notebook

November 14, 2024

[1]:
```
from sympy.physics.quantum import qapply
from sympy.physics.quantum.gate import HadamardGate
from sympy.physics.quantum.qubit import Qubit
from random import getrandbits

from util.measure_all import measure_all_oneshot
from util.util import zero
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[1], line 6
      3 from sympy.physics.quantum.qubit import Qubit
      4 from random import getrandbits
----> 6 from util.measure_all import measure_all_oneshot
      7 from util.util import zero

ModuleNotFoundError: No module named 'util'
```

[6]:
```
def get_alice_bits(size):
    bits = []
    for i in range(size):
        bits.append(getrandbits(1))
    return bits
```

This is a test

[10]:
```
def b92_alice_part(alice_bits):
    """
    :param alice_bits: Alice bits
    :return alice_sent: alice measured qubits
    """

    # get bits and bases Alice side
    print(f"Alice bits: {alice_bits}")

    # we use a non-orthogonal base
    # {|→>, |↗>} = {zero, plus}
```

```
    alice_sent = [qapply(HadamardGate(0) * Qubit(0)) if b == 1 else zero for b␣
↪in alice_bits]

    print(f"Alice sent: {alice_sent}")
    return alice_sent
```

Let's get Alice random bits

[9]:
```
N=8
get_alice_bits(N)
```

[9]:
```
[0, 1, 0, 1, 1, 0, 0, 1]
```