Introduction To Quantum Computing

PHYS/CSCI 3090

Prof. Alexandra Kolla

Alexandra.Kolla@Colorado.edu
ECES 122

Prof. Graeme Smith

Graeme.Smith@Colorado.edu
JILA S326

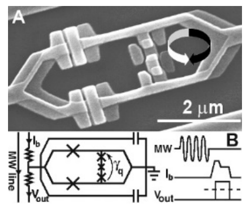
https://home.cs.colorado.edu/~alko5368/indexCSCI3090.html

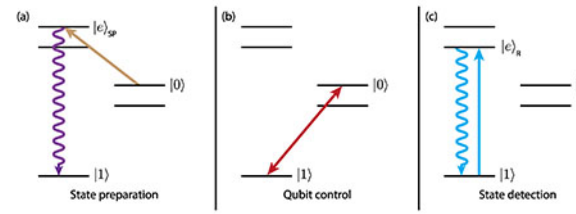
Why are we here?

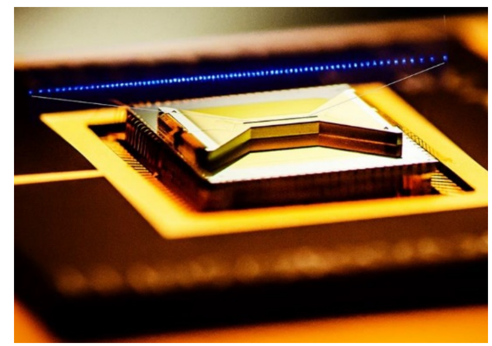
- The world obeys Quantum Theory (old news!)
- Computers that fully harness quantum effects could outperform classical ones.
- Building quantum computers is very hard, but not ridiculously, impossibly hard.
- We are at a special moment: beginning to build nontrivial quantum computers

- **This class:** you will learn what a quantum computer is, why we think it'd be useful (quantum algorithms), and why we think it can be built (quantum error correction).

There are quantum computers







Superconducting qubits
Yale, Google, IBM, Rigetti

Ion traps
ionQ/UMD, NIST Boulder, Honeywell

Significant industrial effort in both hardware and software:
Amazon, Google, IBM, Microsoft, Rigetti, PsiQuantum, ionQ, Intel, Lockheed-Martin, ColdQuanta, Zapata, QC Ware, Xanadu…

Our goal: What are these things (going to be) good for?

Today

- Logistics
  - Who are we?
  - Who are you?
  - Clickers
  - Grading
  - Outline
- Assignment 0
  - Due next Monday
  - Covers linear algebra

Prof. Kolla

Affiliation: Computer Science
Office: ECES 122
Previously:
4 yrs at UIUC
Postdoc at Microsoft,
IAS Princeton

Research: Theoretical computer science, spectral graph theory, statistical physic, quantum computing.

Teaching: Complexity, Algorithms, Discrete math

Prof. Smith

Affiliation: Physics and JILA
Office: JILA S326
Previously:
9 yrs at IBM Research
Postdoc in CS, PhD Physics

Research: Theoretical quantum information, and quantum computing. Esp. Error correction.

Teaching: Anything with Quantum Mechanics!

3090 Team

- Prof. Kolla and Prof. Smith

Alternate lectures by chapter covered.

Joint office hours Friday 3-4 JILA X317

Graders

Steven Kordonowy

Ariel Shlosberg (Phys Helproom 2-4pm Tues + Thurs)

Matteo Wilczak

Who Are You?

- Quantum Computing Enthusiasts!
- Majoring in: Physics, Computer Science, Engineering Physics, Computer Engineering, Math, Astrophysics, Applied Math, Aero, EE, ….
- Took one of APPM 2360, APPM 3310, CSCI 2820, MATH 2130, MATH 2135, or something else covering linear algebra.

Clickers!

Audience participation system.
Excellent way to get feedback.
Helps us not to lose you!

Standard clicker at CU.

Some of you may have or have seen these.

Fire 'em up!

(push the On/Off button)

Should get Green light

If not, hold power until flashing green,
Then push DC. Should go green solid.

Clicker warm-up!

TRUE (A) or FALSE (B):

My clicker is set to DC, is on, and is working.

Clicker experience.

I have used clickers before:

A) YES!
B) No
C) |YES>+|NO>
D) $E=mc^2$
E) Look, it's still not green!

Your iClicker

Put your name and contact information on your clicker!

If you lose it, there is a chance it will be returned.

Pro tip: *You can put your contact information on a piece of tape on the clicker if you plan to return the iClicker in the future*

Your iClicker

Use only *your own* iClicker!

Responding with another student's iClicker is a violation of the Honor Code and you are encouraged not to do it.

Clicker Points!

- Participation points for each question answered.

- **Clicker points count for up to 2% bonus points.**

Grading Scheme

- 30% Weekly Problem sets
- 20% Midterm 1
- 20% Midterm 2
- 30% Final
- +2% bonus from clicking

Exams

- Midterm 1: February 12 (in class)
- Midterm 2: March 18 (in class)
- Final: TBD

There are no rescheduled exams, so please put this in your calendar now.

Extra time midterm exams begin at same time in another location.

Typical Weekly Schedule

Monday	Tuesday	Wednesday	Thursday	Friday
12 noon– HW due 2-2:50pm – Lecture 4pm – old HW sol'n posted; new HW posted	2-4 Ariel S. in Physics Helproom	2-2:50pm – Lecture	2-4 Ariel S. in Physics Helproom	2-2:50pm – Lecture 3-4 Office Hours JILA X317

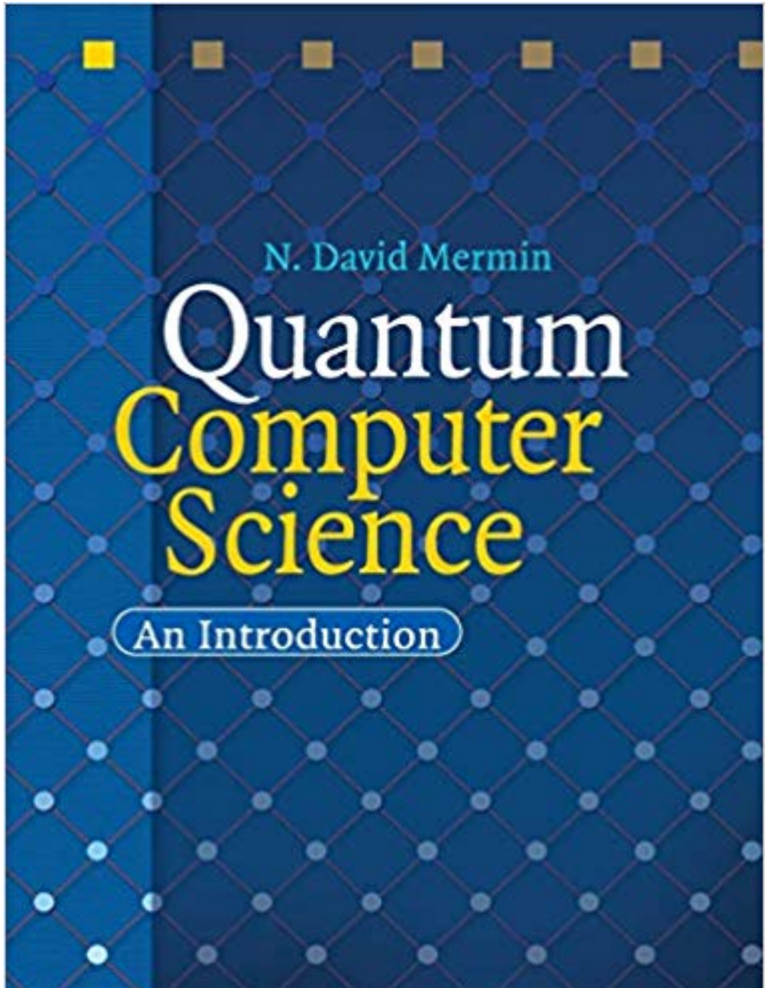
Class website

- https://home.cs.colorado.edu/~alko5368/indexCSCI3090.html
- Has logistics, assignments, additional reading, etc
- Also, keep up to date on Canvas

Weekly Assignments

- Submitted via Canvas
- Scanned pdfs

Textbook



Expected in bookstore on 01/15

What we'll cover

- Chapter 1: Classical and Quantum Bits and Circuits
- Chapter 2: Simple Algorithms (Deutsch, B-V, Simon)
- Chapter 6: Few-qubit Protocols (teleportation, dense coding, quantum cryptography)
- Chapter 4: Quantum Search (Grover's Algorithm)
- Chapter 3: Quantum Factoring (Shor's Algorithm)
- Chapter 5: Quantum Error Correction

This week

- Wednesday: What's a classical bit (c-bit), and what's a quantum bit (qubit)?
  Reading: Mermin 1.1.-1.6

- Friday: Manipulating Quantum systems, quantum circuits
  Reading: Mermin 1.7-1.12

To Do

- Get your book
- Get your clicker
- Do Assignment 0 (due in ~45 hrs)

Add a picture.

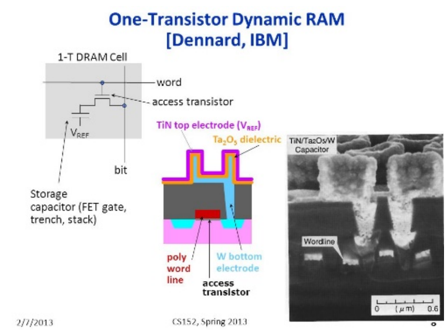
Why are we here?

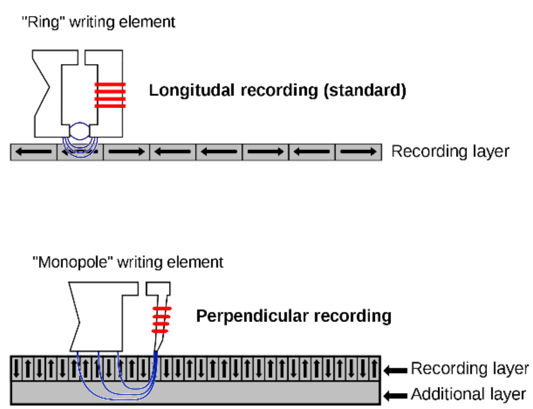
- The world obeys Quantum Theory (old news!)
- Computers that fully harness quantum effects could outperform classical ones.
- Building quantum computers is very hard, but not ridiculously, impossibly hard.
- We are at a special moment: beginning to build nontrivial quantum computers

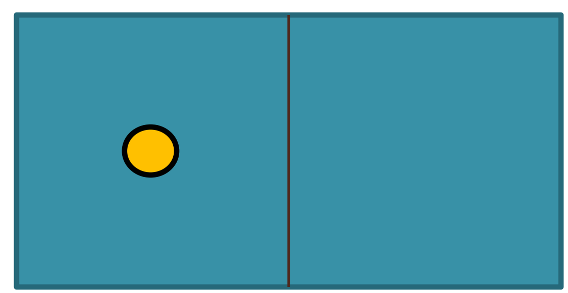
- **This course:** you will learn what a quantum computer is, why we think it'd be useful (quantum algorithms), and why we think it can be built (quantum error correction).

What is a bit?

Anything that can be in one of two states!







L R

Binary Dig**it**

What is a bit?

- Anything that can be in one of two states!

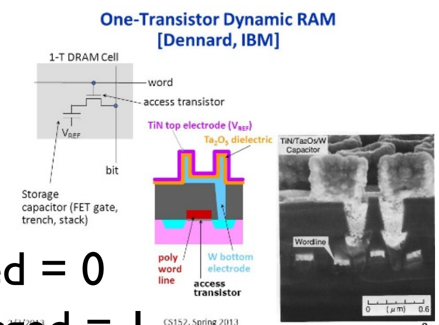
Might as well call these two states 0 and 1

Abstractly, a bit is just a variable that's either 0 or 1.

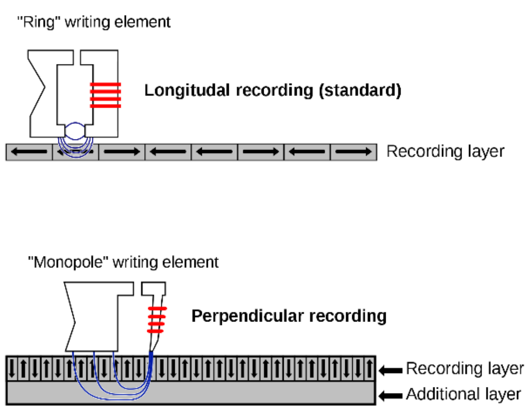
What is a bit?

Anything that can be in one of two states!

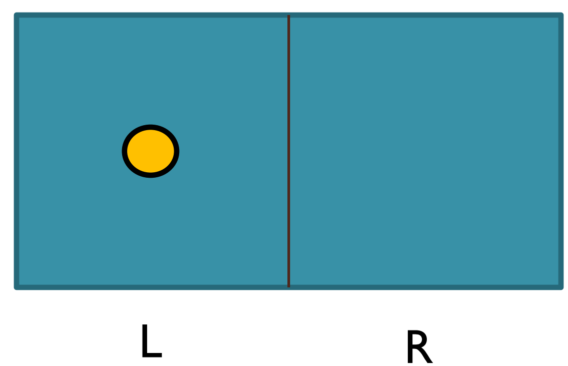
H = 0
T = 1



Charged = 0
Uncharged = 1



U = 0, D = 1



L = 0, R = 1

What is a bit?

- Anything that can be in one of two states!

Might as well call these two states 0 and 1

Abstractly, a bit is just a variable that's either 0 or 1.

Q: X is a bit. How many states can it be in?

What is a bit?

- Anything that can be in one of two states!

Might as well call these two states 0 and 1

Abstractly, a bit is just a variable that's either 0 or 1.

Q: X is a bit. How many different states could it be in?

A: two---either 0 or 1

Concept Question: Three Bits

- X_1 is a bit.
- X_2 is a bit.
- X_3 is a bit.

- Q: How many possible values are there for the bit string $X_1X_2X_3$?

A) 1 B) 3 C) 8 D) ∞

Concept Question: Three Bits

- X_1 is a bit.
- X_2 is a bit.
- X_3 is a bit.

- Q: How many possible values are there for the bit string $X_1 X_2 X_3$?

A) 1 B) 3 $\boxed{\text{C) 8}}$ D) ∞

Concept Question: Three Bits

Q: How many possible values are there for the bit string $X_1X_2X_3$?

A) 1 B) 3 $\boxed{\text{C) 8}}$ D) ∞

- Here are the possible states:

000, 001, 010, 011,
100, 101, 110, 111

Note: $8 = 2^3$

Representing bit-strings as vectors

- One bit: 0 or 1

  $|0\rangle$ or $|1\rangle$

  vectors: $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ or $|0\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

- Two Bits: 00, 01, 10, 11

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Tensor Products

- $$\begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \otimes \begin{bmatrix} y_0 \\ y_1 \end{bmatrix} \otimes \begin{bmatrix} z_0 \\ z_1 \end{bmatrix} = \begin{bmatrix} x_0 y_0 z_0 \\ x_0 y_0 z_1 \\ x_0 y_1 z_0 \\ x_0 y_1 z_1 \\ x_1 y_0 z_0 \\ x_1 y_0 z_1 \\ x_1 y_1 z_0 \\ x_1 y_1 z_1 \end{bmatrix}$$

Entries are 0's and 1's

Exactly one 1 in each vector on LHS means exactly one 1 in big vector on RHS

Manipulating Bits

One bit operation: NOT

NOT(x) = 1 if x=0
0 if x = 1

NOT(x) just flips the bit x.

More compactly: $NOT(x) = \bar{x}$

Manipulating Bits

One bit operation: NOT

NOT(x) = 1 if x=0
0 if x = 1

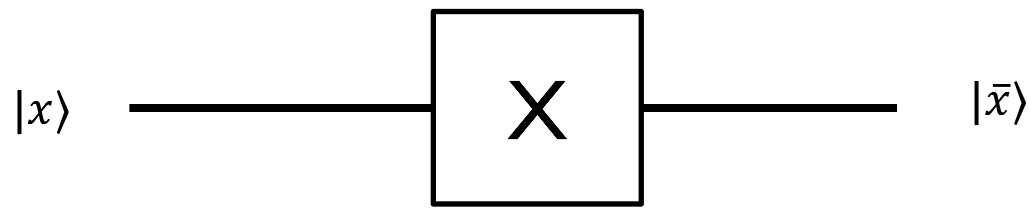
NOT(x) just flips the bit x.

More compactly: $NOT(x) = \bar{x}$

Manipulating Bits

- When represented as a vector
- $x \sim \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ or $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
- $x \sim \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$
- $\bar{x} \sim \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_0 \end{bmatrix}$
- Call $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
- As Kets: $|\bar{x}\rangle = X|x\rangle$

Flipping a bit: circuit diagram



Manipulating one bit

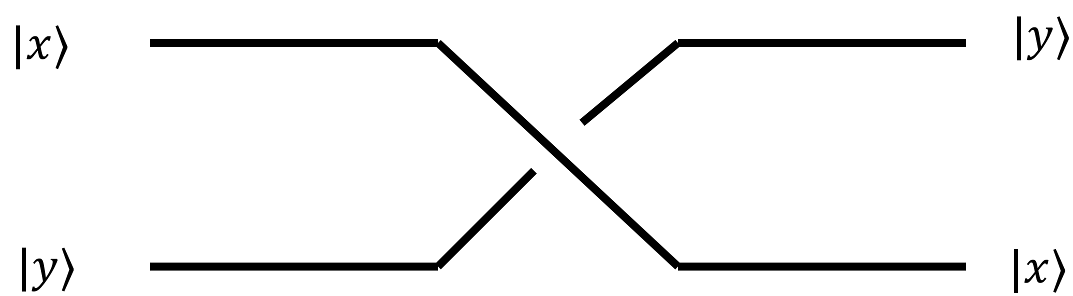
- If someone hands you a bit, there are two ways you can process it to give another bit:

  1) Leave it alone
  2) Flip it

That's it. If we want more interesting computations, we'd better have more bits.

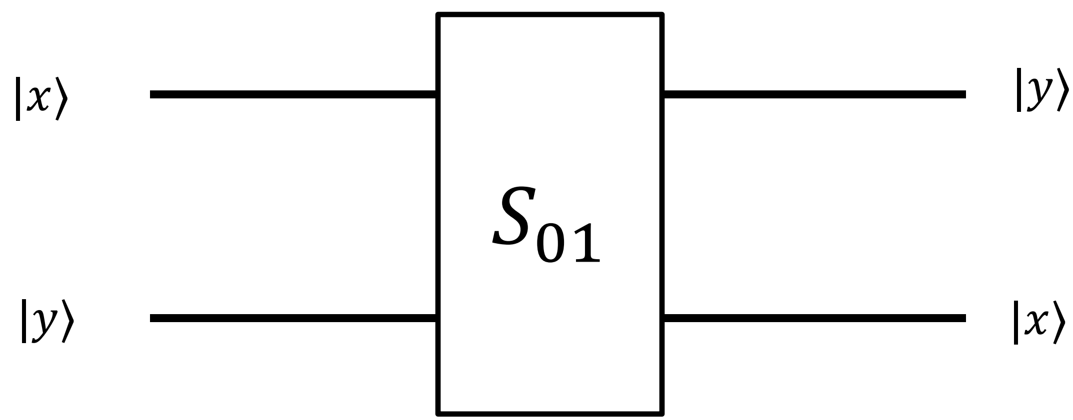
Swapping Two bits

- Given two bits, what can we do? Swap them!



Swapping Two bits

- Given two bits, what can we do? Swap them!



Swapping Two bits

- Given two bits, what can we do? Swap them!

$$S_{01}|x\rangle|y\rangle = |y\rangle|x\rangle$$

As a matrix

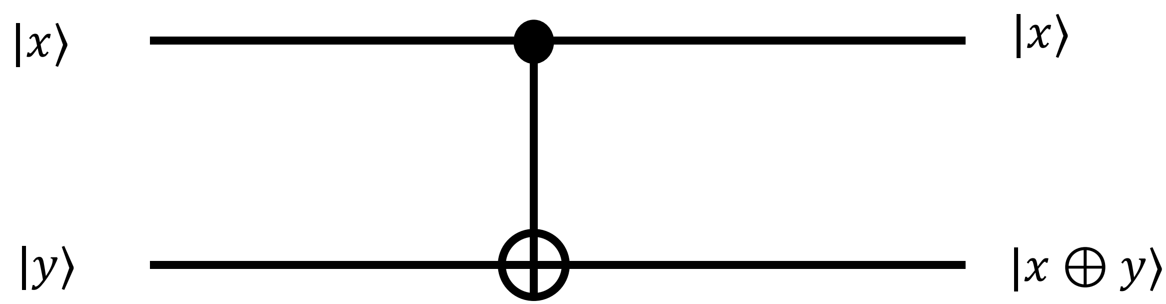
$$S_{01} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Controlled NOT

CNOT: $|x\rangle|y\rangle \rightarrow |x\rangle|x \oplus y\rangle$

where $x \oplus y = (x + y) mod\ 2$

If x = 0, leaves y alone. If x = 1, flips y bit.



Concept Test

CNOT: $|x\rangle|y\rangle \rightarrow |x\rangle|x \oplus y\rangle$

What matrix represents CNOT?

A) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

B) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

C) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$

D) $\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

Concept Test

CNOT: $|x\rangle|y\rangle \rightarrow |x\rangle|x \oplus y\rangle$

What matrix represents CNOT?

A) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

B) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

C) $\boxed{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}}$

D) $\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$

Concept Test: solution

- $\begin{array}{cc} \begin{bmatrix} \; \\ \; \\ \; \\ \; \end{bmatrix} & \begin{array}{c} 00 \\ 01 \\ 10 \\ 11 \end{array} \end{array}$ CNOT swaps 10 and 11

00→ 00, 01→ 01, 10→ 11, 11→ 10

The matrix that does this is

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$