

Title : Implementation of 16 Qubit Random Number Generator

```
In [12]: #16qubit random number genertor
from qiskit import QuantumCircuit, Aer, execute
from qiskit.visualization import circuit_drawer

# Create a quantum circuit with 16 qubits
circuit = QuantumCircuit(16, 16)

# Apply Hadamard gates to put all qubits in superposition
circuit.h(range(16))

# Measure all qubits
circuit.measure(range(16), range(16))

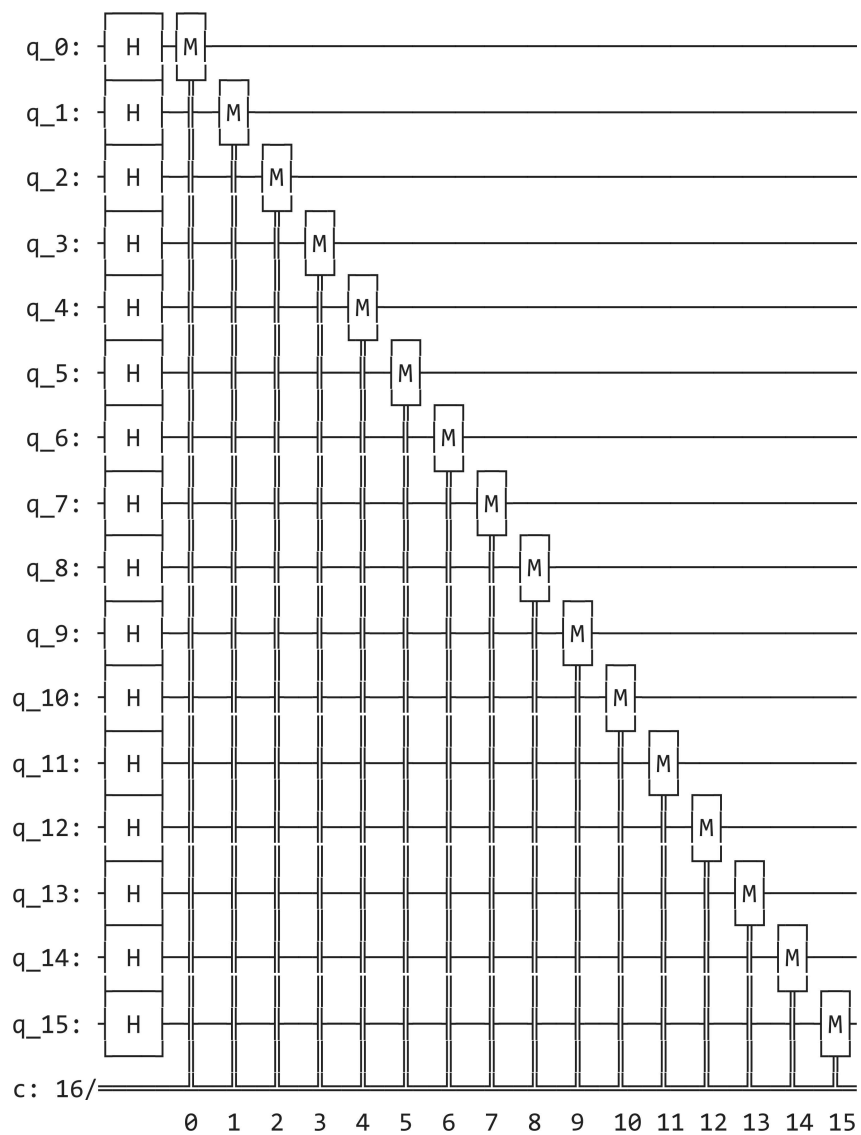
# Visualize the circuit
print(circuit)
circuit_drawer(circuit, output='mpl')

# Simulate the quantum circuit using the QASM simulator
simulator = Aer.get_backend('qasm_simulator')
job = execute(circuit, simulator, shots=1)
result = job.result()
counts = result.get_counts(circuit)

# Extract the random number from the measurement outcome
random_number = int(list(counts.keys())[0], 2)

# Convert the random number to binary representation
binary_number = bin(random_number)[2:].zfill(16)

print("Random number (decimal):", random_number)
print("Random number (binary):", binary_number)
```



Random number (decimal): 27337

Random number (binary): 0110101011001001