional bonus tasks. It uses simulation only; no IBM key, CRN, or real hardware job is required.

Lecture/Notebook	Homework
Step 1: known states, X	HW 1, 2
Step 2: Hadamard, sampling	HW 3, 9
Step 3: relative phase, interference	HW 4, 5
Step 4: measurement bases; optional 4C	HW 5, 8
Step 5: CNOT, Bell pair; optional 5C	HW 6, 7
Step 6: BB84; optional 6B (different example)	HW 8, 10
Step 8: GHZ, no cloning	Bonus A, B

Five ideas worth remembering

1. X swaps $|0\rangle$ and $|1\rangle$; H can prepare a superposition.
2. Probabilities are squared amplitudes; a single measurement always produces one classical result.
3. Relative phase affects later interference, even when the immediate histograms look identical.
4. CNOT can link qubits; H followed by CNOT prepares an entangled Bell state.
5. A measurement basis determines which quantum information is accessible; BB84 relies on matching and comparing bases.

One-page study reference

Use this summary to check your predictions before starting the homework. Think about the *state just before measurement*, not only the first gate in a circuit.

Circuit from $\|0\rangle$	State just before measurement	Ideal measured bits
No gates	$\|0\rangle$	only 0
X	$\|1\rangle$	only 1
H	$\|+\rangle = (\|0\rangle + \|1\rangle)/\sqrt{2}$	0 and 1, 50/50
HH	$\|0\rangle$	only 0
HZH	$\|1\rangle$	only 1

Two-qubit circuit	What was prepared?	Ideal outcomes
H on both qubits	Two independent $\|+\rangle$ states	00, 01, 10, 11 each 25%
H on q_0, then CX$(0, 1)$	$\|\Phi^+\rangle = (\|00\rangle + \|11\rangle)/\sqrt{2}$	00 and 11 each 50%

Three habits for reading any new circuit.

1. Trace each gate *from left to right*; do not measure early unless the circuit says so.
2. Name the *basis* of the final measurement. A final H can change what a standard readout reveals.
3. Write down the *ideal probabilities* before checking the sampled counts. Counts can vary from run to run.

Sources and further reading

1. IBM Quantum, *Quickstart* (Qiskit, StatevectorSampler, Bell experiment). https://quantum.cloud.ibm.com/docs/en/guides/quick-start.
2. IBM Quantum, *Initialize your IBM Quantum Compute Service account* (API key and instance CRN). https://quantum.cloud.ibm.com/docs/en/guides/initialize-account.
3. IBM Quantum, *Sampler quickstart* and *Migrate from server-side to client-side Sampler and Estimator*. https://quantum.cloud.ibm.com/docs/en/guides/get-started-with-sampler; https://quantum.cloud.ibm.com/docs/en/guides/migrate-to-client-side-primitives.
4. IBM Quantum Learning, *Quantum Key Distribution*. https://quantum.cloud.ibm.com/learning/en/modules/computer-science/quantum-key-distribution.
5. C. H. Bennett and G. Brassard, "Quantum Cryptography: Public Key Distribution and Coin Tossing," *Proc. IEEE International Conference on Computers, Systems and Signal Processing*, Bangalore, pp. 175–179, 1984.
6. W. K. Wootters and W. H. Zurek, "A Single Quantum Cannot Be Cloned," *Nature* 299, 802–803 (1982). https://doi.org/10.1038/299802a0.
7. D. Dieks, "Communication by EPR Devices," *Physics Letters A* 92, 271–272 (1982). https://doi.org/10.1016/0375-9601(82)90084-6.
8. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th Anniversary Edition, Cambridge University Press, 2010.