

# Quantum Gate Emulation

AS043

**James Kiessling**  
**Daniel Williams**  
**Christian D'Ovidio**

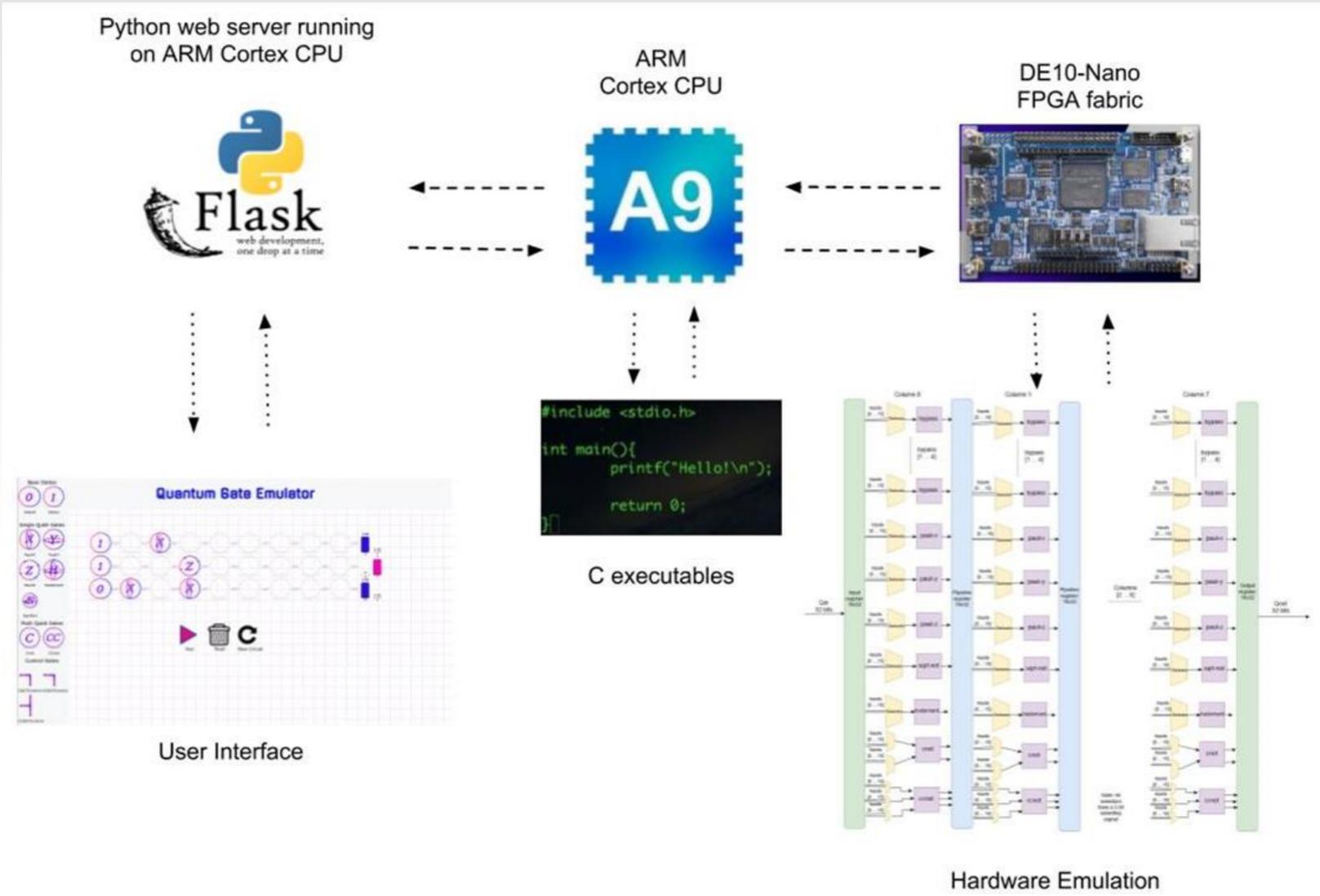
**Dr. Jien-Chung Lo**   **Dr. Bin Li**

## High Level Description

**AS043**

- ⊙ Goal: Create a simple, user-friendly sandbox for experimenting with quantum circuits
- ⊙ “Ideal” Quantum Gate Emulator
  - Ignores real world physical phenomena such as noise, entanglement

# High Level Block Diagram



## Quantum Bit (Qubit) Representation

- ⊙ Qubits represented by probability of measuring a specific state
- ⊙ General form:  $|v\rangle = a|0\rangle + b|1\rangle$ 
  - $a, b$  represent probability of measuring state  $|0\rangle, |1\rangle$  respectively
  - $a$  and  $b$  can both be complex numbers
  - Probability of measuring  $|0\rangle$  is  $|a|^2$  and probability of measuring  $|1\rangle$  is  $|b|^2$
- ⊙ In hardware, represented by 32-bit fixed-point signal
  - $a, b$  are 16 bits each
  - $a, b$  have a real and imaginary component - 8 bits each
  - Qubit  $\leq |a.\text{real}| |a.\text{imag}| |b.\text{real}| |b.\text{imag}|$
  - MSB has weight of  $2^0$ , subsequent bits have negative decreasing powers
  - Uses 2's complement representation for signed numbers
  - Ex: 01100000 =  $0*2^0 + 1*2^{-1} + 1*2^{-2} + 0*2^{-3} + \dots + 0*2^{-7} = 0.75$
  - Ex: 11100000 =  $-1*2^0 + 1*2^{-1} + 1*2^{-2} + 0*2^{-3} + \dots + 0*2^{-7} = -0.25$

## Quantum Gates

- **Pauli-X** → Acts as a NOT gate by switching probability of  $|0\rangle$  and  $|1\rangle$
- **Pauli-Y** → Maps  $|0\rangle$  to  $i|1\rangle$  and  $|1\rangle$  to  $-i|0\rangle$ , rotation around y-axis of Bloch sphere by  $\pi$  radians
- **Pauli-Z** → Leaves  $|0\rangle$  unchanged and maps  $|1\rangle$  to  $-|1\rangle$ , rotation around z-axis of Bloch sphere by  $\pi$  radians
- **Hadamard** → Places qubit into a state of superposition by making probabilities of  $|0\rangle$  and  $|1\rangle$  equal
- **Sqrt(NOT)** → Two in a row acts as NOT gate, mapping follows matrix:  $\frac{1}{\sqrt{2}} \begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}$ .
- **CNOT** → If first input bit is in state  $|1\rangle$ , NOT second input bit
- **CCNOT (Toffoli)** → If first and second input bits are both in state  $|1\rangle$ , NOT third input bit

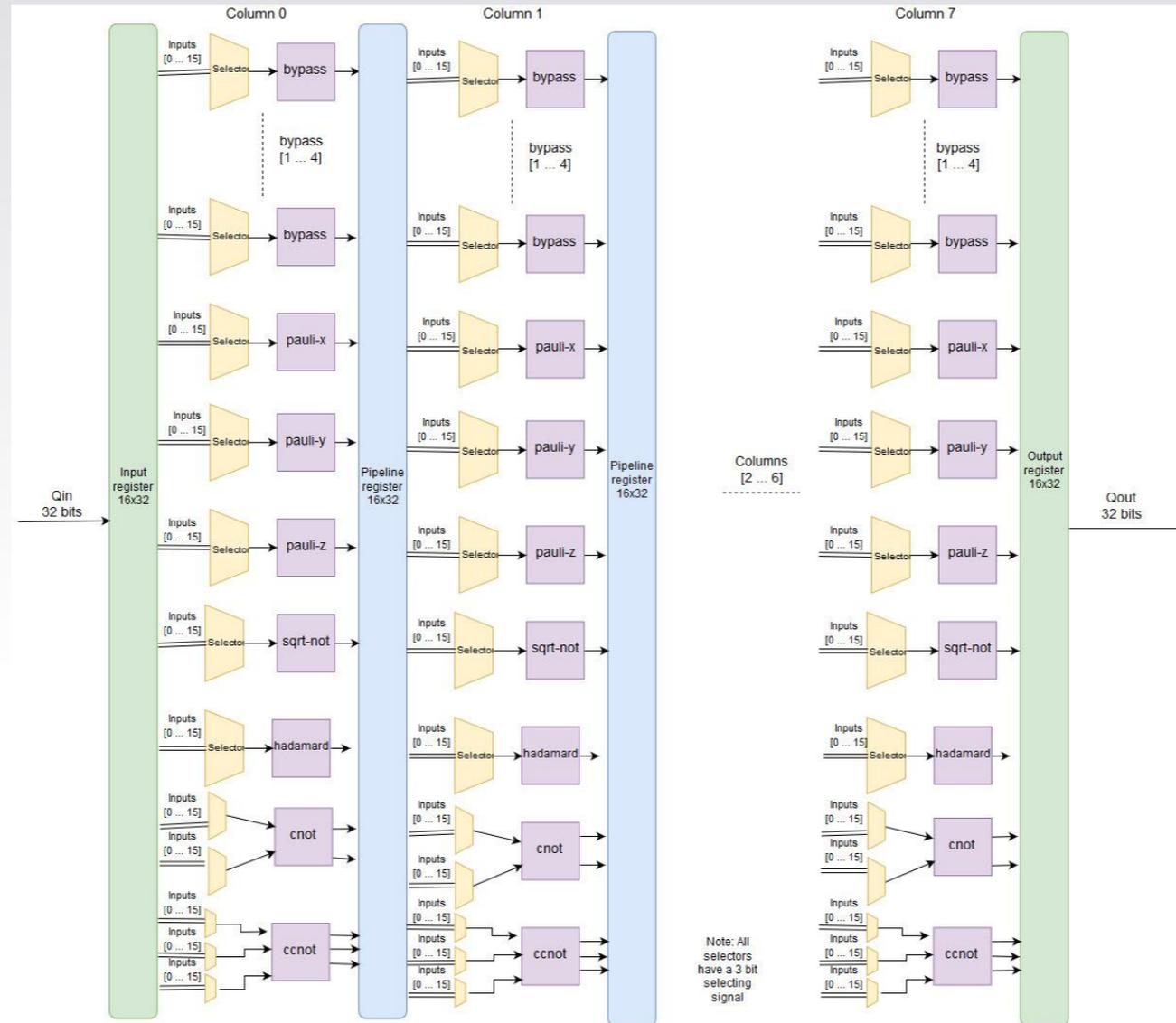
## Quantum Emulation Circuit

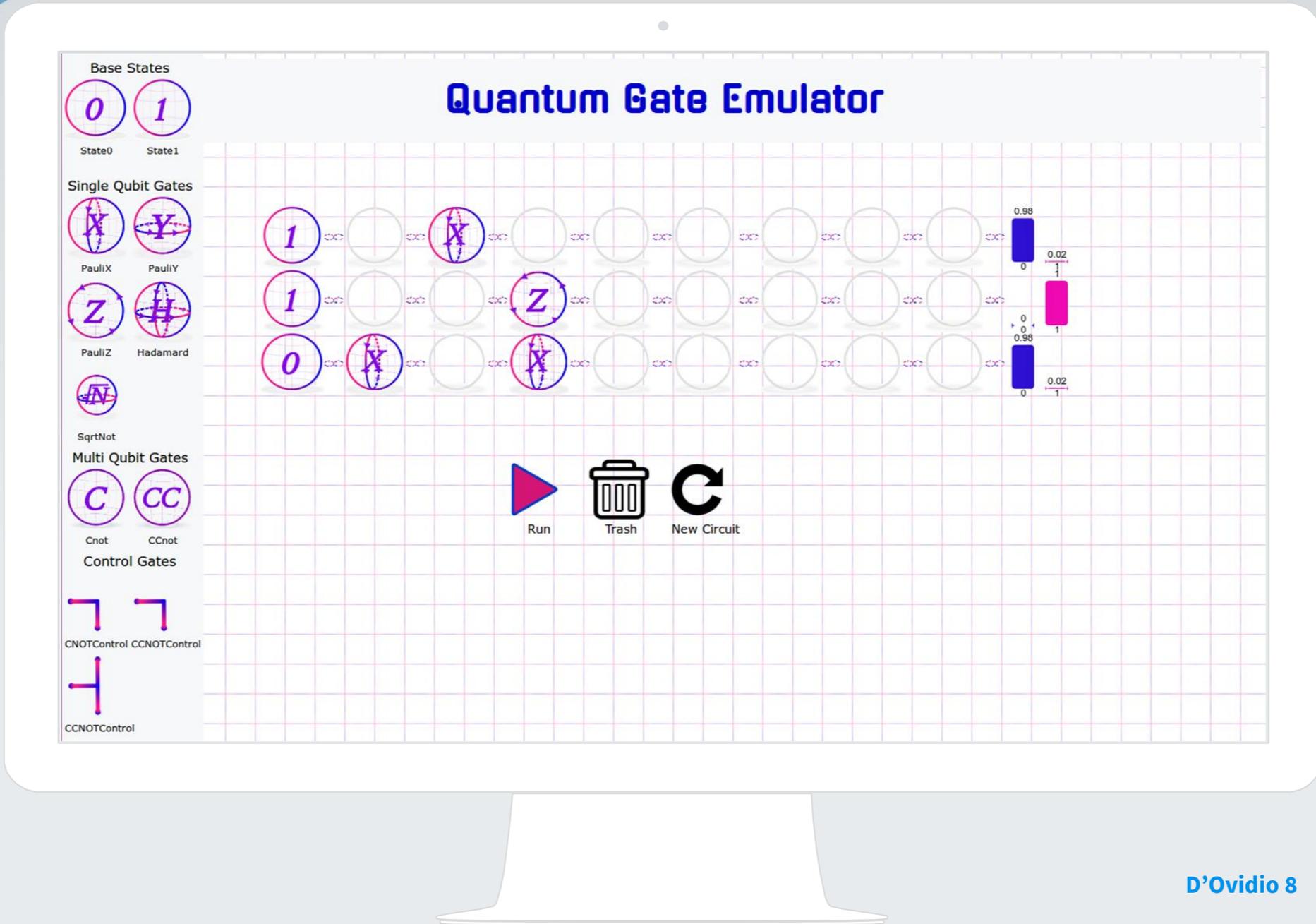
**AS043**

- ⊙ Written in VHDL, programmed for the DE10-nano development kit
- ⊙ Combinational circuit with pipeline architecture
- ⊙ Comprised of 8 pipeline stages with 12 gates at each stage
- ⊙ Mux for each gate at each stage to select input
  - Mux connects input of a gate to the output of a specific gate at the previous stage
  - Input at first pipeline stage is initial qubit state
  - Input at every subsequent stage is the output of previous stage
- ⊙ Each mux addressed with 8 bits - <1cccrrrr>
  - 'ccc' → 3 bits to indicate column (i.e. pipeline stage)
  - 'rrrr' → 4 bits to indicate row (i.e. gate)
  - '1' MSB to indicate mux addressing

# Circuit Block Diagram

AS043





# Gates

Base States

State0: 0    State1: 1

Single Qubit Gates

PauliX    PauliY

PauliZ    Hadamard

SqrtNot

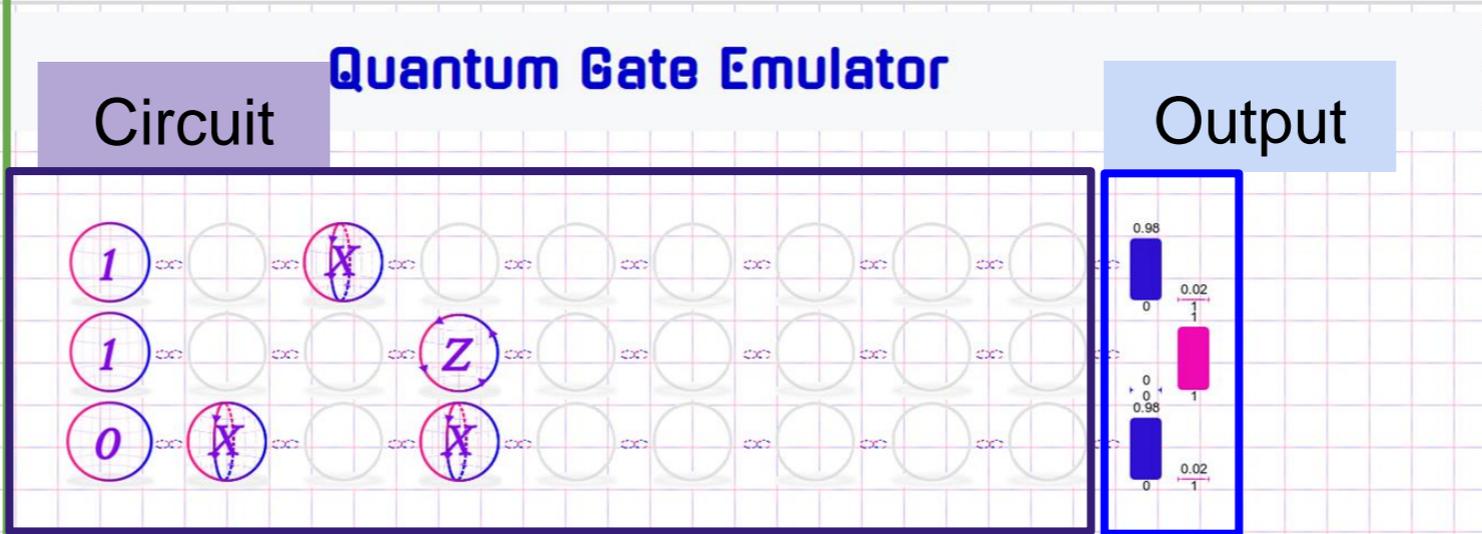
Multi Qubit Gates

Cnot    CCnot

Control Gates

CNOTControl    CCNOTControl

CCNOTControl



**Buttons**

Run    Trash    New Circuit

## Web Server

**AS043**

- ⊙ Written in Python using Flask
- ⊙ Runs on DE10-nano on-board processor
- ⊙ Hosts website on local network only
- ⊙ Handles user interaction:
  - Adding gates to emulation
  - Removing gates from emulation
  - Changing initial input qubit states
  - Requesting resulting qubit states at output
  - Re-initializing the emulation

- ⦿ Start states of  $|0\rangle$  and  $|1\rangle$  can be set
- ⦿ Gates being dragged into circuit trigger server communication, so VHDL circuit is always up-to-date
- ⦿ Circuit is 'run' once to minimize costly memory operations
- ⦿ Output is an easy to understand probability graph, which directly represents the physical characteristics of the qubit

- ⦿ Handles HPS-FPGA communication
- ⦿ Receives user input from website as JSON data
- ⦿ Parses JSON data to determine which actions to take
- ⦿ Invokes compiled C executables to interact with FPGA running emulation
  - “read\_output” → Read the final qubit state at the output stage of emulation
  - “write\_input” → Change the initial state of the qubit at the input stage of emulation
  - “write\_gate” → Change the gate at the specified stage of the emulation

## C Programs

- ⊙ Compiled C executables to perform generic operations for FPGA communication
- ⊙ Open FPGA memory map, read/write specific PIO addresses
  - PIO addresses correspond to the addresses of Qsys-generated PIO signals connected to emulation
- ⊙ Programs follow correct handshaking protocol for reading/writing
- ⊙ “Read\_output”
  - Usage: ./read\_output OUTPUT\_REGISTER
  - Reads output signal at location \$OUTPUT\_REGISTER
- ⊙ “Write\_input”
  - Usage: ./write\_input STATE ROW
  - Writes STATE to the specified ROW of the input stage
- ⊙ “Write\_gate”
  - Usage: ./write\_gate COL ROW MUXVAL
  - Writes the specified MUXVAL to the selecting mux of the gate at the specified (COL, ROW)

## Top-Level Communication Circuit

- ⊙ Sequential circuit for communication between web server and emulation
- ⊙ Parallel I/O (PIO) external signals created through Qsys tool
  - **Qin** → 32-bit input signal used for general-purpose write operations
  - **Qout** → 32-bit output signal connected to output stage of pipeline (i.e. final qubit state)
  - **read\_en** → 1-bit read enable signal
  - **write\_en** → 1-bit write enable signal
  - **reset** → 1-bit reset signal, active low
- ⊙ Qin signal used for all write operations (gate mux, initial qubit states)
- ⊙ One 32-bit control register used for writing to 256 32-bit data registers
- ⊙ read\_en and write\_en signals for handshaking
- ⊙ All PIO signals can be read/written using C programs

# Simple Circuit

AS043

## Quantum Gate Emulator

The interface includes the following components:

- Base States:** State0 (0) and State1 (1).
- Single Qubit Gates:** PauliX, PauliY, PauliZ, Hadamard, and SqrtNot.
- Multi Qubit Gates:** Cnot and CCnot.
- Qubit Lines:** Three horizontal lines representing qubits, each starting with a '0' in a circle.
- Control Panel:** Run (red play button), Trash (trash can icon), and New Circuit (circular refresh icon).

## Quantum Gate Emulator

**Base States**



0  
State0



1  
State1

**Single Qubit Gates**



X  
Pauli X

The Pauli-X gate acts on a single qubit. It is the quantum equivalent of the NOT gate for classical computers. It equates to a rotation of the Bloch sphere around the X-axis by pi radians. It maps |0> to |1> and |1> to |0>.



Y  
Pauli Y



Z  
Pauli Z



H  
Hadamard

  
SqrtNot

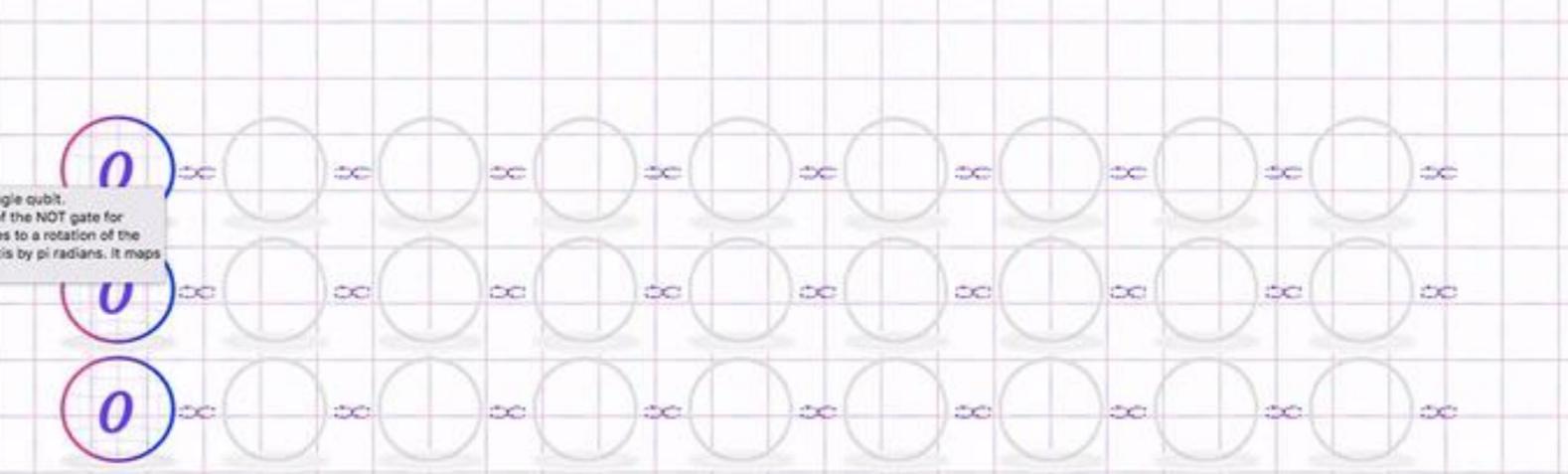
**Multi Qubit Gates**



C  
Cnot



CC  
CCnot



Run    Trash    New Circuit

# Multi Qubit Gates

AS043

## Quantum Gate Emulator

The interface features a grid of qubits on a light blue background. On the left, a sidebar lists gate options:

- Base States:** State0 (0) and State1 (1).
- Single Qubit Gates:** PauliX, PauliY, PauliZ, Hadamard, and SqrtNot.
- Multi Qubit Gates:** Cnot and CCnot.

The main grid contains three rows of qubits. The first qubit in each row is highlighted with a blue border and contains the number '0'. The remaining qubits in each row are empty circles. At the bottom of the grid are three control buttons: a red play button labeled 'Run', a trash can icon labeled 'Trash', and a circular refresh icon labeled 'New Circuit'.

AS043

## Quantum Gate Emulator

The interface features a grid of qubits on a grid background. The top row of qubits is highlighted in blue and contains the number '0'. The second and third rows of qubits are highlighted in purple and also contain the number '0'. The rest of the qubits in the grid are greyed out. To the left of the grid is a sidebar with various quantum gates and states:

- Base States:** State0 (0) and State1 (1).
- Single Qubit Gates:** PauliX, PauliY, PauliZ, Hadamard, and SqrtNot.
- Multi Qubit Gates:** Cnot and CCnot.

At the bottom of the interface are three control buttons: a red play button labeled 'Run', a trash can icon labeled 'Trash', and a circular refresh icon labeled 'New Circuit'.

AS043

### Quantum Gate Emulator

**Base States**

State0 State1

**Single Qubit Gates**

PauliX PauliY

PauliZ Hadamard

SqrtNot

**Multi Qubit Gates**

Cnot CCnot

Run Trash New Circuit

# Quantum Gate Emulation

AS043

**James Kiessling**  
**Daniel Williams**  
**Christian D'Ovidio**

**Dr. Jien-Chung Lo**   **Dr. Bin Li**