nature portfolio

https://doi.org/10.1038/s41586-025-09848-5

Supplementary information

A fault-tolerant neutral-atom architecture for universal quantum computation

In the format provided by the authors and unedited

Supplementary Information

Contents

Error Model 2

Repeated QEC stim circuit 4

Repeated QEC experimental command strings 15

Tesseract code experimental command strings 40

Summary of decoders 54

Error model parameters

Supplementary Table I: Error model

Noise type	Error rate	Error type in Fig. 2f
idle_loss_rate	0.09e-6	T1/T2
idle_error_rate	(0.1e-6, 0.1e-6, 0.5e-6)	T1/T2
entangling_zone_error_rate	(5e-5, 5e-5, 10e-4)	CZ Pauli
entangling_gate_error_rate	(5e-5, 5e-5, 5e-4, 5e-5, 0, 0, 0, 5e-5, 0, 0, 0, 5e-4, 0, 0, 10e-4)	CZ Pauli
entangling_gate_loss_rate	0.00183	CZ loss
single_qubit_error_rate	(0.2e-3, 0.6e-3, 0.2e-3)	SQ
reset_error_rate	3e-3	SPAM
measurement_error_rate	5e-3	SPAM
reset_loss_rate	6e-3	SPAM loss
measurement_loss_rate	6e-3	SPAM loss
ancilla_idle_loss_rate	0.09e-6	T1/T2
ancilla_idle_error_rate	(0.5e-6, 0.5e-6, 1e-6)	T1/T2
ancilla_reset_error_rate	3e-3	SPAM
ancilla_measurement_error_rate	5e-3	SPAM
ancilla_reset_loss_rate	1.1e-2	SPAM loss
ancilla_measurement_loss_rate	6e-3	SPAM loss

For entries with 3 numbers, these correspond to:

X, Y, Z

For entangling_gate_error_rate, these correspond to:

IX, IY, IZ,

XI, XX, XY, XZ,

YI, YX, YY, YZ,

ZI, ZX, ZY, ZZ

Pseudocode to add noise to circuit operation

If operation is measurement:

- For each measured qubit:
    - If measured qubit is a data qubit:
        - Apply X error before measurement with probability measurement_error_rate
        - Lose qubit with probability measurement_loss_rate

    - If measured qubit is an ancilla qubit:
        - Apply X error before measurement with probability ancilla_measurement_error_rate
        - Lose qubit with probability measurement_loss_rate

If operation is entangling gate:

- For each pair of qubits doing an entangling gate:
    - Apply two-qubit depolarizing channel after gate with probabilities entangling_gate_error_rate
    - Lose qubit after gate with probability entangling_gate_loss_rate
- For each qubit in the entangling zone not doing a gate:
    - Apply a single-qubit depolarizing channel with probabilities entangling_zone_error_rate
    - Lose qubit with probability entangling_gate_loss_rate

If operation is qubit initialization:

- For each initialized qubit:
    - If initialized qubit is a data qubit:
        - Apply X error after initialization with probability reset_error_rate
        - Lose qubit with probability reset_loss_rate
    - If initialized qubit is an ancilla qubit
        - Apply X error after initialization with probability ancilla_reset_error_rate
        - Lose qubit with probability ancilla_reset_loss_rate

If operation is single qubit gate:

- For each qubit doing a single qubit gate:
    - Apply single-qubit depolarizing channel after gate with probabilities single_qubit_error_rate

If operation is idle (input parameter: idle_duration):

- For all data qubits:
    - Apply single-qubit depolarizing channel with probabilities
      $$1-(1-\text{idle\_error\_rate})^{\text{idle\_duration}}$$
    - Lose qubit with probability
      $$1-(1-\text{idle\_loss\_rate})^{\text{idle\_duration}}$$
- For all ancilla qubits:
    - Apply single-qubit depolarizing channel with probabilities
      $$1-(1-\text{ancilla\_idle\_error\_rate})^{\text{idle\_duration}}$$
    - Lose qubit with probability
      $$1-(1-\text{ancilla\_loss\_error\_rate})^{\text{idle\_duration}}$$

Repeated QEC stim circuit

```
R 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
X_ERROR(0.003) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
SQRT_Y_DAG 2 4 6 13 15 22 24 26 33 35 42 44 46
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 4 6 13 15 22 24 26 33 35 42 44 46
TICK
PAULI_CHANNEL_1(0.000128492, 0.000128492, 0.000642294) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32
33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000642294, 0.000642294, 0.00128418) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30
31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27
28 29 30 31 37 38 39 40 41 47 48
R 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
X_ERROR(0.003) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 7 9 11 19 21 27 29 31 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 7 9 11 19 21 27 29 31 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32
33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30
31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27
28 29 30 31 37 38 39 40 41 47 48
CZ 2 0 4 1 13 8 15 10 22 18 24 20 33 28 35 30 42 38 44 40 7 12 9 14 11 16 19 23 21 25 27 32 29 34 31 36
39 43 41 45
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 2 0 4 1 13 8 15
10 22 18 24 20 33 28 35 30 42 38 44 40 7 12 9 14 11 16 19 23 21 25 27 32 29 34 31 36 39 43 41 45
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 3 5 6 37 46 47 48 17 26
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32
33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30
31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27
28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43
44 45 46
Y 11 31
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 11 31
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32
33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30
31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27
28 29 30 31 37 38 39 40 41 47 48
CZ 3 8 5 10 12 18 14 20 23 28 25 30 32 38 34 40 43 47 45 48 7 2 9 4 11 6 19 13 21 15 27 22 29 24 31 26
39 33 41 35
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 3 8 5 10 12 18
14 20 23 28 25 30 32 38 34 40 43 47 45 48 7 2 9 4 11 6 19 13 21 15 27 22 29 24 31 26 39 33 41 35
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 0 1 36 37 42 44 46 16 17
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32
33 34 35 36 42 43 44 45 46
```

PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 11 31
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 11 31
Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
SQRT_Y 17 37
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 17 37
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 3 0 5 1 14 8 16 10 23 18 25 20 34 28 36 30 43 38 45 40 7 13 9 15 17 22 19 24 21 26 27 33 29 35 37 42 39 44 41 46
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 3 0 5 1 14 8 16 10 23 18 25 20 34 28 36 30 43 38 45 40 7 13 9 15 17 22 19 24 21 26 27 33 29 35 37 42 39 44 41 46
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 32 2 4 6 11 12 47 48 31
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
Y 17 37
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 17 37
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 4 8 6 10 13 18 15 20 24 28 26 30 33 38 35 40 44 47 46 48 7 3 9 5 17 12 19 14 21 16 27 23 29 25 37 32 39 34 41 36
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 4 8 6 10 13 18 15 20 24 28 26 30 33 38 35 40 44 47 46 48 7 3 9 5 17 12 19 14 21 16 27 23 29 25 37 32 39 34 41 36
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 0 1 2 42 11 43 45 22 31
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 7 9 17 19 21 27 29 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 7 9 17 19 21 27 29 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48

```
PAULI_CHANNEL_1(0.000128492, 0.000128492, 0.000642294) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000642294, 0.000642294, 0.00128418) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
X_ERROR(0.005) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
M 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
DETECTOR rec[-22]
DETECTOR rec[-20]
DETECTOR rec[-18]
DETECTOR rec[-17]
DETECTOR rec[-15]
DETECTOR rec[-13]
DETECTOR rec[-12]
DETECTOR rec[-10]
DETECTOR rec[-8]
DETECTOR rec[-7]
DETECTOR rec[-5]
DETECTOR rec[-3]
TICK
TICK
PAULI_CHANNEL_1(0.000128492, 0.000128492, 0.000642294) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000642294, 0.000642294, 0.00128418) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
R 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
X_ERROR(0.003) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 2 0 4 1 13 8 15 10 22 18 24 20 33 28 35 30 42 38 44 40 7 12 9 14 11 16 19 23 21 25 27 32 29 34 31 36 39 43 41 45
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 2 0 4 1 13 8 15 10 22 18 24 20 33 28 35 30 42 38 44 40 7 12 9 14 11 16 19 23 21 25 27 32 29 34 31 36 39 43 41 45
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 3 5 6 37 46 47 48 17 26
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
```

SQRT_Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
Y 0 1
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 0 1
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 3 0 5 1 14 8 16 10 23 18 25 20 34 28 36 30 43 38 45 40 7 13 9 15 17 22 19 24 21 26 27 33 29 35 37 42 39 44 41 46
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 3 0 5 1 14 8 16 10 23 18 25 20 34 28 36 30 43 38 45 40 7 13 9 15 17 22 19 24 21 26 27 33 29 35 37 42 39 44 41 46
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 32 2 4 6 11 12 47 48 31
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 0 1
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 0 1
Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
SQRT_Y 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 3 8 5 10 12 18 14 20 23 28 25 30 32 38 34 40 43 47 45 48 7 2 9 4 11 6 19 13 21 15 27 22 29 24 31 26 39 33 41 35
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 3 8 5 10 12 18 14 20 23 28 25 30 32 38 34 40 43 47 45 48 7 2 9 4 11 6 19 13 21 15 27 22 29 24 31 26 39 33 41 35
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 0 1 36 37 42 44 46 16 17
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
Y 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32

```
33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 4 8 6 10 13 18 15 20 24 28 26 30 33 38 35 40 44 47 46 48 7 3 9 5 17 12 19 14 21 16 27 23 29 25 37 32 39 34 41 36
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 4 8 6 10 13 18 15 20 24 28 26 30 33 38 35 40 44 47 46 48 7 3 9 5 17 12 19 14 21 16 27 23 29 25 37 32 39 34 41 36
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 0 1 2 42 11 43 45 22 31
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 7 9 11 17 19 21 27 29 31 37 39 41 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 7 9 11 17 19 21 27 29 31 37 39 41 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.000128492, 0.000128492, 0.000642294) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000642294, 0.000642294, 0.00128418) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
X_ERROR(0.005) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
M 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
DETECTOR rec[-24] rec[-48]
DETECTOR rec[-23] rec[-47]
DETECTOR rec[-22] rec[-46]
DETECTOR rec[-21] rec[-45]
DETECTOR rec[-20] rec[-44]
DETECTOR rec[-19] rec[-43]
DETECTOR rec[-18] rec[-42]
DETECTOR rec[-17] rec[-41]
DETECTOR rec[-16] rec[-40]
DETECTOR rec[-15] rec[-39]
DETECTOR rec[-14] rec[-38]
DETECTOR rec[-13] rec[-37]
DETECTOR rec[-12] rec[-36]
DETECTOR rec[-11] rec[-35]
DETECTOR rec[-10] rec[-34]
DETECTOR rec[-9] rec[-33]
DETECTOR rec[-8] rec[-32]
DETECTOR rec[-7] rec[-31]
DETECTOR rec[-6] rec[-30]
DETECTOR rec[-5] rec[-29]
DETECTOR rec[-4] rec[-28]
DETECTOR rec[-3] rec[-27]
DETECTOR rec[-2] rec[-26]
```

```
DETECTOR rec[-1] rec[-25]
TICK
TICK
PAULI_CHANNEL_1(0.000128492, 0.000128492, 0.000642294) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000642294, 0.000642294, 0.00128418) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
R 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
X_ERROR(0.003) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 7 9 11 17 19 21 27 29 31 37 39 41 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 7 9 11 17 19 21 27 29 31 37 39 41 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 4 8 6 10 13 18 15 20 24 28 26 30 33 38 35 40 44 47 46 48 7 3 9 5 17 12 19 14 21 16 27 23 29 25 37 32 39 34 41 36
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 4 8 6 10 13 18 15 20 24 28 26 30 33 38 35 40 44 47 46 48 7 3 9 5 17 12 19 14 21 16 27 23 29 25 37 32 39 34 41 36
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 0 1 2 42 11 43 45 22 31
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
Y 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 3 8 5 10 12 18 14 20 23 28 25 30 32 38 34 40 43 47 45 48 7 2 9 4 11 6 19 13 21 15 27 22 29 24 31 26 39 33 41 35
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 3 8 5 10 12 18 14 20 23 28 25 30 32 38 34 40 43 47 45 48 7 2 9 4 11 6 19 13 21 15 27 22 29 24 31 26 39 33 41 35
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 0 1 36 37 42 44 46 16 17
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27
```

28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 47 48
Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
SQRT_Y 0 1
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 0 1
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 3 0 5 1 14 8 16 10 23 18 25 20 34 28 36 30 43 38 45 40 7 13 9 15 17 22 19 24 21 26 27 33 29 35 37 42 39 44 41 46
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 3 0 5 1 14 8 16 10 23 18 25 20 34 28 36 30 43 38 45 40 7 13 9 15 17 22 19 24 21 26 27 33 29 35 37 42 39 44 41 46
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 32 2 4 6 11 12 47 48 31
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
Y 0 1
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 0 1
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 2 0 4 1 13 8 15 10 22 18 24 20 33 28 35 30 42 38 44 40 7 12 9 14 11 16 19 23 21 25 27 32 29 34 31 36 39 43 41 45
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 2 0 4 1 13 8 15 10 22 18 24 20 33 28 35 30 42 38 44 40 7 12 9 14 11 16 19 23 21 25 27 32 29 34 31 36 39 43 41 45
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 3 5 6 37 46 47 48 17 26
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40
PAULI_CHANNEL_1(0.000128492, 0.000128492, 0.000642294) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000642294, 0.000642294, 0.00128418) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30

```
31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
X_ERROR(0.005) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
M 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
DETECTOR rec[-24] rec[-48]
DETECTOR rec[-23] rec[-47]
DETECTOR rec[-22] rec[-46]
DETECTOR rec[-21] rec[-45]
DETECTOR rec[-20] rec[-44]
DETECTOR rec[-19] rec[-43]
DETECTOR rec[-18] rec[-42]
DETECTOR rec[-17] rec[-41]
DETECTOR rec[-16] rec[-40]
DETECTOR rec[-15] rec[-39]
DETECTOR rec[-14] rec[-38]
DETECTOR rec[-13] rec[-37]
DETECTOR rec[-12] rec[-36]
DETECTOR rec[-11] rec[-35]
DETECTOR rec[-10] rec[-34]
DETECTOR rec[-9] rec[-33]
DETECTOR rec[-8] rec[-32]
DETECTOR rec[-7] rec[-31]
DETECTOR rec[-6] rec[-30]
DETECTOR rec[-5] rec[-29]
DETECTOR rec[-4] rec[-28]
DETECTOR rec[-3] rec[-27]
DETECTOR rec[-2] rec[-26]
DETECTOR rec[-1] rec[-25]
TICK
TICK
PAULI_CHANNEL_1(0.000128492, 0.000128492, 0.000642294) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000642294, 0.000642294, 0.00128418) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
R 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
X_ERROR(0.003) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 7 9 17 19 21 27 29 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 7 9 17 19 21 27 29 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 4 8 6 10 13 18 15 20 24 28 26 30 33 38 35 40 44 47 46 48 7 3 9 5 17 12 19 14 21 16 27 23 29 25 37 32 39 34 41 36
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 4 8 6 10 13 18 15 20 24 28 26 30 33 38 35 40 44 47 46 48 7 3 9 5 17 12 19 14 21 16 27 23 29 25 37 32 39 34 41 36
```

PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 0 1 2 42 11 43 45 22 31
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
Y 17 37
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 17 37
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 3 0 5 1 14 8 16 10 23 18 25 20 34 28 36 30 43 38 45 40 7 13 9 15 17 22 19 24 21 26 27 33 29 35 37 42 39 44 41 46
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 3 0 5 1 14 8 16 10 23 18 25 20 34 28 36 30 43 38 45 40 7 13 9 15 17 22 19 24 21 26 27 33 29 35 37 42 39 44 41 46
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 32 2 4 6 11 12 47 48 31
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 17 37
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 17 37
Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
SQRT_Y 11 31
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 11 31
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 3 8 5 10 12 18 14 20 23 28 25 30 32 38 34 40 43 47 45 48 7 2 9 4 11 6 19 13 21 15 27 22 29 24 31 26 39 33 41 35
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 3 8 5 10 12 18 14 20 23 28 25 30 32 38 34 40 43 47 45 48 7 2 9 4 11 6 19 13 21 15 27 22 29 24 31 26 39 33 41 35
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 0 1 36 37 42 44 46 16 17
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30

```
31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
Y 11 31
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 11 31
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
CZ 2 0 4 1 13 8 15 10 22 18 24 20 33 28 35 30 42 38 44 40 7 12 9 14 11 16 19 23 21 25 27 32 29 34 31 36 39 43 41 45
PAULI_CHANNEL_2(5e-05, 5e-05, 0.0005, 5e-05, 0, 0, 0, 5e-05, 0, 0, 0, 0.0005, 0, 0, 0.001) 2 0 4 1 13 8 15 10 22 18 24 20 33 28 35 30 42 38 44 40 7 12 9 14 11 16 19 23 21 25 27 32 29 34 31 36 39 43 41 45
PAULI_CHANNEL_1(5e-05, 5e-05, 0.001) 3 5 6 37 46 47 48 17 26
I 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
PAULI_CHANNEL_1(2.34997e-05, 2.34997e-05, 0.000117493) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000117493, 0.000117493, 0.000234973) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
SQRT_Y 7 9 11 19 21 27 29 31 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 7 9 11 19 21 27 29 31 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.000128492, 0.000128492, 0.000642294) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
PAULI_CHANNEL_1(0.000642294, 0.000642294, 0.00128418) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
Y 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46 7 9 11 17 19 21 27 29 31 37 39 41 0 1 8 10 18 20 28 30 38 40 47 48
X_ERROR(0.005) 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
I 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
M 0 1 7 8 9 10 11 17 18 19 20 21 27 28 29 30 31 37 38 39 40 41 47 48
DETECTOR rec[-24] rec[-48]
DETECTOR rec[-23] rec[-47]
DETECTOR rec[-22] rec[-46]
DETECTOR rec[-21] rec[-45]
DETECTOR rec[-20] rec[-44]
DETECTOR rec[-19] rec[-43]
DETECTOR rec[-18] rec[-42]
DETECTOR rec[-17] rec[-41]
DETECTOR rec[-16] rec[-40]
DETECTOR rec[-15] rec[-39]
DETECTOR rec[-14] rec[-38]
DETECTOR rec[-13] rec[-37]
DETECTOR rec[-12] rec[-36]
DETECTOR rec[-11] rec[-35]
DETECTOR rec[-10] rec[-34]
```

```
DETECTOR rec[-9] rec[-33]
DETECTOR rec[-8] rec[-32]
DETECTOR rec[-7] rec[-31]
DETECTOR rec[-6] rec[-30]
DETECTOR rec[-5] rec[-29]
DETECTOR rec[-4] rec[-28]
DETECTOR rec[-3] rec[-27]
DETECTOR rec[-2] rec[-26]
DETECTOR rec[-1] rec[-25]
TICK
SQRT_Y_DAG 2 4 6 13 15 22 24 26 33 35 42 44 46
PAULI_CHANNEL_1(0.0002, 0.0006, 0.0002) 2 4 6 13 15 22 24 26 33 35 42 44 46
X_ERROR(0.005) 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
I 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
M 2 3 4 5 6 12 13 14 15 16 22 23 24 25 26 32 33 34 35 36 42 43 44 45 46
DETECTOR rec[-47] rec[-20] rec[-19] rec[-25] rec[-24]
DETECTOR rec[-45] rec[-18] rec[-17] rec[-23] rec[-22]
DETECTOR rec[-43] rec[-16] rec[-21]
DETECTOR rec[-42] rec[-15] rec[-20]
DETECTOR rec[-40] rec[-14] rec[-13] rec[-19] rec[-18]
DETECTOR rec[-38] rec[-12] rec[-11] rec[-17] rec[-16]
DETECTOR rec[-37] rec[-10] rec[-9] rec[-15] rec[-14]
DETECTOR rec[-35] rec[-8] rec[-7] rec[-13] rec[-12]
DETECTOR rec[-33] rec[-6] rec[-11]
DETECTOR rec[-32] rec[-5] rec[-10]
DETECTOR rec[-30] rec[-4] rec[-3] rec[-9] rec[-8]
DETECTOR rec[-28] rec[-2] rec[-1] rec[-7] rec[-6]
OBSERVABLE_INCLUDE(0) rec[-15] rec[-14] rec[-13] rec[-12] rec[-11]
```

Moving AWG command string

The command string used to program the RF waveforms for the Moving AWG, which control the positions and powers of moveable AOD tweezers, is given below. Waveforms are uploaded to different segments of the AWG using the SEGMENT[] command, which can then be individually looped or triggered with an external trigger.
Each move is defined using a MOVEAOD command. The initial position and power is specified for the first command when tweezers are turned on. For subsequent commands, the arguments specify where to move to (and optionally the power to ramp to), and take the form MOVEAOD[[x_positions, y_positions],([xpowers,ypowers],) time_for_move]. Additional arguments such as SHIFT apply a global (x,y) coordinate shift to the coordinates in that move. Coordinates are given in 'SLM units' where 1 SLM unit here is approximately 4.6um in the x direction and 4.9um in the y direction (corresponding to 7 and 7.5 units in the corresponding Supplementary Video, respectively)

```
### Define coordinates and powers

COMB[xd_ent, [13,3,6]]
COMB[yd_ent, [2,2,6]]
COMB[ykill_ent, [0, 2, 7]]
REPEAT_VAL[xkill_pow, [0.4, 6]]
REPEAT_VAL[ykill_pow, [1.65, 7]]

COMB[xanc_ent, [14.86,3,6]]
COMB[yanc_ent, [1,2,6]]

REPEAT_VAL[xno6_pow, [0.14, 6]]
REPEAT_VAL[yno6_pow, [0.0001, 6]]
REPEAT_VAL[yno7_pow, [0.0001, 7]]

REPEAT_VAL[xpow_imaging, [0.25, 6]]
REPEAT_VAL[ypow_imaging, [1.01, 6]]

REPEAT_VAL[xpow_gates, [0.18, 6]]
REPEAT_VAL[ypow_gates, [0.91, 6]]

COMB[x_st, [-0.98, 3, 6]]
COMB[y_st, [21, 1.98, 6]]

REPEAT_VAL[xpow_pickup, [0.36, 6]]
REPEAT_VAL[ypow_pickup, [1.83, 6]]
REPEAT_VAL[x2pow_pickup, [0.36, 6]]
REPEAT_VAL[y2pow_pickup, [1.33, 6]]

COMB[x_ssr, [-0.98,2,18]]

VAR[xpow1_ssr,
[0.26,0.26,0.24,0.24,0.22,0.21,0.21,0.21,0.21,0.21,0.21,0.23,0.23,0.23,0.24,0.2
4,0.24,0.20]]
REPEAT_VAL[xno18_pow, [0.0001,18]]
VAR[y3pow_pickup, [1.19, 1.20, 1.23, 1.28, 1.29, 1.47]]


### First segment, used to clear unwanted atoms from entangling zone traps
```

```
SEGMENT[1, LOOP_UNTIL_TRIG]
MoveAOD[Initial[xd_ent, ykill_ent],
InitialPows[xkill_pow, yno7_pow],
FinalPows[xkill_pow,ykill_pow], 0.1 ms]
MoveAOD[[xd_ent, ykill_ent], SHIFT[[-1.5],[-1]], 0.1 ms]
WAIT[0.02 ms]

### Second segment, move atoms to their positions for the start of the circuit

SEGMENT[2]
MoveAOD[Initial[x_st, y_st],
InitialPows[xno6_pow, xno6_pow],
FinalPows[xpow_pickup,ypow_pickup],0.2 ms]
SAVE_PREVIOUS[]

NOTE[Move data qubits to the entangling zone]
MoveAOD[[x_st,y_st],SHIFT[[0.5],[0]],0.1 ms]
MoveAOD[[x_st,yd_ent],SHIFT[[0.5],[-0.5]],0.4 ms]
MoveAOD[[xd_ent,yd_ent],SHIFT[[0],[-0.5]],0.2 ms]
MoveAOD[[xd_ent,yd_ent],SHIFT[[0],[0]],0.1 ms]
MoveAOD[[xd_ent,yd_ent],FinalPows[xno6_pow,xno6_pow],0.3 ms]

NOTE[Move first ancilla set to entangling zone]
MoveAOD[Initial[x_st,y_st],SHIFT[[1],[0]],
InitialPows[xno6_pow, xno6_pow],
FinalPows[xpow_pickup, ypow_pickup],0.2 ms]

MoveAOD[[x_st,y_st],SHIFT[[1.5],[0]],0.1 ms]
MoveAOD[[x_st,yanc_ent],SHIFT[[1.5],[0]],0.3 ms]
MoveAOD[[xanc_ent,yanc_ent], 0.3 ms]
MoveAOD[[xanc_ent,yanc_ent], FinalPows[xpow_imaging, ypow_imaging],0.05ms]
SAVE_PREVIOUS[]
SAVE_PREVIOUS[]

### Start of circuit

SEGMENT[4]

Wait[1 us]
MoveAOD[[xanc_ent,yanc_ent], FinalPows[xpow_gates, ypow_gates], 0.1 ms]

NOTE[Data pi/2 pulses for first round]
Wait[403 us]
NOTE[Fake ancilla pi/2 pulses, first round]
Wait[70 us]
NOTE[Data qubit pi pulse, followed by Ancilla pi/2 pulses]

Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[-1.5],[1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[-1.5],[-1]], 0.195 ms]
```

```
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0.75],[1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0.78],[-1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]

NOTE[Move back to original spot]

MoveAOD[[xanc_ent,yanc_ent], FinalPows[x2pow_pickup, y2pow_pickup], 0.1 ms]
MoveAOD[[x_st,yanc_ent],SHIFT[[1.53],[0]],0.4 ms]
MoveAOD[[x_st,y_st],SHIFT[[1.53],[0]],0.4 ms]
MoveAOD[[x_st,y_st], SHIFT[[1.03],[0]],0.1 ms]
MoveAOD[[x_st,y_st], FinalPows[xno6_pow, yno6_pow], SHIFT[[1.03],[0]], 0.250
ms]

NOTE[Pick up next group of atoms, and do again. Exchanging the ancilla arrays
is currently given 1 ms total time.]

MoveAOD[Initial[x_st,y_st], InitialPows[xno6_pow, yno6_pow],
FinalPows[x2pow_pickup, y2pow_pickup],SHIFT[[1.938],[0]], 0.250 ms]
MoveAOD[[x_st,y_st],SHIFT[[2.438],[0]],0.1 ms]
MoveAOD[[x_st,yanc_ent],SHIFT[[2.438],[0]],0.4 ms]
MoveAOD[[xanc_ent,yanc_ent], 0.4 ms]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], FinalPows[xpow_gates, ypow_gates],
0.1 ms]

Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[-1.5],[1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0.78],[1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[-1.5],[-1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0.78],[-1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]

MoveAOD[[xanc_ent,yanc_ent], FinalPows[x2pow_pickup, y2pow_pickup], 0.1 ms]
MoveAOD[[x_st,yanc_ent],SHIFT[[2.438],[0]],0.4 ms]
MoveAOD[[x_st,y_st],SHIFT[[2.438],[0]],0.4 ms]
MoveAOD[[x_st,y_st], SHIFT[[1.938],[0]],0.1 ms]
```

```
MoveAOD[[x_st,y_st], FinalPows[xno6_pow, yno6_pow], SHIFT[[1.938],[0]], 0.250
ms]

Note[Third ancilla block]

MoveAOD[Initial[x_st,y_st], InitialPows[xno6_pow, yno6_pow],
FinalPows[x2pow_pickup, y2pow_pickup],SHIFT[[17.959],[0]], 0.250 ms]
MoveAOD[[x_st,y_st],SHIFT[[18.459],[0]],0.1 ms]
MoveAOD[[x_st,yanc_ent],SHIFT[[18.459],[0]],0.4 ms]
MoveAOD[[xanc_ent,yanc_ent], 0.4 ms]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], FinalPows[xpow_gates, ypow_gates],
0.1 ms]

Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0.78],[-1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[-1.5],[-1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0.78],[1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[-1.5],[1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]

MoveAOD[[xanc_ent,yanc_ent], FinalPows[x2pow_pickup, y2pow_pickup], 0.1 ms]
MoveAOD[[x_st,yanc_ent],SHIFT[[18.459],[0]],0.4 ms]
MoveAOD[[x_st,y_st],SHIFT[[18.459],[0]],0.4 ms]
MoveAOD[[x_st,y_st], SHIFT[[17.959],[0]],0.1 ms]
MoveAOD[[x_st,y_st], FinalPows[xno6_pow, yno6_pow], SHIFT[[17.959],[0]], 0.250
ms]

Note[Fourth ancilla block]

MoveAOD[Initial[x_st,y_st], InitialPows[xno6_pow, yno6_pow],
FinalPows[x2pow_pickup, y2pow_pickup],SHIFT[[18.972],[0]], 0.250 ms]
MoveAOD[[x_st,y_st],SHIFT[[19.472],[0]],0.1 ms]
MoveAOD[[x_st,yanc_ent],SHIFT[[19.472],[0]],0.4 ms]
MoveAOD[[xanc_ent,yanc_ent], 0.4 ms]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], FinalPows[xpow_gates, ypow_gates],
0.1 ms]

Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0.78],[-1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0.78],[1]], 0.195 ms]
```

```
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[-1.5],[-1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[-1.5],[1]], 0.195 ms]
Wait[10 us]
MoveAOD[[xanc_ent,yanc_ent], SHIFT[[0],[0]], 0.195 ms]
Wait[70 us]

MoveAOD[[xanc_ent,yanc_ent], FinalPows[x2pow_pickup, y2pow_pickup], 0.1 ms]
MoveAOD[[x_st,yanc_ent],SHIFT[[19.472],[0]],0.4 ms]
MoveAOD[[x_st,y_st],SHIFT[[19.472],[0]],0.4 ms]
MoveAOD[[x_st,y_st], SHIFT[[18.972],[0]],0.1 ms]
MoveAOD[[x_st,y_st], FinalPows[xno6_pow, yno6_pow], SHIFT[[18.972],[0]], 0.250
ms]

Note[Move data back to storage zone, 1.2 ms]

MoveAOD[Initial[xd_ent, yd_ent], InitialPows[xno6_pow, yno6_pow],
FinalPows[x2pow_pickup,y2pow_pickup], 0.35 ms]
MoveAOD[[xd_ent,yd_ent],SHIFT[[0],[-0.5]],0.2 ms]
MoveAOD[[x_st,yd_ent], SHIFT[[0.53],[-0.5]],0.3 ms]
MoveAOD[[x_st,y_st], SHIFT[[0.53],[0]],0.5 ms]
MoveAOD[[x_st,y_st], SHIFT[[0.03],[0]],0.2 ms]
MoveAOD[[x_st,y_st], FinalPows[xno6_pow, yno6_pow], SHIFT[[0.03],[0]], 0.3 ms]


SEGMENT[6]
Note[SSR on half, 1.2 ms]

WAIT[0.2 ms]

MoveAOD[Initial[x_ssr, y_st],
InitialPows[xno18_pow, yno6_pow],
FinalPows[xpow1_ssr,y3pow_pickup],0.5 ms]
WAIT[100 us]
MoveAOD[[x_ssr, y_st], SHIFT[[0.5],[0]], 0.7 ms]
MoveAOD[[x_ssr, y_st], SHIFT[[0.5],[-1]],0.2 ms]
MoveAOD[[x_ssr, y_st], SHIFT[[0],[-1]],0.1 ms]
MoveAOD[[x_ssr,y_st], FinalPows[xno18_pow, yno6_pow], SHIFT[[0],[-1]], 0.3 ms]

Note[SSR on half, 1.23 ms]

MoveAOD[Initial[x_ssr, y_st],
InitialPows[xno18_pow, yno6_pow],
FinalPows[xpow1_ssr,y3pow_pickup], SHIFT[[1],[0]],0.5 ms]
WAIT[100 us]
MoveAOD[[x_ssr, y_st], SHIFT[[1.5],[0]], 0.7 ms]
MoveAOD[[x_ssr, y_st], SHIFT[[1.5],[-1]],0.2 ms]
MoveAOD[[x_ssr, y_st], SHIFT[[1],[-1]],0.1 ms]
MoveAOD[[x_ssr,y_st], FinalPows[xno18_pow, yno6_pow], SHIFT[[1],[-1]], 0.3 ms]
```

```
END_COMMAND[]
```

Raman AWG command string

The command string used to program the I/Q waveforms and AOM amplitudes for local and global Raman control is given below. The command specifies the parameters for each single-qubit gate, which is then converted into the appropriate waveforms to modulate the light intensity and microwave phase and frequency (defined relative to the resonance frequency).
For local gates, the pulses are optionally gated on and off by the Raman AOD command.
Examples of typical commands are
Z : global Z rotation, in units of 2*pi
E : wait time, idling at the specified frequency
SC1_ECHO : global SCROFULOUS Y(pi) pulse for dynamical decoupling, idling at the specified frequency
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP : SCROFULOUS Y(pi/2) local gaussian pulses
SC1PI_FIXEDLOOPED_SMOOTH_PCOMP : SCROFULOUS Y(pi) local gaussian pulses
BB1_FIXED: global BB1 pulse

```
### State preparation before circuit begins

E[50.5 us, 12 MHz]
CPROBUST180_OFFSETFREQSMOOTH_COMPENSATED[1.73 us, 0.035 us, 12.1056 MHz, 0.75, 4]
E[0.2 us, 8.3 MHz]
CPROBUST180_OFFSETFREQSMOOTH[2.455 us, 0.035 us, 4.063 MHz, 1, 4]
E[2.65 us, 0.1 MHz]
SEGMENTBREAK[1896]
E[16.6 us, 0.1 MHz]
BB1_FIXED[0.5,0.12 us,5.65 us]
E[4 us, 5 MHz]
SMOOTHPUMPINGLOOPEDTTL[0, 2.18 us, -11.7 MHz, 11.7 MHz, 1, 4, 24, 8.53 us, 0.1us, 3us]
E[494.052 us, 5 MHz]

### Start of circuit

# Prepare data qubit
SC1_ECHO[6 us, 0 MHz]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
SC1PI_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.45710, 6, local]
Z[0.45710]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]

# Start first QEC round

Z[0]
E[397.968 us, 0.07 MHz]
Z[0.000]

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]
SC1_ECHO[76 us, 0.07 MHz]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]

Z[0.287]
```

```
E[397.968 us, 0.07 MHz]
Z[0.000]

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
Z[0.46744]
SC1PI_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.45710, 6, local]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
Z[0.46744]

Z[0.000]
Z[0.287]
E[397.968 us, 0.07 MHz]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]
SC1_ECHO[4 us, 0.07 MHz]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]
E[397.968 us, 0.07 MHz]
Z[0.287]
Z[0.000]

Z[0.46744]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
SC1PI_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.45710, 6, local]
Z[0.46744]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]

Z[0.287]
E[397.968 us, 0.07 MHz]
Z[0.000]

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]

Z[0.000]
SC1_ECHO[2574.028 us, 0.07 MHz]
Z[0.000]

# Start second QEC round

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]

Z[0.287]
E[397.968 us, 0.07 MHz]
Z[0.000]

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
Z[0.46744]
SC1PI_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.45710, 6, local]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
Z[0.46744]

Z[0.000]
Z[0.287]
E[397.968 us, 0.07 MHz]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]
SC1_ECHO[4 us, 0.07 MHz]
```

```
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]
E[397.968 us, 0.07 MHz]
Z[0.287]
Z[0.000]

Z[0.46744]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
SC1PI_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.45710, 6, local]
Z[0.46744]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]

Z[0.287]
E[397.968 us, 0.07 MHz]
Z[0.000]

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]

Z[0.000]
SC1_ECHO[2574.028 us, 0.07 MHz]
Z[0.000]


# Start third QEC round


SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]

Z[0.287]
E[397.968 us, 0.07 MHz]
Z[0.000]

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
Z[0.46744]
SC1PI_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.45710, 6, local]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
Z[0.46744]

Z[0.000]
Z[0.287]
E[397.968 us, 0.07 MHz]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]
SC1_ECHO[4 us, 0.07 MHz]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]
E[397.968 us, 0.07 MHz]
Z[0.287]
Z[0.000]

Z[0.46744]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
SC1PI_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.45710, 6, local]
Z[0.46744]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]

Z[0.287]
```

```
E[397.968 us, 0.07 MHz]
Z[0.000]

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]

Z[0.000]
SC1_ECHO[2574.028 us, 0.07 MHz]
Z[0.000]


# Start fourth QEC round


SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]

Z[0.287]
E[397.968 us, 0.07 MHz]
Z[0.000]

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
Z[0.46744]
SC1PI_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.45710, 6, local]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
Z[0.46744]

Z[0.000]
Z[0.287]
E[397.968 us, 0.07 MHz]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]
SC1_ECHO[4 us, 0.07 MHz]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]
E[397.968 us, 0.07 MHz]
Z[0.287]
Z[0.000]

Z[0.46744]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
SC1PI_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.45710, 6, local]
Z[0.46744]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]

Z[0.287]
E[397.968 us, 0.07 MHz]
Z[0.000]

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]


SC1_ECHO[76 us, 0.07 MHz]

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 6, local]
Z[0]
E[397.968 us, 0.07 MHz]
Z[0.000]
```

```
# Readout data qubits

SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]
SC1PI_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.45710, 6, local]
Z[0.46744]
SC1_H_FIXEDLOOPED_SMOOTH_PCOMP[0.01us, 0.88, 4, 5.8180 us, 0.46744, 3, local]

# Bright state transfer for spin-to-position conversion

E[2.872 ms, 5 MHz]

BB1_FIXED[0.5,0.1us,6us]

E[1.819ms, 5MHz]
CPROBUST180_OFFSETFREQSMOOTH[4.55 us, 0.035 us, 4.083 MHz, 1, 4]
E[0.2 us, 8.3 MHz]
CPROBUST180_OFFSETFREQSMOOTH_COMPENSATED[3.1 us, 0.035 us, 12.3256 MHz, 0.75,
4]


END_COMMAND[]
```

Raman AOD AWG command string

The command string used to program the RF waveforms for the Raman AOD AWG, which controls the positions and powers of 2D grids of optical tweezers used for local single-qubit gates, is given below.
Positions and powers are separately calibrated and loaded from a saved file. Different pulse configurations are specified by variables, for example in this circuit:
r0_Ad --> row 0, sublattice A of data qubits
r0_a --> row 0, ancilla qubits
r2_nl_a --> row 2, no left edge, ancilla qubits
r1_r_a --> row 1, right edge only, ancilla qubits

```
FILE[repetitiveQEC_d5_variables.txt] # Coordinates are loaded from file
VAR[time[], WAIT[5.8233us]]
VAR[turnoff[], MOVEAOD[Initial[[-3],[-3]],INITIALPOWS[[0],[0]],10ns]]
WAIT[57.47 us]

### Start of circuit

# Prepare data qubit in logical state

r0_Ad[]
time[]
r1_Ad[]
time[]
r2_Ad[]
time[]
turnoff[]
WAIT[34.99us]
r3_Ad[]
time[]
r4_Ad[]
time[]
r5_Ad[]
time[]
turnoff[]


# Start first QEC round

WAIT[398.02us]

WAIT[70us]
WAIT[6us]
WAIT[35us]
r0_a[]
time[]
r1_a[]
time[]
r2_nl_a[]
time[]
r3_a[]
time[]
r4_nl_a[]
```

```
time[]
r5_a[]
time[]

turnoff[]
WAIT[398.02us]

r5_d[]
time[]
r4_d[]
time[]
r3_d[]
time[]
turnoff[]
time[]
r1_r_a[]
time[]
turnoff[]
time[]
r3_r_a[]
time[]
turnoff[]
time[]
turnoff[]
time[]
r2_d[]
time[]
r1_d[]
time[]
r0_d[]
time[]

turnoff[]
WAIT[397.99us]
turnoff[]
time[]
r1_r_a[]
time[]
turnoff[]
time[]
r3_r_a[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
WAIT[3.99us]
turnoff[]
time[]
r4_l_a[]
time[]
turnoff[]
time[]
```

```
r2_l_a[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
WAIT[397.99us]

r0_d[]
time[]
r1_d[]
time[]
r2_d[]
time[]
turnoff[]
time[]
r4_l_a[]
time[]
turnoff[]
time[]
r2_l_a[]
time[]
turnoff[]
time[]
turnoff[]
time[]
r3_d[]
time[]
r4_d[]
time[]
r5_d[]
time[]

turnoff[]
WAIT[398.02us]
r5_a[]
time[]
r4_a[]
time[]
r3_nr_a[]
time[]
r2_a[]
time[]
r1_nr_a[]
time[]
r0_a[]
time[]
turnoff[]
WAIT[34.99us]

turnoff[]
WAIT[2503.99us]
```

```
# Start second QEC round


WAIT[35us]
r0_a[]
time[]
r1_a[]
time[]
r2_a[]
time[]
r3_a[]
time[]
r4_a[]
time[]
turnoff[]
time[]

turnoff[]
WAIT[398.02us]

r5_d[]
time[]
r4_d[]
time[]
r3_d[]
time[]
r0_a[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
r2_d[]
time[]
r1_d[]
time[]
r0_d[]
time[]

turnoff[]
WAIT[397.99us]
r0_a[]
time[]
turnoff[]
time[]
turnoff[]
time[]
```

```
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
WAIT[3.99us]
r5_a[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
WAIT[397.99us]

r0_d[]
time[]
r1_d[]
time[]
r2_d[]
time[]
r5_a[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
r3_d[]
time[]
r4_d[]
time[]
r5_d[]
time[]

turnoff[]
WAIT[398.02us]
r5_a[]
time[]
r4_a[]
time[]
```

```
r3_a[]
time[]
r2_a[]
time[]
r1_a[]
time[]
turnoff[]
time[]
turnoff[]
WAIT[34.99us]

turnoff[]
WAIT[2503.99us]


# Start third QEC round


WAIT[35us]
turnoff[]
time[]
r1_a[]
time[]
r2_a[]
time[]
r3_a[]
time[]
r4_a[]
time[]
r5_a[]
time[]

turnoff[]
WAIT[398.02us]

r5_d[]
time[]
r4_d[]
time[]
r3_d[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
r5_a[]
time[]
r2_d[]
```

```
time[]
r1_d[]
time[]
r0_d[]
time[]

turnoff[]
WAIT[397.99us]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
r5_a[]
time[]
turnoff[]
WAIT[3.99us]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
r0_a[]
time[]
turnoff[]
WAIT[397.99us]

r0_d[]
time[]
r1_d[]
time[]
r2_d[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
turnoff[]
time[]
r0_a[]
```

```
time[]
r3_d[]
time[]
r4_d[]
time[]
r5_d[]
time[]

turnoff[]
WAIT[398.02us]
turnoff[]
time[]
r4_a[]
time[]
r3_a[]
time[]
r2_a[]
time[]
r1_a[]
time[]
r0_a[]
time[]
turnoff[]
WAIT[34.99us]

turnoff[]
WAIT[2503.99us]


# Start fourth QEC round


WAIT[35us]
r0_a[]
time[]
r1_nr_a[]
time[]
r2_a[]
time[]
r3_nr_a[]
time[]
r4_a[]
time[]
r5_a[]
time[]

turnoff[]
WAIT[398.02us]

r5_d[]
time[]
r4_d[]
time[]
r3_d[]
```

```
time[]
turnoff[]
time[]
turnoff[]
time[]
r2_l_a[]
time[]
turnoff[]
time[]
r4_l_a[]
time[]
turnoff[]
time[]
r2_d[]
time[]
r1_d[]
time[]
r0_d[]
time[]

turnoff[]
WAIT[397.99us]
turnoff[]
time[]
turnoff[]
time[]
r2_l_a[]
time[]
turnoff[]
time[]
r4_l_a[]
time[]
turnoff[]
time[]
turnoff[]
WAIT[3.99us]
turnoff[]
time[]
turnoff[]
time[]
r3_r_a[]
time[]
turnoff[]
time[]
r1_r_a[]
time[]
turnoff[]
time[]
turnoff[]
WAIT[397.99us]

r0_d[]
time[]
r1_d[]
```

```
time[]
r2_d[]
time[]
turnoff[]
time[]
turnoff[]
time[]
r3_r_a[]
time[]
turnoff[]
time[]
r1_r_a[]
time[]
turnoff[]
time[]
r3_d[]
time[]
r4_d[]
time[]
r5_d[]
time[]

turnoff[]
WAIT[398.02us]
r5_a[]
time[]
r4_nl_a[]
time[]
r3_a[]
time[]
r2_nl_a[]
time[]
r1_a[]
time[]
r0_a[]
time[]
turnoff[]
WAIT[34.99us]

turnoff[]
WAIT[5.99us]

WAIT[70us]

turnoff[]
E[397.968 us, 0.07 MHz]

# Readout data qubits

r5_Ad[]
time[]
r4_Ad[]
time[]
r3_Ad[]
```

```
time[]
turnoff[]
WAIT[34.99us]
r2_Ad[]
time[]
r1_Ad[]
time[]
r0_Ad[]
time[]

turnoff[]
WAIT[8us]

END_COMMAND[]
```

Rydberg AWG command string

The command string, used to program the amplitude, frequency, and phase for the 420-nm AOM for entangling gates and local SLM AOM for local detunings, is given below. Gate profiles (amplitude, detuning, and phase) are loaded from saved calibration files.

```
Wait[500.91 us]

### Start of circuit

# Start first QEC round


Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]
Wait[467.800 us]
Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Wait[2570.02 us]


# Start second QEC round


Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]
```

```
Wait[467.800 us]

Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Wait[2570.02 us]


# Start third QEC round


Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]
```

```
Wait[467.800 us]

Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Wait[2570.02 us]


# Start fourth QEC round


Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]

Pulse[420Amp[amplitudeFromFile,idx=1001,loops=1,delay=0.2],
420Freq[detuningFromFile,idx=1001],
420Phase[phaseFromFile,idx=1001, loops=1,delay=0.2],
localShiftAmp[RabiPulseLoop,time=0.40,loops=1,A=1,delay = 0.18],
localShiftFreq[fixedFrequency,detuning = 0.0]]

Wait[467.800 us]
```

Moving AWG command string

The command string used to program the RF waveforms for the Moving AWG, which control the positions and powers of moveable AOD tweezers, is given below. Waveforms are uploaded to different segments of the AWG using the SEGMENT[] command, which can then be individually looped or triggered with an external trigger.
Each move is defined using a MOVEAOD command. The initial position and power is specified for the first command when tweezers are turned on. For subsequent commands, the arguments specify where to move to (and optionally the power to ramp to), and take the form MOVEAOD[[x_positions, y_positions],([xpowers,ypowers],) time_for_move]. Additional arguments such as SHIFT apply a global (x,y) coordinate shift to the coordinates in that move. Coordinates are given in 'SLM units' where 1 SLM unit here is approximately 5.6um in the x direction and 4.3um in the y direction.

```
### Define coordinates and powers

VAR[RESET[], MoveAOD[Initial[[-2],[-2]], InitialPows[[0.0001], [0.0001]], FinalPows[[0.0001], [0.0001]], 1 us]]

COMB[xe, [-0.02, 2.0, 16]]
COMB[ye, [22.04, 2.0, 8]]

COMB[x1_e_alt, [-0.02, 4, 8]]
VAR[x2_e_alt, [-0.02, 1.98, 7.98, 9.98, 15.98, 17.98, 23.98, 25.98]]
COMB[y1_e_alt, [22.04, 4, 4]]
VAR[y2_e_alt, [22.06, 24.06, 30.06, 32.06]]
COMB[y_e_top, [22.06, 2, 4]]

COMB[x_store_SLM, [0.03, 2, 16]]
COMB[y_st, [48, 1, 8]]

COMB[y_SSR, [4, 2, 8]]
COMB[x_SSR, [0, 1, 32]]
COMB[y2_SSR, [12, 2, 4]]

REPEAT_VAL[x_clear_pow, [0.271, 16]]
REPEAT_VAL[y_clear_pow, [1.05, 8]]

REPEAT_VAL[x_image_pow, [0.2, 16]]
REPEAT_VAL[y_image_pow, [0.4, 8]]

VAR[xSSR_32pow, [0.195, 0.195, 0.195, 0.195, 0.195, 0.195, 0.195, 0.190, 0.190, 0.190, 0.190, 0.185, 0.185, 0.185, 0.185, 0.175, 0.175, 0.175, 0.175, 0.175, 0.175, 0.175, 0.175, 0.170,0.170, 0.170,0.170, 0.170,0.165, 0.165,0.165, 0.165]]
REPEAT_VAL[x0_32pow, [0.0, 32]]
REPEAT_VAL[x0_pow, [0.0, 16]]
REPEAT_VAL[x_init_8pow, [0.0, 8]]
REPEAT_VAL[x_pickup_pow, [0.221, 16]]
REPEAT_VAL[x_high_pow, [0.31, 16]]
REPEAT_VAL[x2_high_pow, [0.39, 16]]
REPEAT_VAL[x_high_8pow, [0.185, 8]]

REPEAT_VAL[x_gate_pow, [0.23, 16]]
```

```
REPEAT_VAL[x_gate_8pow, [0.095, 8]]


REPEAT_VAL[y0_pow, [0.0, 8]]
REPEAT_VAL[y0_4pow, [0.0, 4]]
VAR[y1_up_pow, [0.00001, 0.00002, 0.00003, 0.00004, 1.05, 1.05, 1.05, 1.05]]
REPEAT_VAL[y2_up_pow, [0.93, 8]]
VAR[y3_up_pow, [1.15, 1.12, 1.09, 1.05, 0.97, 0.93, 0.93, 0.93]]
REPEAT_VAL[y_4pow, [0.6,4]]
VAR[y2_high_4pow, [0.98,0.98,0.98,0.98]]
REPEAT_VAL[y_high_pow, [1.05, 8]]
VAR[ySSR_4pow, [1.50, 1.45, 1.35, 1.30]]
REPEAT_VAL[ylowSSR_4pow, [1.21, 4]]


### First segment, used to clear unwanted atoms from entangling zone traps

SEGMENT[1, LOOP_UNTIL_TRIG]
MoveAOD[Initial[xe, ye],
InitialPows[x0_pow, y0_pow],
FinalPows[x_clear_pow, y_clear_pow], 0.05 ms]
MoveAOD[[xe, ye], SHIFT[-1, -1], 0.05 ms]
MoveAOD[Initial[x_store_SLM, y_st],
InitialPows[x0_pow, y0_pow],
FinalPows[x_clear_pow, y_clear_pow], 0.05 ms]
MoveAOD[[x_store_SLM, y_st], SHIFT[-1, -1], 0.05 ms]
WAIT[0.01 ms]

SEGMENT[2]
WAIT[214 us]
RESET[]

### Move atoms to their positions for the start of the circuit

SEGMENT[4]
WAIT[10 us]
MoveAOD[Initial[xe, y_SSR],
InitialPows[x0_pow, y0_pow],
FinalPows[x_high_pow, y1_up_pow], SHIFT[0, 0], YPHASES[6.2401, 2.1642, 3.1614,
4.1102, 3.6525,5.2447, 3.1304, 1.1961], 0.15 ms]
MoveAOD[[xe, y_SSR], SHIFT[0.5, 0], 0.1 ms]
MoveAOD[[xe, y_SSR], SHIFT[0.5, 8], 0.3 ms]
MoveAOD[[xe, y_SSR], SHIFT[1, 8], 0.1 ms]
MoveAOD[[xe, y_SSR], FinalPows[x_high_pow, y2_up_pow], SHIFT[1, 8], 0.15 ms]
MoveAOD[[xe, y_SSR], SHIFT[0.5, 8], 0.1 ms]
MoveAOD[[xe, y_st], SHIFT[0.5, 0.0], 0.4 ms]
MoveAOD[[xe, y_st], SHIFT[0.0, 0.0], 0.1 ms]
MoveAOD[[xe, y_st], FinalPows[x0_pow, y0_pow], SHIFT[0.0, 0.0], 0.15 ms]
NOTE[Group A in storage zone]

SAVE_PREVIOUS[]
SAVE_PREVIOUS[]

WAIT[642.5 us]
```

```
RESET[]


### Start circuit here (looping single waveform per layer)


SEGMENT[5, LOOP_UNTIL_TRIG]
WAIT[4 us]
RESET[]

NOTE[Group B: Readout to entangling]
MoveAOD[Initial[xe, y_SSR],
InitialPows[x0_pow, y0_pow],
FinalPows[x_high_pow, y1_up_pow], SHIFT[0, 1], YPHASES[6.2401, 2.1642, 3.1614,
4.1102, 3.6525,5.2447, 3.1304, 1.1961], 0.15 ms]
MoveAOD[[xe, y_SSR], SHIFT[0.5, 1], 0.1 ms]
MoveAOD[[xe, y_SSR], SHIFT[0.5, 9], 0.25 ms]
MoveAOD[[xe, y_SSR], SHIFT[1, 9], 0.1 ms]
MoveAOD[[xe, y_SSR], FinalPows[x_high_pow, y2_up_pow], SHIFT[1, 9], 0.15 ms]
MoveAOD[[xe, y_SSR], SHIFT[0.5, 9], 0.1 ms]
MoveAOD[[xe, ye], SHIFT[0.5, 0], 0.25 ms]
MoveAOD[[xe, ye], SHIFT[0, 0], 0.1 ms]
MoveAOD[[xe, ye], FinalPows[x0_pow, y0_pow], SHIFT[0,0], 0.15 ms]

NOTE[Group B prep]

WAIT[100 us]

NOTE[First gate layer of state prep]
MoveAOD[Initial[x1_e_alt, ye],
InitialPows[x_init_8pow, y0_pow],
FinalPows[x_high_8pow, y_high_pow], SHIFT[2, 0], 0.15 ms]
MoveAOD[[x1_e_alt, ye], SHIFT[2, 0.5], 0.1 ms]
MoveAOD[[x1_e_alt, ye], SHIFT[0.3, 0.5], 0.25 ms]
MoveAOD[[x1_e_alt, ye], SHIFT[0.3, 0.5], FinalPows[x_gate_8pow, y_high_pow],
0.05 ms]
MoveAOD[[x1_e_alt, ye], SHIFT[0.3, 0], 0.095 ms]
WAIT[10 us]
MoveAOD[[x1_e_alt, ye], SHIFT[0.3, 0.5], 0.095 ms]
MoveAOD[[x1_e_alt, ye], SHIFT[0.3, 0.5], FinalPows[x_high_8pow, y_high_pow],
0.05 ms]
MoveAOD[[x1_e_alt, ye], SHIFT[2, 0.5], 0.25 ms]
MoveAOD[[x1_e_alt, ye], SHIFT[2, 0], 0.1 ms]
MoveAOD[[x1_e_alt, ye], FinalPows[x_init_8pow, y0_pow], SHIFT[2, 0], 0.15 ms]

WAIT[99 us]
RESET[]


NOTE[Second gate layer of state prep]
MoveAOD[Initial[xe, y1_e_alt],
InitialPows[x0_pow, y0_4pow],
FinalPows[x_high_pow, y_4pow], SHIFT[0, 0], 0.15 ms]
MoveAOD[[xe, y1_e_alt], SHIFT[0.5, 0], 0.1 ms]
```

```
MoveAOD[[xe, y1_e_alt], SHIFT[0.5, 2], 0.25 ms]
MoveAOD[[xe, y1_e_alt], SHIFT[0.5, 2], FinalPows[x_gate_pow, y_4pow], 0.05 ms]
MoveAOD[[xe, y1_e_alt], SHIFT[0.3, 2], 0.095 ms]
WAIT[10 us]
MoveAOD[[xe, y1_e_alt], SHIFT[0.5, 2], 0.095 ms]
MoveAOD[[xe, y1_e_alt], SHIFT[0.5, 2], FinalPows[x_high_pow, y_4pow], 0.05 ms]
MoveAOD[[xe, y1_e_alt], SHIFT[0.5, 0], 0.25 ms]
MoveAOD[[xe, y1_e_alt], SHIFT[0, 0], 0.1 ms]
MoveAOD[[xe, y1_e_alt], FinalPows[x0_pow, y0_4pow], SHIFT[0, 0], 0.15 ms]

WAIT[99 us]
RESET[]

NOTE[Third gate layer of state prep]
MoveAOD[Initial[x2_e_alt, ye],
InitialPows[x_init_8pow, y0_pow],
FinalPows[x_high_8pow, y_high_pow], SHIFT[4, 0], 0.12 ms]
MoveAOD[[x2_e_alt, ye], SHIFT[4, 0.5], 0.1 ms]
MoveAOD[[x2_e_alt, ye], SHIFT[0.3, 0.5], 0.3 ms]
MoveAOD[[x2_e_alt, ye], SHIFT[0.3, 0.5], FinalPows[x_gate_8pow, y_high_pow],
0.03 ms]
MoveAOD[[x2_e_alt, ye], SHIFT[0.3, 0], 0.095 ms]
WAIT[10 us]
MoveAOD[[x2_e_alt, ye], SHIFT[0.3, 0.5], 0.095 ms]
MoveAOD[[x2_e_alt, ye], SHIFT[0.3, 0.5], FinalPows[x_high_8pow, y_high_pow],
0.03 ms]
MoveAOD[[x2_e_alt, ye], SHIFT[4, 0.5], 0.3 ms]
MoveAOD[[x2_e_alt, ye], SHIFT[4, 0], 0.1 ms]
MoveAOD[[x2_e_alt, ye], FinalPows[x_init_8pow, y0_pow], SHIFT[4, 0], 0.12 ms]

WAIT[99 us]
RESET[]

NOTE[Fourth gate layer of state prep]
MoveAOD[Initial[xe, y2_e_alt],
InitialPows[x0_pow, y0_4pow],
FinalPows[x_high_pow, y_4pow], SHIFT[0, 4], 0.15 ms]
MoveAOD[[xe, y2_e_alt], SHIFT[0.5, 4], 0.1 ms]
MoveAOD[[xe, y2_e_alt], SHIFT[0.5, 0], 0.25 ms]
MoveAOD[[xe, y2_e_alt], SHIFT[0.5, 0], FinalPows[x_gate_pow, y_4pow], 0.05 ms]
MoveAOD[[xe, y2_e_alt], SHIFT[0.3, 0], 0.095 ms]
WAIT[10 us]
MoveAOD[[xe, y2_e_alt], SHIFT[0.5, 0], 0.095 ms]
MoveAOD[[xe, y2_e_alt], SHIFT[0.5, 0], FinalPows[x_high_pow, y_4pow], 0.05 ms]
MoveAOD[[xe, y2_e_alt], SHIFT[0.5, 4], 0.25 ms]
MoveAOD[[xe, y2_e_alt], SHIFT[0, 4], 0.1 ms]
MoveAOD[[xe, y2_e_alt], FinalPows[x0_pow, y_4pow], SHIFT[0, 4], 0.15 ms]

WAIT[99 us]
RESET[]


NOTE[Transversal cluster state in space]
MoveAOD[Initial[xe, y_e_top],
```

```
InitialPows[x0_pow, y0_4pow],
FinalPows[x_high_pow, y_4pow], SHIFT[0, 8], 0.15 ms]
MoveAOD[[xe, y_e_top], SHIFT[-0.5, 8], 0.1 ms]
MoveAOD[[xe, y_e_top], SHIFT[-0.5, 0], 0.4 ms]
MoveAOD[[xe, y_e_top], SHIFT[-0.3, 0], 0.095 ms]

WAIT[10 us]

MoveAOD[[xe, y_e_top], FinalPows[x_high_pow, y_4pow], SHIFT[-0.5, 0], 0.095 ms]
MoveAOD[[xe, y_e_top], SHIFT[-0.5, 0.5], 0.1 ms]
MoveAOD[[xe, y_e_top], SHIFT[-7.5, 0.5], 0.45 ms]
MoveAOD[[xe, y_e_top], SHIFT[-7.5, 0.5], 0.1 ms]
MoveAOD[[xe, y_e_top], SHIFT[-7.7, 0], 0.095 ms]

WAIT[10 us]

MoveAOD[[xe, y_e_top], FinalPows[x_high_pow, y_4pow], SHIFT[-7.5, 0.5], 0.095
ms]
MoveAOD[[xe, y_e_top], SHIFT[-7.5, 0.5], 0.1 ms]
MoveAOD[[xe, y_e_top], SHIFT[-0.5, 0.5], 0.45 ms]
MoveAOD[[xe, y_e_top], SHIFT[-0.5, 0], 0.1 ms]
MoveAOD[[xe, y_e_top], SHIFT[-0.3, 0], 0.095 ms]

WAIT[10 us]

MoveAOD[[xe, y_e_top], FinalPows[x_high_pow, y_4pow], SHIFT[-0.5, 0], 0.095 ms]
MoveAOD[[xe, y_e_top], SHIFT[-0.5, 8], 0.4 ms]
MoveAOD[[xe, y_e_top], SHIFT[0, 8], 0.1 ms]
MoveAOD[[xe, y_e_top],
FinalPows[x0_pow, y0_4pow], SHIFT[0, 8], 0.149 ms]
RESET[]


NOTE[Group A: storage to entangling]
MoveAOD[Initial[xe, y_st],
InitialPows[x0_pow, y0_pow],
FinalPows[x_high_pow, y2_up_pow], SHIFT[0, 0], 0.15 ms]
MoveAOD[[xe, y_st], SHIFT[0.5, 0], 0.1 ms]
MoveAOD[[xe, ye], SHIFT[0.5, 1], 0.37 ms]
MoveAOD[[xe, ye], SHIFT[1, 1], 0.13 ms]


WAIT[100 us]
MoveAOD[[xe, ye], FinalPows[x_gate_pow, y2_up_pow], SHIFT[1, 1], 0.03 ms]
MoveAOD[[xe, ye], SHIFT[0.3, 0], 0.115 ms]
NOTE[Transversal CNOT for time direction of cluster state]
WAIT[10 us]
MoveAOD[[xe, ye], SHIFT[1, 1], 0.115 ms]
MoveAOD[[xe, ye], FinalPows[x_high_pow, y2_up_pow], SHIFT[1, 1], 0.03 ms]
WAIT[100 us]

NOTE[Group A: Entangling to readout]
MoveAOD[[xe, y_SSR], SHIFT[0.5, 9], 0.3 ms]
MoveAOD[[xe, y_SSR], SHIFT[1, 9], 0.1 ms]
```

```
MoveAOD[[xe, y_SSR], FinalPows[x_high_pow, y1_up_pow], SHIFT[1, 9], 0.15 ms]
MoveAOD[[xe, y_SSR], SHIFT[0.5, 9], 0.1 ms]
MoveAOD[[xe, y_SSR], SHIFT[0.5, 1], 0.2 ms]
MoveAOD[[xe, y_SSR], SHIFT[0, 1], 0.1 ms]
MoveAOD[[xe, y_SSR], FinalPows[x0_pow, y0_pow], SHIFT[0, 1], 0.149 ms]


RESET[]

NOTE[Group B: Entangling to storage]
MoveAOD[Initial[xe, ye],
InitialPows[x0_pow, y0_pow],
FinalPows[x_high_pow, y2_up_pow], 0.15 ms]
MoveAOD[[xe, ye], SHIFT[0.5, 0], 0.1 ms]
MoveAOD[[xe, y_st], SHIFT[0.5, 0], 0.4 ms]
MoveAOD[[xe, y_st], SHIFT[0, 0], 0.1 ms]
MoveAOD[[xe, y_st], FinalPows[x0_pow, y0_pow], SHIFT[0, 0], 0.15 ms]


WAIT[1.499 ms]
RESET[]

NOTE[Readout group A]
MoveAOD[Initial[x_SSR, y2_SSR],
InitialPows[x0_32pow, y0_4pow],
FinalPows[xSSR_32pow, ySSR_4pow], SHIFT[0, 1], 0.25 ms]
MoveAOD[[x_SSR, y2_SSR], SHIFT[0.5, 1], 0.5 ms]
MoveAOD[[x_SSR, y2_SSR], SHIFT[0.5, 0], 0.3 ms]
MoveAOD[[x_SSR, y2_SSR], SHIFT[0, 0], 0.2 ms]
MoveAOD[[x_SSR, y2_SSR],
FinalPows[x0_32pow, y0_4pow], SHIFT[0, 0], 0.25 ms]

SSR_SAVE_PREVIOUS[]
WAIT[10.5 ms]

RESET[]
WAIT[14 us]
NOTE[Recombine spin states to original tweezer positions]
MoveAOD[Initial[x_SSR, y2_SSR],
InitialPows[x0_32pow, y0_4pow],
FinalPows[xSSR_32pow, ySSR_4pow], SHIFT[0, 0], 0.5 ms]
MoveAOD[[x_SSR, y2_SSR], SHIFT[0.5, 0], 0.5 ms]
MoveAOD[[x_SSR, y2_SSR], FinalPows[xSSR_32pow, ylowSSR_4pow], SHIFT[0.5, 0],
0.3 ms]
MoveAOD[[x_SSR, y2_SSR], SHIFT[0.5, 1], 0.8 ms]
MoveAOD[[x_SSR, y2_SSR], SHIFT[0, 1], 0.8 ms]
MoveAOD[[x_SSR, y2_SSR],
FinalPows[x0_32pow, y0_4pow], SHIFT[0, 1], 0.499 ms]

RESET[]
WAIT[11.427133 ms]

END_COMMAND[]
```

Raman AWG command string

The command string used to program the I/Q waveforms and AOM amplitudes for local and global Raman control is given below. The command specifies the parameters for each single-qubit gate, which is then converted into the appropriate waveforms to modulate the light intensity and microwave phase and frequency (defined relative to the resonance frequency).
For local gates, the pulses are optionally gated on and off by the Raman AOD command.

Examples of typical commands are
Z : global Z rotation, in units of 2*pi
E : wait time, idling at the specified frequency
SC1_ECHO : global SCROFULOUS Y(pi) pulse for dynamical decoupling, idling at the specified frequency
SC1_H_LOCAL : SCROFULOUS Y(pi/2) local gaussian pulses
SC1_PI_ECHO : SCROFULOUS Y(pi) local gaussian pulses
FIXEDPUMPINGLOOPEDTTL: global Raman-assisted optical pumping
FIXEDPUMPINGLOOPEDTTLMOD: local Raman-assisted optical pumping

```
### Define variables for repeated commands

STARTVAR[ADDLAYER]

SC1_ECHO[1.356885 ms, 0.02 MHz]
PREP_GATES
SPACE_TRANSVERSAL_GATE # 3 pi
TIME_TRANSVERSAL_GATE
STORE_AND_READ # 3 pi
READOUT_ECHOES # 5 pi
REINITIALISE # 1 pi

ENDVAR

STARTVAR[SPACE_TRANSVERSAL_GATE]
SC1_ECHO[1.5 ms, 0.02 MHz]
Z[0.205]
SC1_ECHO[0.19992 ms, 0.02 MHz]
Z[0.205]
SC1_ECHO[1.5 ms, 0.02 MHz]
ENDVAR

STARTVAR[PREP_GATES]

Z[0.0]
SC1_H_LOCAL[0.01us, 0.928, 4, 46 us, 0.5, 8, 1]
Z[0]
SC1_ECHO[1.40792 ms, 0.02 MHz]
Z[0.205]
SC1_H_LOCAL[0.01us, 0.928, 4, 46 us, 0.5, 8, 1]
Z[0]
SC1_ECHO[1.29992 ms, 0.02 MHz]
Z[0.205]
SC1_H_LOCAL[0.01us, 0.928, 4, 46 us, 0.5, 8, 1]
Z[0]
SC1_ECHO[1.40792 ms, 0.02 MHz]
```

```
Z[0.205]
SC1_H_LOCAL[0.01us, 0.928, 4, 46 us, 0.5, 8, 1]
Z[0]
SC1_ECHO[1.29992 ms, 0.02 MHz]
Z[0.205]
SC1_H_LOCAL[0.01us, 0.886, 4, 46 us, 0.5, 8, 1]
SC1_ECHO[8 us, 0.02 MHz]
SC1_PI_LOCAL[0.01us, 0.96, 4, 46 us, 0.5, 8, 1]

ENDVAR


STARTVAR[TIME_TRANSVERSAL_GATE]
NOTE[storage to entangling here]
SC1_ECHO[0.150 ms, 0.02 MHz]
SC1_ECHO[0.600 ms, 0.02 MHz]

SC1_ECHO[100 us, 0.02 MHz]
Z[0.00]
SC1_ECHO[299.92 us, 0.02 MHz]
Z[0.205]
SC1_ECHO[53.92 us, 0.02 MHz]
Z[0.785]
SC1_H_LOCAL_PHASES[0.01us, 0.928, 4, 46 us, 0.5+0.5+0.5+0.50+0.5+0.5+0.5+0.5,
8, 1]
Z[-0.785]
SC1_ECHO[8 us, 0.02 MHz]
SC1_H_LOCAL[0.01us, 0.928, 4, 46 us, 0.5, 8, 1]
ENDVAR

STARTVAR[TIME_TRANSVERSAL_NO_GATE]
NOTE[storage to entangling here]
SC1_ECHO[0.150 ms, 0.02 MHz]
SC1_ECHO[0.600 ms, 0.02 MHz]

SC1_ECHO[100 us, 0.02 MHz]
Z[0.00]
SC1_ECHO[299.92 us, 0.02 MHz]
Z[0.00]
SC1_ECHO[53.92 us, 0.02 MHz]
Z[0.785]
SC1_H_LOCAL_PHASES[0.01us, 0.928, 4, 46 us, 0.5+0.5+0.5+0.5+0.5+0.5+0.5+0.5, 8,
1]
Z[-0.785]
SC1_ECHO[8 us, 0.02 MHz]
SC1_H_LOCAL[0.01us, 0.928, 4, 46 us, 0.5, 8, 1]
ENDVAR


STARTVAR[STORE_AND_READ]
NOTE[entangling to readout]
SC1_ECHO[0.5234 ms, 0.02 MHz]
SC1_ECHO[0.5234 ms, 0.02 MHz]
NOTE[entangling to storage]
```

```
SC1_ECHO[0.750 ms , 0.02 MHz]
SC1_ECHO[0.150 ms, 0.02 MHz]
ENDVAR


STARTVAR[READOUT_ECHOES]
NOTE[Lattice pumping here]
SC1_ECHO[3.9903 ms, 0.02 MHz]

SC1_ECHO[3.33 ms, 0.02 MHz]
SC1_ECHO[3.33 ms, 0.02 MHz]
SC1_ECHO[3.34 ms, 0.02 MHz]

Z[0.5]
SC1_ECHO[3.299975 ms, 0.02 MHz]

SC1_ECHO[3.34 ms, 0.02 MHz]
SC1_ECHO[3.33 ms, 0.02 MHz]
SC1_ECHO[3.33 ms, 0.02 MHz]

SC1_ECHO[106 us, 0.02 MHz]

ENDVAR

STARTVAR[REINITIALISE]
FIXEDPUMPINGLOOPEDTTLMOD[0, 2.6 us, -12.55 MHz, 12.55 MHz, 1, 6, 46, 10 us,
0.1us, 1, 464.5 us, local]
SC1_ECHO[14.38 us, 0.02 MHz]
FIXEDPUMPINGLOOPEDTTLMOD[0, 2.6 us, -12.55 MHz, 12.55 MHz, 1, 6, 46, 10 us,
0.1us, 0, 464.5 us, local]
ENDVAR


NOTE[CIRCUIT STARTS HERE]


ADDSEGMENT[0]
FIXEDPUMPINGLOOPEDTTL[0, 2.66 us, -11.6 MHz, 11.6 MHz, 1, 4, 24, 10 us, 0.1us]
E[10.625005 ms, 5MHz]


ADDSEGMENT[1]
E[1.356885 ms, 5 MHz]
PREP_GATES
SPACE_TRANSVERSAL_GATE
TIME_TRANSVERSAL_NO_GATE
STORE_AND_READ # 3 pi
READOUT_ECHOES # 5 pi
REINITIALISE # 1 pi
ADDSEGMENT[2]
ADDLAYER
ADDSEGMENT[3]
ADDLAYER
ADDSEGMENT[4]
```

```
ADDLAYER
ADDSEGMENT[5]
ADDLAYER
ADDSEGMENT[6]
ADDLAYER
ADDSEGMENT[7]
ADDLAYER
ADDSEGMENT[8]
ADDLAYER
ADDSEGMENT[9]
ADDLAYER
ADDSEGMENT[10]
ADDLAYER
ADDSEGMENT[11]
ADDLAYER
ADDSEGMENT[12]
ADDLAYER
ADDSEGMENT[13]
ADDLAYER
ADDSEGMENT[14]
ADDLAYER
ADDSEGMENT[15]
ADDLAYER
ADDSEGMENT[16]
ADDLAYER
ADDSEGMENT[17]
ADDLAYER
ADDSEGMENT[18]
ADDLAYER
ADDSEGMENT[19]
ADDLAYER
ADDSEGMENT[20]
ADDLAYER
ADDSEGMENT[21]
ADDLAYER
ADDSEGMENT[22]
ADDLAYER
ADDSEGMENT[23]
ADDLAYER
ADDSEGMENT[24]
ADDLAYER
ADDSEGMENT[25]
ADDLAYER
ADDSEGMENT[26]
ADDLAYER
ADDSEGMENT[27]
ADDLAYER
ADDSEGMENT[28]
ADDLAYER

END_COMMAND[]
```

Raman AOD AWG command string

The command string used to program the RF waveforms for the Raman AOD AWG, which controls the positions and powers of 2D grids of optical tweezers used for local single-qubit gates, is given below.
Positions and powers are separately calibrated and loaded from a saved file. Different pulse configurations are specified by variables, for example 'r_0_t_0_' corresponding to pulses on row 0 at time step 0 of this particular circuit.

```
FILE[qubit_reuse_variables_tesseract_++++++.txt]
VAR[turnoff[], MOVEAOD[Initial[[-3],[-3]],INITIALPOWS[[0],[0]],10ns]]
VAR[extra_gap[], WAIT[108 us]]
VAR[pi_gap[], WAIT[7.99 us]]
VAR[RMgate_gap[],  WAIT[1299.98 us]]
VAR[pump_gap, WAIT[9.99us]]

SEGMENT[1, LOOP_UNTIL_TRIG]


WAIT[1.35343 ms]

LOOPPULSES[r_0_t_0_, r_1_t_0_, r_2_t_0_, r_3_t_0_, r_4_t_0_, r_5_t_0_,
r_6_t_0_, r_7_t_0_, 46us]
turnoff[]
extra_gap[]
RMgate_gap[]

LOOPPULSES[r_7_t_1_, r_6_t_1_, r_5_t_1_, r_4_t_1_, r_3_t_1_, r_2_t_1_,
r_1_t_1_, r_0_t_1_, 46us]
turnoff[]
RMgate_gap[]

LOOPPULSES[r_0_t_2_, r_1_t_2_, r_2_t_2_, r_3_t_2_, r_4_t_2_, r_5_t_2_,
r_6_t_2_, r_7_t_2_, 46us]
turnoff[]
extra_gap[]
RMgate_gap[]

LOOPPULSES[r_7_t_3_, r_6_t_3_, r_5_t_3_, r_4_t_3_, r_3_t_3_, r_2_t_3_,
r_1_t_3_, r_0_t_3_, 46us]
turnoff[]
RMgate_gap[]

LOOPPULSES[r_0_t_4_, r_1_t_4_, r_2_t_4_, r_3_t_4_, r_4_t_4_, r_5_t_4_,
r_6_t_4_, r_7_t_4_, 46us]
turnoff[]
pi_gap[]
WAIT[46us]
turnoff[]

WAIT[3.95 ms]


WAIT[46us]
```

```
turnoff[]
pi_gap[]
WAIT[46us]
turnoff[]

WAIT[300 us]

LOOPPULSES[empty, empty, empty, empty, empty, empty, empty, empty, 46us]
turnoff[]
pi_gap[]
LOOPPULSES[r_aod0, r_aod1, r_aod2, r_aod3, empty, empty, empty, empty, 46us]
turnoff[]

WAIT[29.39950 ms]

LOOPVARS[bright_even,pump_gap,bright_odd,pump_gap,23] # looped local pi pulses
for Raman-assisted optical pumping

MOVEAOD[Initial[[-3],[-3]],INITIALPOWS[[0],[0]],10ns]

WAIT[484.5 us]


WAIT[1.35343 ms]

LOOPPULSES[r_0_t_0_, r_1_t_0_, r_2_t_0_, r_3_t_0_, r_4_t_0_, r_5_t_0_,
r_6_t_0_, r_7_t_0_, 46us]
turnoff[]
extra_gap[]
RMgate_gap[]

LOOPPULSES[r_7_t_1_, r_6_t_1_, r_5_t_1_, r_4_t_1_, r_3_t_1_, r_2_t_1_,
r_1_t_1_, r_0_t_1_, 46us]
turnoff[]
RMgate_gap[]

LOOPPULSES[r_0_t_2_, r_1_t_2_, r_2_t_2_, r_3_t_2_, r_4_t_2_, r_5_t_2_,
r_6_t_2_, r_7_t_2_, 46us]
turnoff[]
extra_gap[]
RMgate_gap[]

LOOPPULSES[r_7_t_3_, r_6_t_3_, r_5_t_3_, r_4_t_3_, r_3_t_3_, r_2_t_3_,
r_1_t_3_, r_0_t_3_, 46us]
turnoff[]
RMgate_gap[]

LOOPPULSES[r_0_t_4_, r_1_t_4_, r_2_t_4_, r_3_t_4_, r_4_t_4_, r_5_t_4_,
r_6_t_4_, r_7_t_4_, 46us]
turnoff[]
pi_gap[]
WAIT[46us]
turnoff[]
```

```
WAIT[3.95 ms]


WAIT[46us]
turnoff[]
pi_gap[]
WAIT[46us]
turnoff[]

WAIT[300 us]

LOOPPULSES[empty, empty, empty, empty, empty, empty, empty, empty, 46us]
turnoff[]
pi_gap[]
LOOPPULSES[empty, empty, empty, empty, r_aod4, r_aod5, r_aod6, r_aod7, 46us]
turnoff[]

WAIT[29.39939 ms]

LOOPVARS[bright_even,pump_gap,bright_odd,pump_gap,23]  # looped local pi pulses
for Raman-assisted optical pumping

MOVEAOD[Initial[[-3],[-3]],INITIALPOWS[[0],[0]],10ns]

WAIT[484.5 us]

END_COMMAND[]
```

Rydberg AWG command string

The command string used to program the AOM amplitude, frequency, and phase for the 420-nm laser for entangling gates is given below. Gate profiles (amplitude, detuning, and phase) are loaded from saved calibration files.

```
Wait[2079.90 us]
Pulse[420Amp[amplitudeFromFile,idx=1002,loops=1],
420Freq[detuningFromFile,idx=1002],
420Phase[phaseFromFile,idx=1002, loops=1]]
Wait[1.398 ms]
Pulse[420Amp[amplitudeFromFile,idx=1002,loops=1],
420Freq[detuningFromFile,idx=1002],
420Phase[phaseFromFile,idx=1002, loops=1]]
Wait[1.398 ms]
Pulse[420Amp[amplitudeFromFile,idx=1002,loops=1],
420Freq[detuningFromFile,idx=1002],
420Phase[phaseFromFile,idx=1002, loops=1]]
Wait[1.398 ms]
Pulse[420Amp[amplitudeFromFile,idx=1002,loops=1],
420Freq[detuningFromFile,idx=1002],
420Phase[phaseFromFile,idx=1002, loops=1]]
Wait[1.498 ms]

Pulse[420Amp[amplitudeFromFile,idx=1002,loops=1],
420Freq[detuningFromFile,idx=1002],
420Phase[phaseFromFile,idx=1002, loops=1]]
Wait[0.848 ms]
Pulse[420Amp[amplitudeFromFile,idx=1002,loops=1],
420Freq[detuningFromFile,idx=1002],
420Phase[phaseFromFile,idx=1002, loops=1]]

Wait[2.598 ms]
Pulse[420Amp[amplitudeFromFile,idx=1002,loops=1],
420Freq[detuningFromFile,idx=1002],
420Phase[phaseFromFile,idx=1002, loops=1]]

Wait[30.6146 ms]
```

Summary of decoders

Here we summarize the different types of decoding and methods of error correction and error detection used throughout this work.

Supplementary Table II: Summary of decoders

Figure	Decoder	Error correction and detection methods
Fig. 1c	MLE, with loss information	All curves use an acceptance fraction of 50%, based on the MLE logical gap.
Fig. 2b,c	N/A	Detectors are postselected based on loss information as described in the caption and Methods.
Fig. 2d,e	As labeled: MLE or ML (FCNN), with or without loss information.	Correction only (no postselection)
Fig. 3c and Fig. 3d (left)	MLE, with loss information	For the transversal CNOT, no postselection is used. For the lattice surgery, error detection is used for the ancilla measurements along the seam.
Fig. 3d (right)	MLE, with loss information	Curves are plotted with different acceptance fractions, based on the MLE logical gap.
Fig. 4a (top)	Look-up table, after loss postselection	The three 3D color code and one 2D color code curves use acceptance fractions of 46% and 74%, respectively.
Fig. 4c,d	Look-up table, after loss postselection	Error detection using all stabilizers in the measured bases (4 in X/Y and 10 in Z, per code)
Fig. 6b	N/A	Stabilizers are postselected on loss.
Fig. 6c,d,g,i	ML (CNN), with loss information	Sliding scale acceptance fraction using the ML decoder confidence. A constant confidence threshold is used when comparing operators of different weight (as opposed to a constant acceptance fraction).

MLE = most-likely error ; ML = machine learning ; FCNN = fully-connected neural network; CNN = convolutional neural network