InnovateFPGA 2018 Grand Final

Quantum Gate Emulation

AS043

James Kiessling
Daniel Williams
Christian D'Ovidio

Dr. Jien-Chung Lo Dr. Bin Li

InnovateFPGA 2018 Grand Final

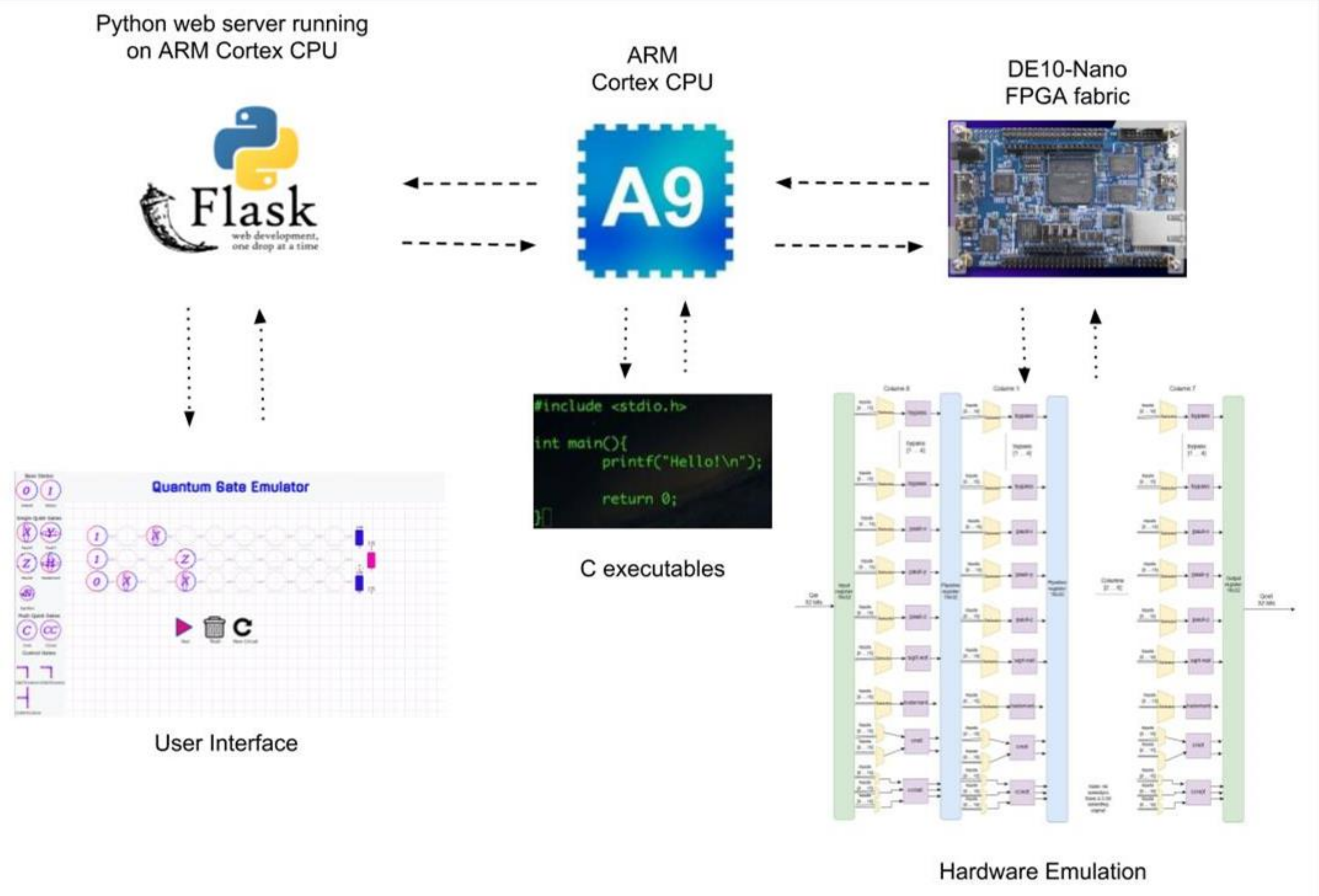
AS043

High Level Description

- 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

AS043

High Level Block Diagram



AS043

Quantum Bit (Qubit) Representation

- Qubits represented by probability of measuring a specific state
- General form: |v} = a|0} + b|1}
  - a, b represent probability of measuring state |0}, |1} respectively
  - a and b can both be complex numbers
  - Probability of measuring |0} is $|a|^2$ and probability of measuring |1} 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 <= | a.real | a.imag | b.real | b.imag |
  - MSB has weight of 2^0, subsequent bits have negative decreasing powers
  - Uses 2's complement representation for signed numbers
  - Ex: $01100000 = 0*2^0 + 1*2^{-1} + 1*2^{-2} + 0*2^{-3} + \ldots + 0*2^{-7} = 0.75$
  - Ex: $11100000 = -1*2^0 + 1*2^{-1} + 1*2^{-2} + 0*2^{-3} + \ldots + 0*2^{-7} = -0.25$

InnovateFPGA
2018
Grand Final

AS043

Quantum Gates

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

Quantum Emulation Circuit

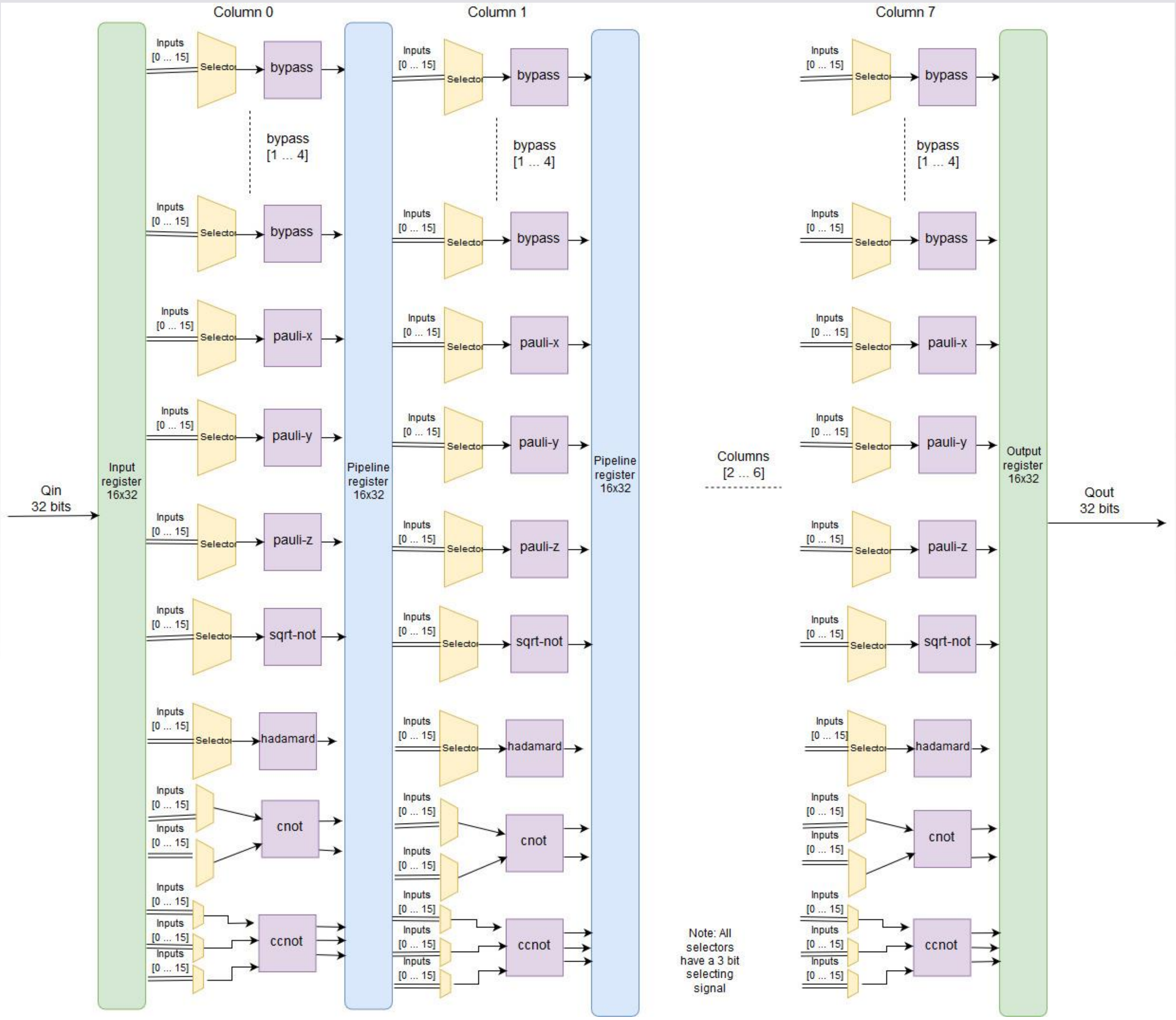
AS043

- 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

InnovateFPGA 2018 Grand Final

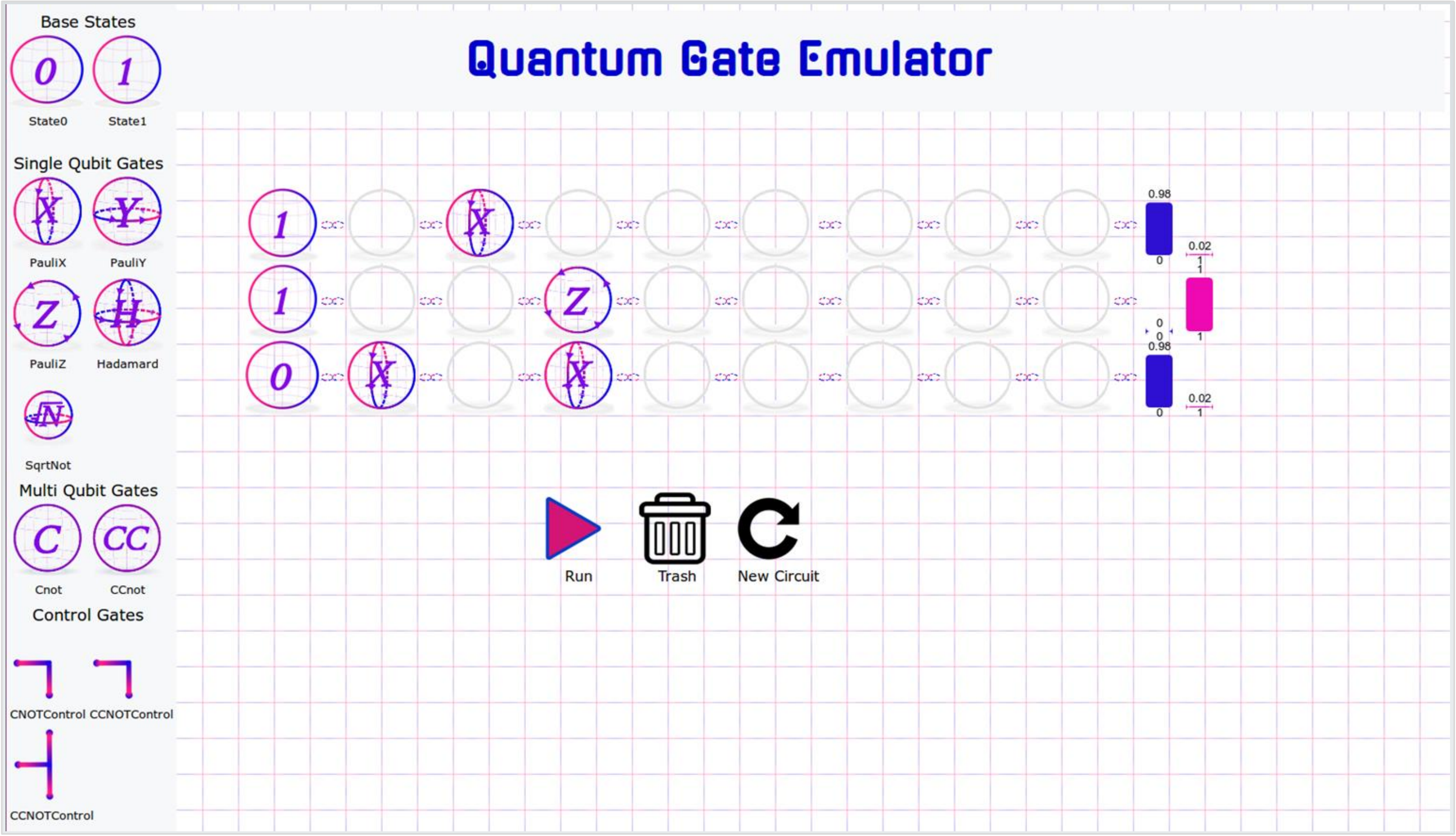
AS043

Circuit Block Diagram



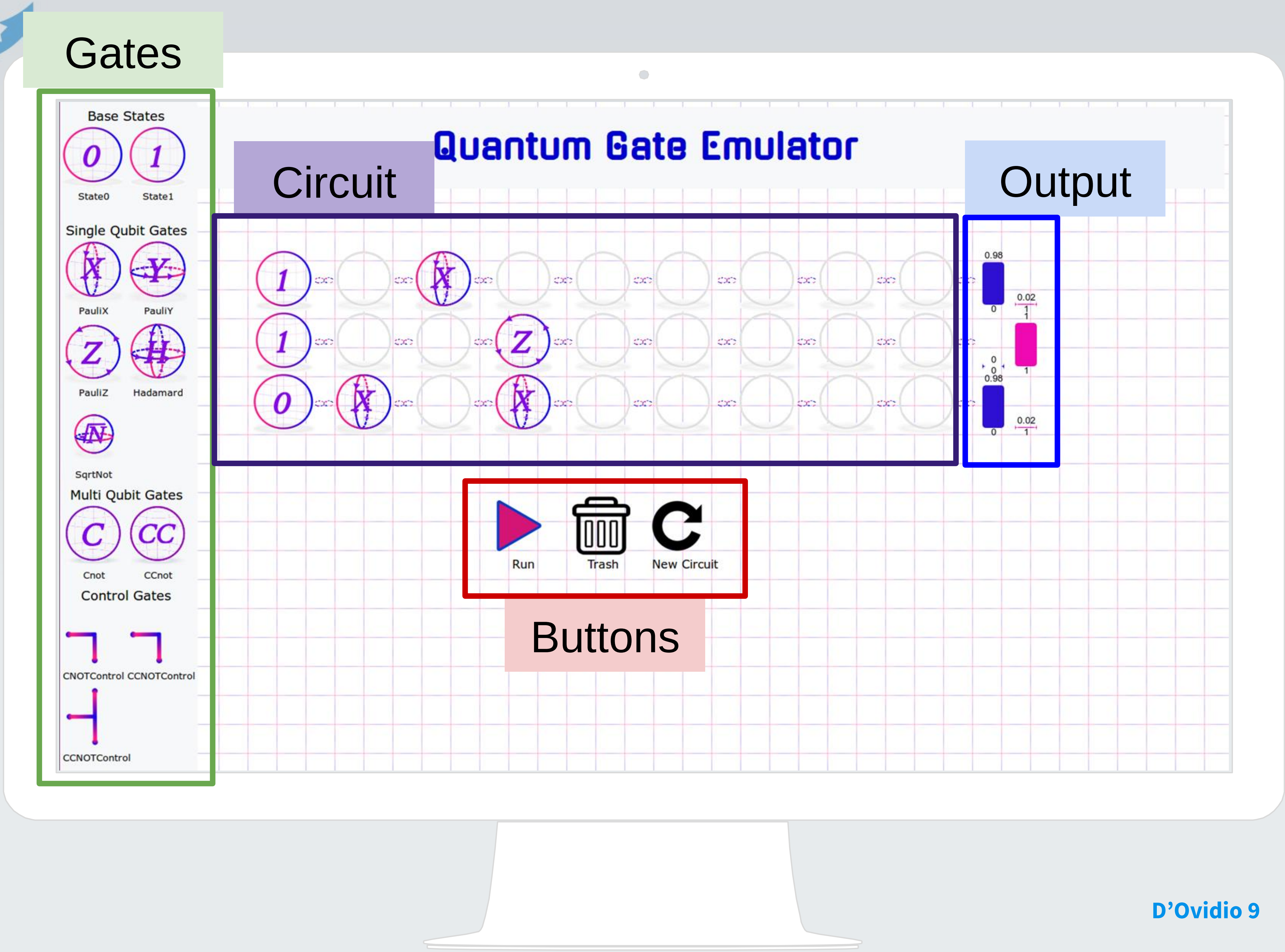
InnovateFPGA
2018
Grand Final

AS043



InnovateFPGA
2018
Grand Final

AS043



Web Server

- 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

InnovateFPGA
2018
Grand Final

AS043

Web App Functionality

- Start states of $|0\}$ and $|1\}$ 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

Web Server Continued

- 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

AS043

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)

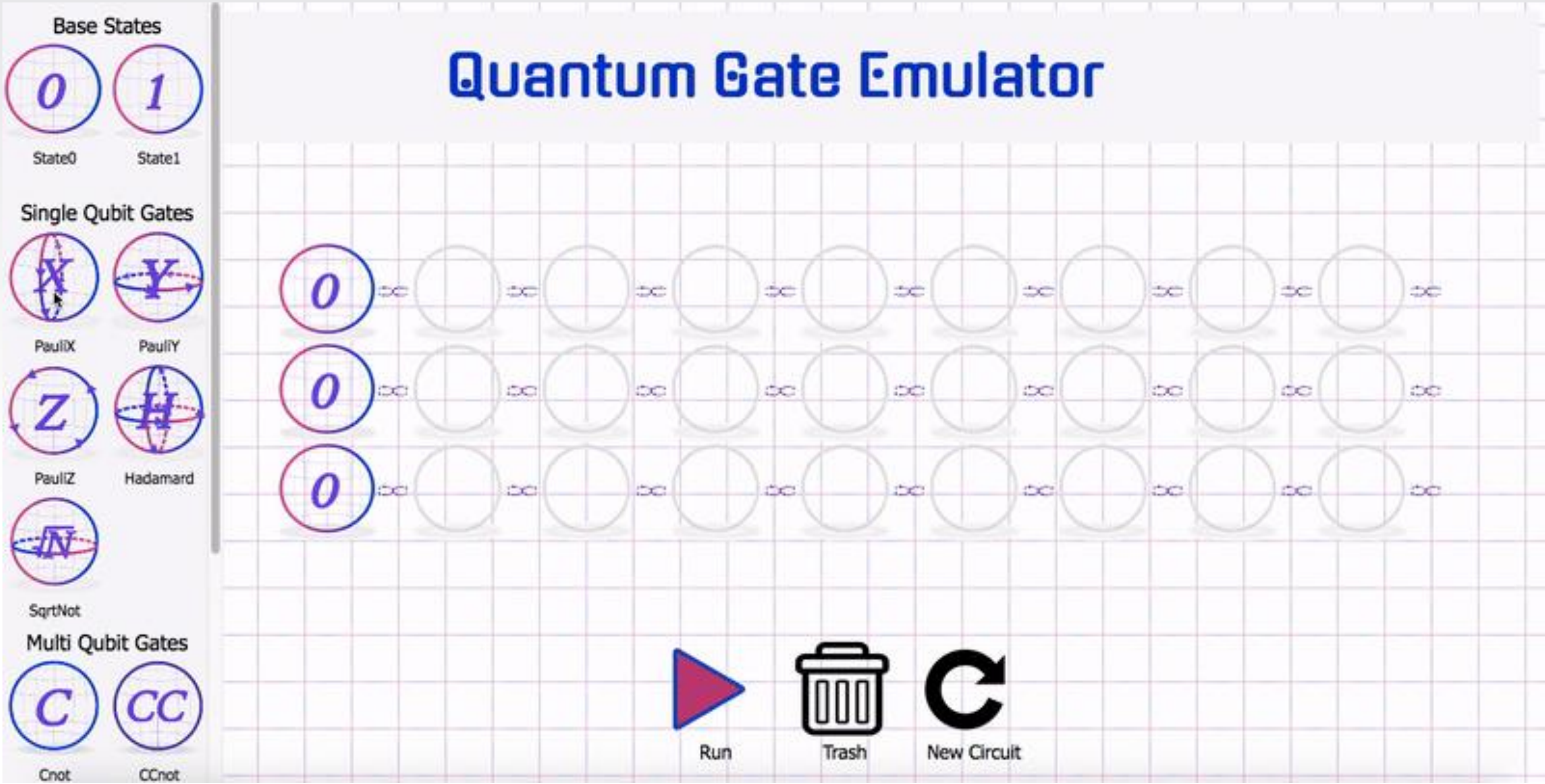
AS043

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

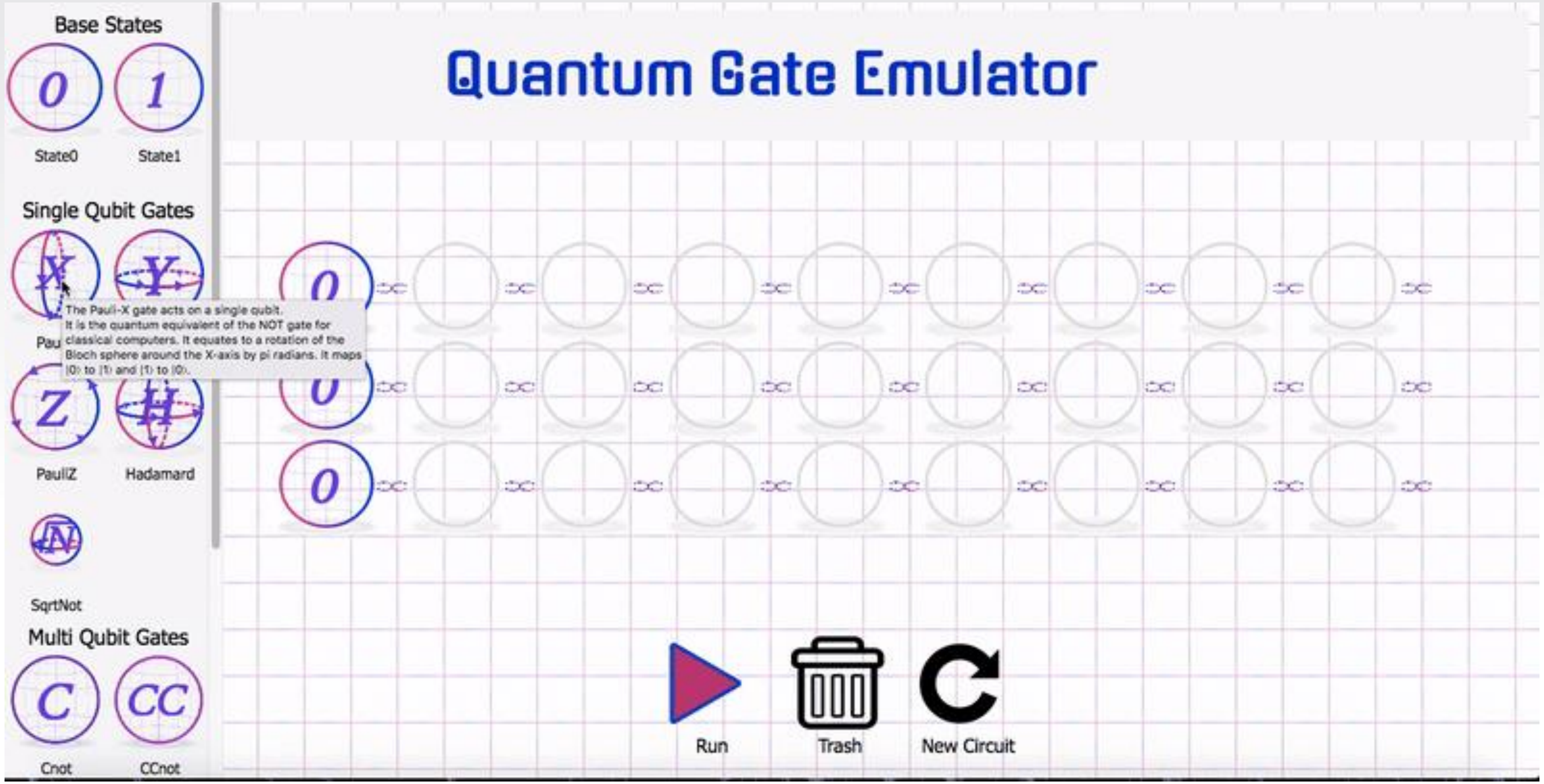
Simple Circuit

AS043



pauliX

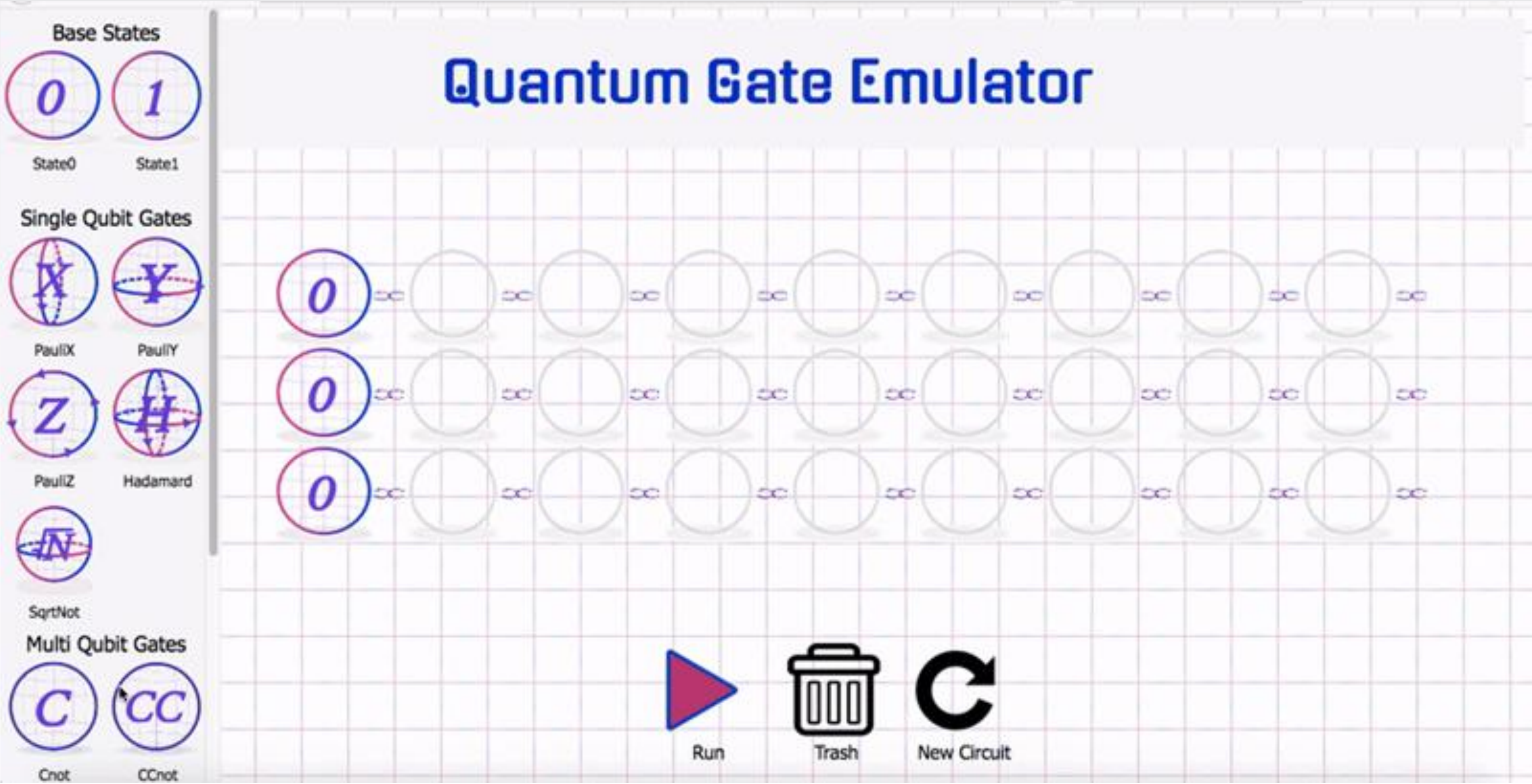
AS043



InnovateFPGA
2018
Grand Final

Multi Qubit Gates

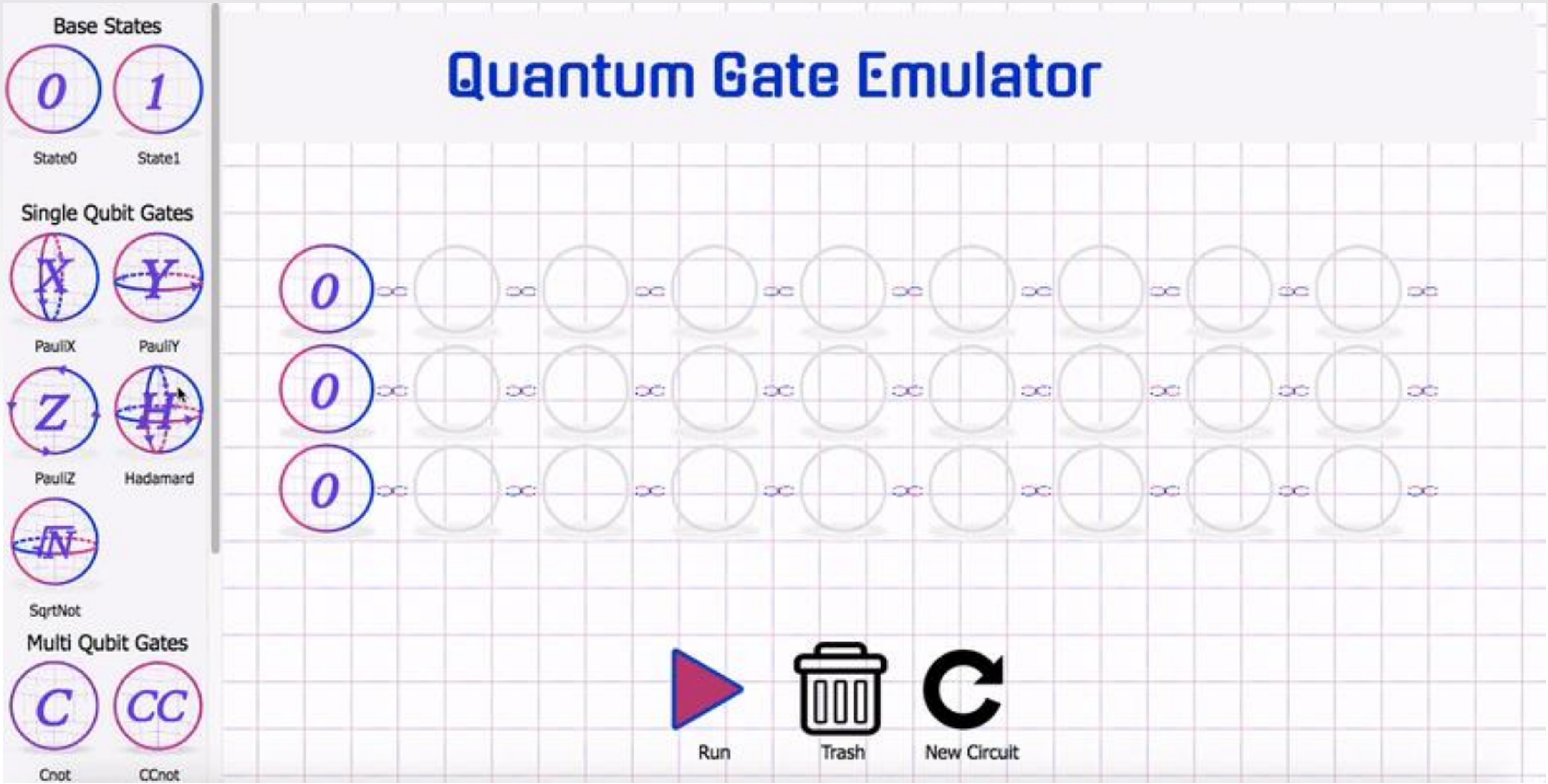
AS043



InnovateFPGA
2018
Grand Final

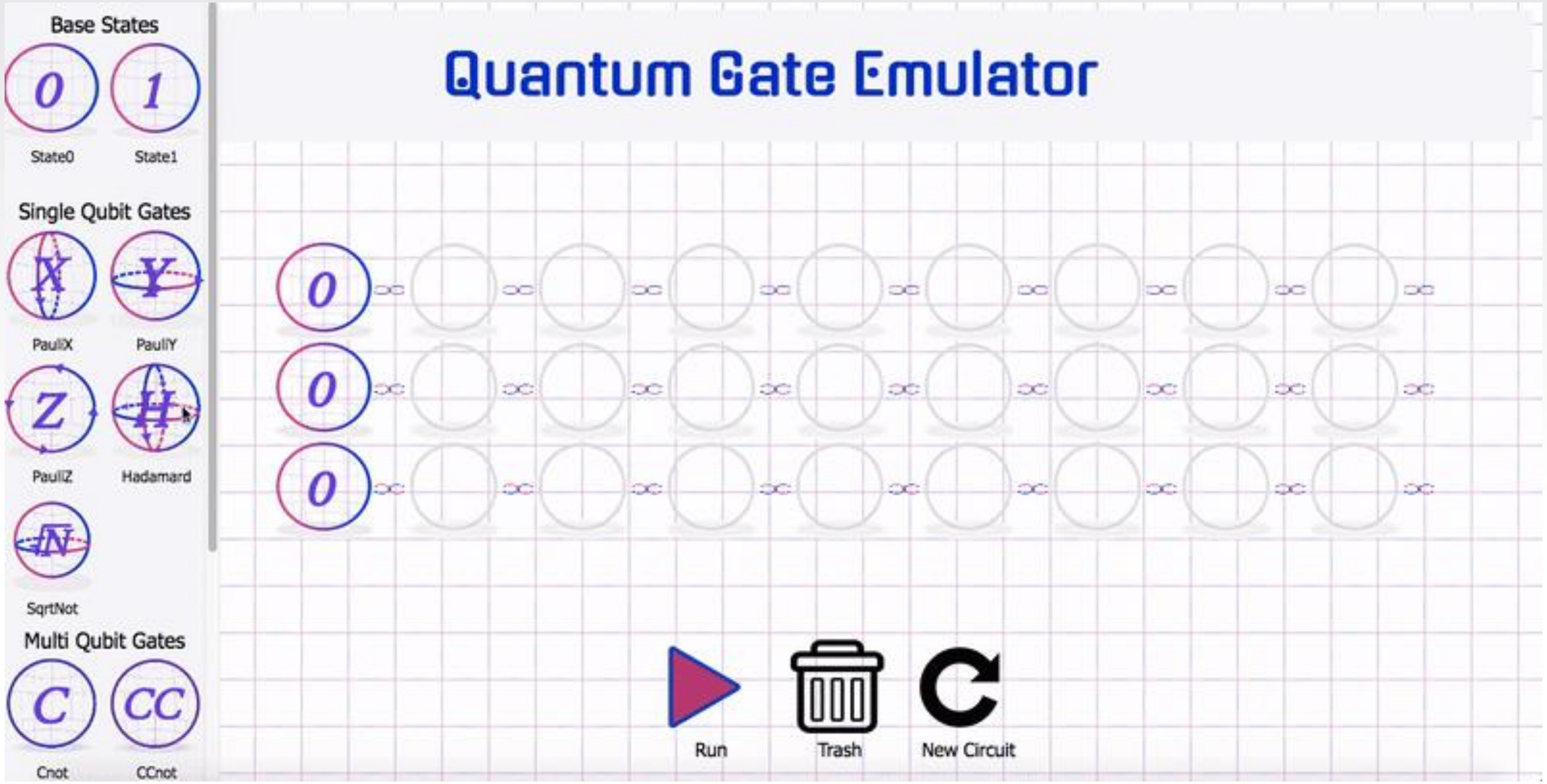
Superposition

AS043



Issue Handling

AS043



InnovateFPGA
2018
Grand Final

Quantum Gate Emulation

AS043

James Kiessling
Daniel Williams
Christian D'Ovidio

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