Preparing Bethe states using Gray codes

Rafael Nepomechie
Department of Physics
University of Miami

Recent Advances in Quantum Integrable Systems (RAQIS)

Annecy, France

September 2026

Outline

I. Gray codes

II. Quantum computers & quantum state preparation

III. Bethe state preparation

- spin 1/2
- spin s
- nesting

new

IV. Conclusions

Outline

I. Gray codes

II. Quantum computers & quantum state preparation

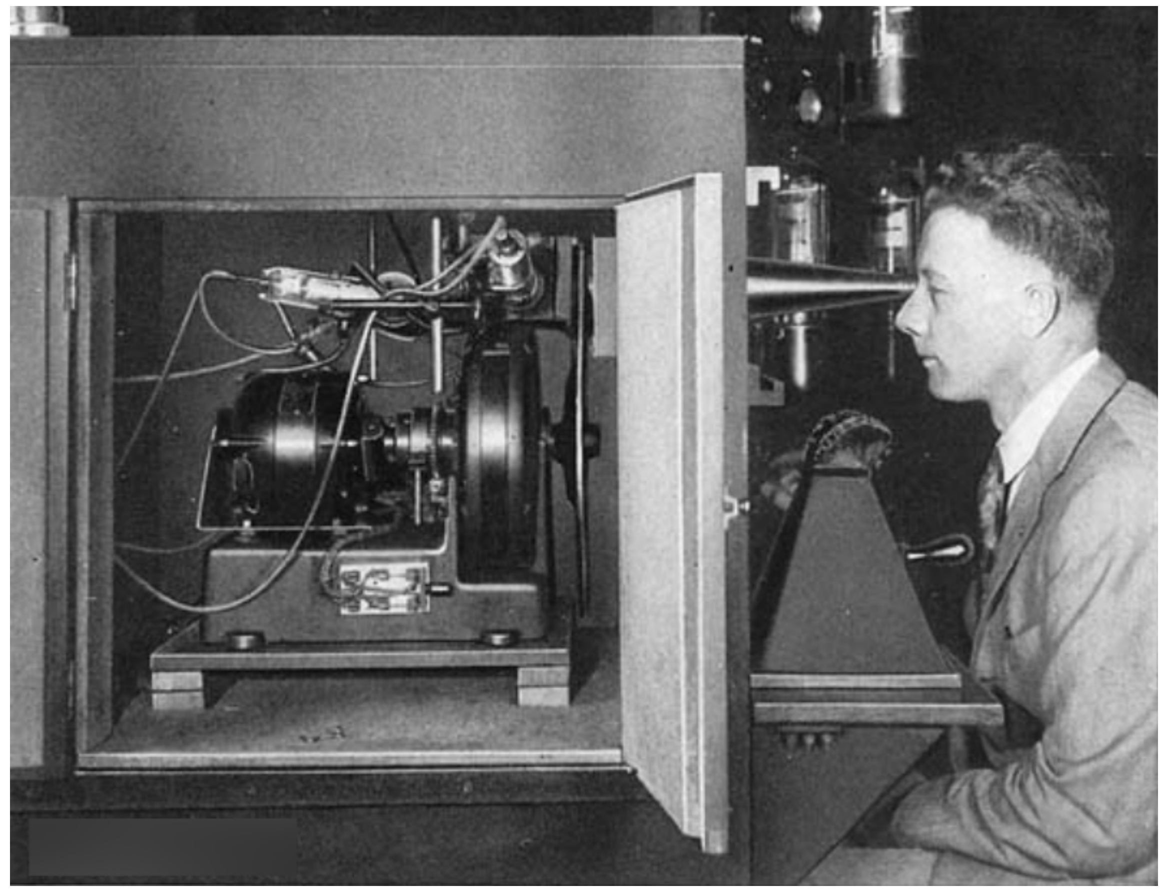
III. Bethe state preparation

- spin 1/2
- spin s
- nesting

new

IV. Conclusions

Gray codes

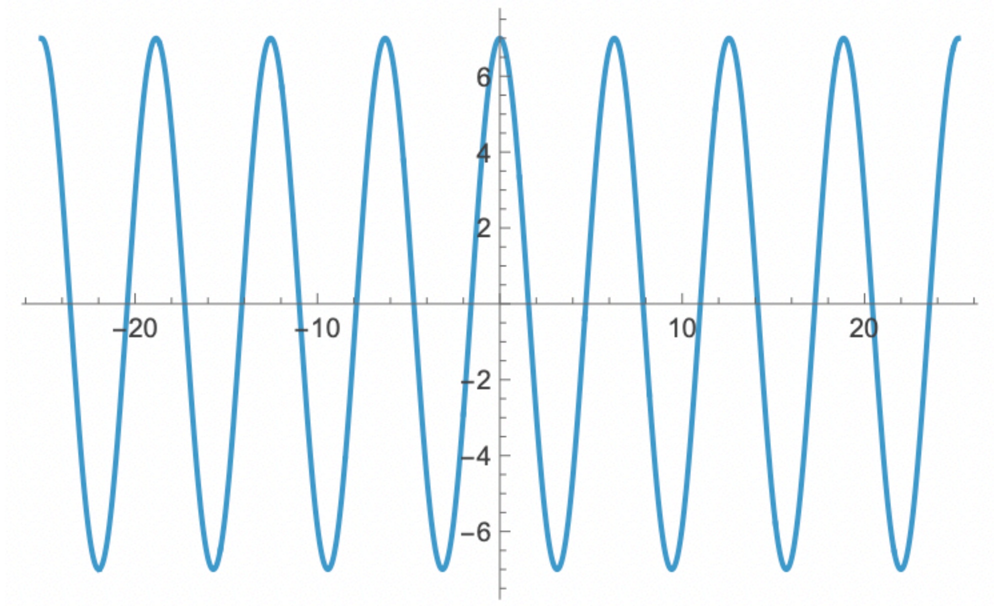


Frank Gray (1887 - 1969)

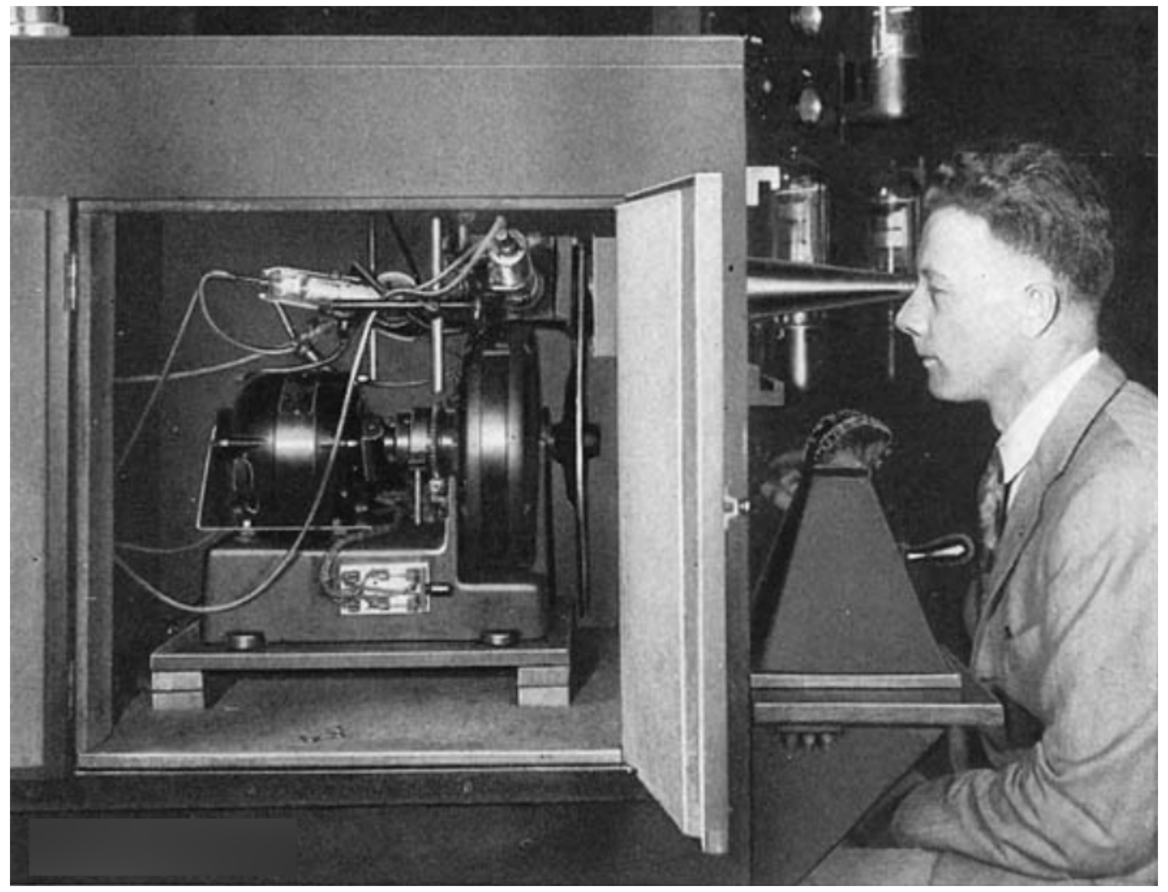
physicist @ Bell Labs

Motivation: Suppose you want to transmit a string of bits (0, 1) using an *analog* device.

naive:	bitstring (binary)	⟶	integer (base 10)	transmit a signal of that amplitude
Example:	0111	⟶	7	



Gray codes



Frank Gray (1887 - 1969)

physicist @ Bell Labs

Motivation: Suppose you want to transmit a string of bits (0, 1) using an *analog* device.

naive: bitstring (binary) ⟶ integer (base 10) transmit a signal of that amplitude

problem: small error in signal amplitude

$\Rightarrow$ small error in integer

$\Rightarrow$ possibly *huge* error in bitstring

e.g.

	7:	0111	
small error	$\updownarrow$	$\updownarrow$	*huge* error
	8:	1000	

How to associate integers with bitstrings, such that

small changes in integer $\Rightarrow$ small changes in its bitstring ?

one way: list the bitstrings in a sequence, such that each string differs from its successor in a small way

Example: bitstrings of length 4

$2^4 = 16$ possibilities

"Gray code"

only one bit changes at each step

	binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

one way: list the bitstrings in a sequence, such that each string differs from its successor in a small way

Example: bitstrings of length 4

"Gray code"

$2^4 = 16$ possibilities

	binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

only one bit changes at each step

one way: list the bitstrings in a sequence, such that each string differs from its successor in a small way

Example: bitstrings of length 4

$2^4 = 16$ possibilities

"Gray code"

	binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

only one bit changes at each step

one way: list the bitstrings in a sequence, such that each string differs from its successor in a small way

"Gray code"

Example: bitstrings of length 4

$2^4 = 16$ possibilities

	binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

only one bit changes at each step

etc.

one way: list the bitstrings in a sequence, such that each string differs from its successor in a small way

"Gray code"

Example: bitstrings of length 4

$2^4 = 16$ possibilities

	binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

only one bit changes at each step

"binary reflected Gray code"

one way: list the bitstrings in a sequence, such that each string differs from its successor in a small way

"Gray code"

- used by Gray for pulse-code modulation - patent 1953
- invented earlier: George Stibitz (patent 1943), Émile Baudot (1875), Louis Gros (1872), …
- still widely used for analog-to-digital conversion
- nice mathematical properties

recursive construction: $\mathcal{L}_n$: list of all n-bit strings arranged in Gray code order

$$\begin{array}{ccc} \mathcal{L}_1 & & \\ 0 & \longrightarrow & 00 \\ 1 & & 01 \end{array}$$

reverse: $\bar{\mathcal{L}}_1$

$$\begin{array}{ccc} 1 & \longrightarrow & 11 \\ 0 & & 10 \end{array}$$

recursive construction: $\mathcal{L}_n$: list of all n-bit strings arranged in Gray code order

	$\mathcal{L}_1$		$\mathcal{L}_2$
	0	$\longrightarrow$	00
	1		01
reverse:	$\bar{\mathcal{L}}_1$		
	1	$\longrightarrow$	11
	0		10

recursive construction: $\mathcal{L}_n$: list of all n-bit strings arranged in Gray code order

$\mathcal{L}_1$		$\mathcal{L}_2$
0	$\longrightarrow$	00
1		01
reverse: $\bar{\mathcal{L}}_1$		
1	$\longrightarrow$	11
0		10

$$\mathcal{L}_n := 0 \oplus \mathcal{L}_{n-1},\ 1 \oplus \bar{\mathcal{L}}_{n-1}$$

	binary	Gray code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

- many generalizations studied in combinatorics

"Gray property": *small* change at each step

Outline

I. Gray codes

II. Quantum computers & quantum state preparation

III. Bethe state preparation

- spin 1/2
- spin s
- nesting

new

IV. Conclusions

Quantum computers & quantum state preparation

Quantum computer: device for preparing and measuring multi-qubit states

Bet($$$) : provide advantage (over classical devices) for certain tasks

	classical		quantum	
Advantage for *storing* n-qubit quantum state:	2^n complex numbers	vs	n qubits	🙂
	scales exponentially in n		scales linearly in n	

But how to *prepare* a given quantum state on a quantum computer,

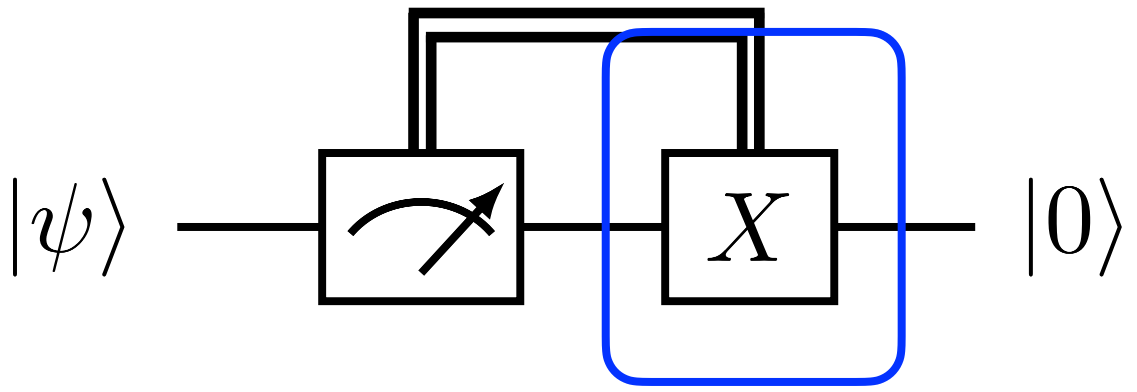
using the elementary 1-qubit & 2-qubit "gates" (unitary operators) that are typically available?

Quantum state preparation problem

Simple examples:

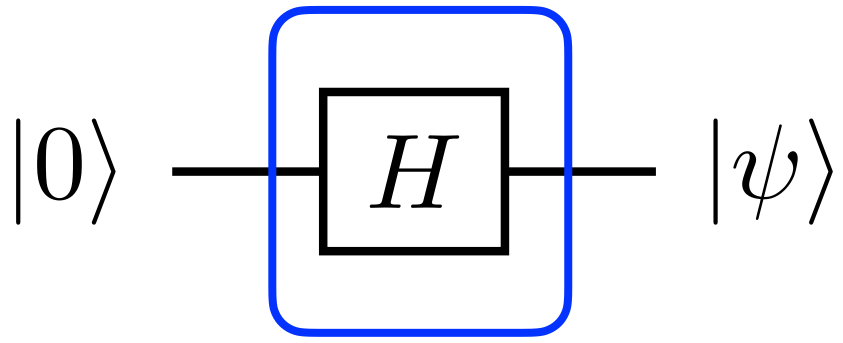
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$|0\rangle$:



conditional NOT

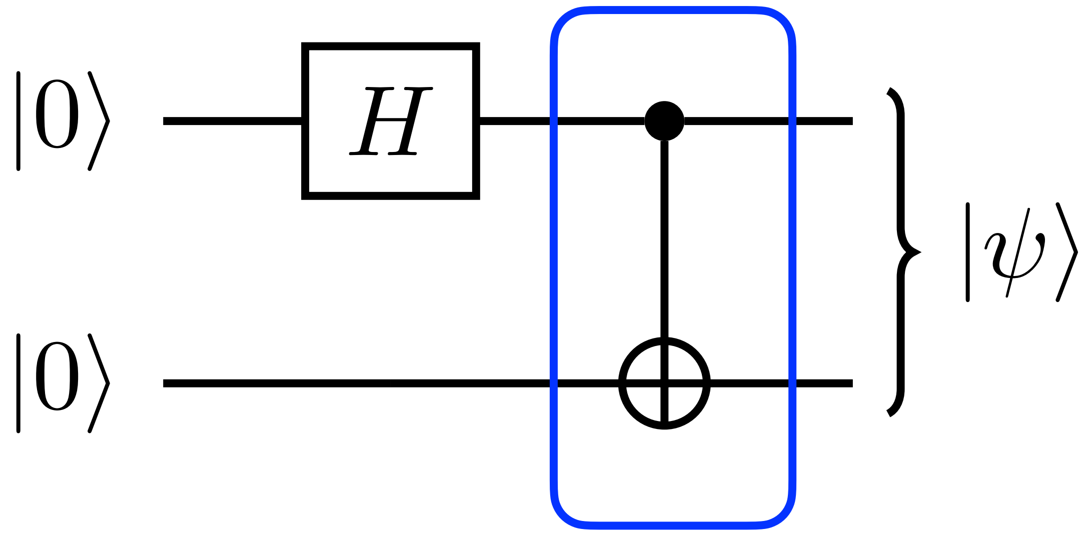
$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) :$$



Hadamard

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) :$$



CNOT

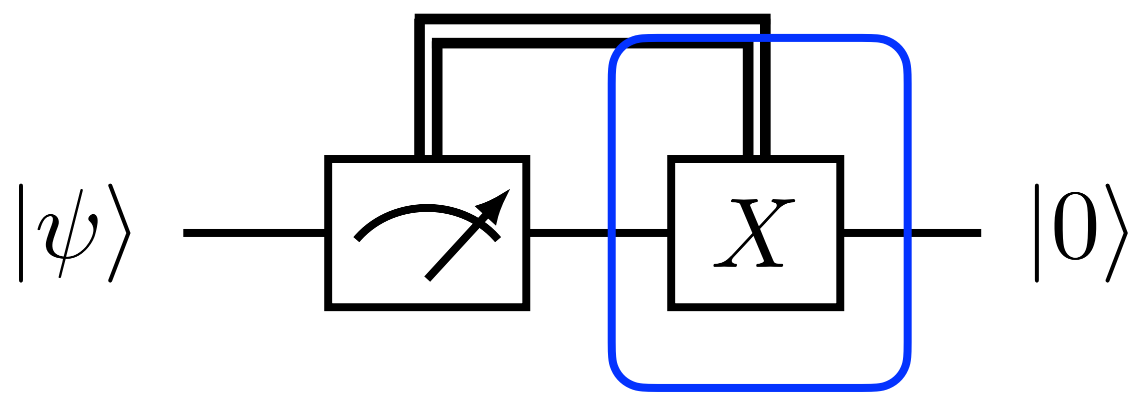
$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$|00\rangle = |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad |01\rangle = |0\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad |10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad |11\rangle = |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Simple examples:

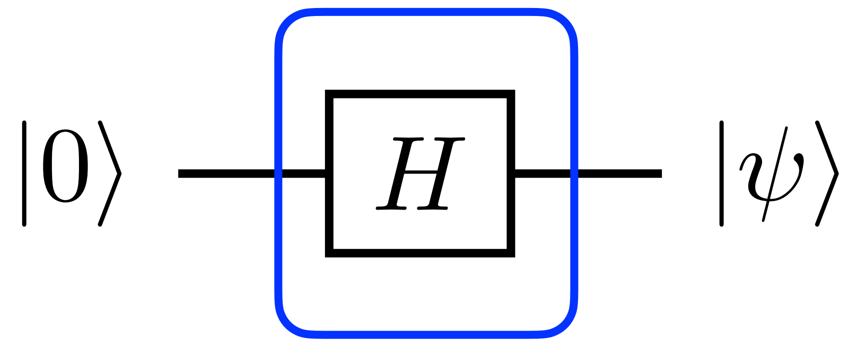
$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$|0\rangle$:



conditional NOT

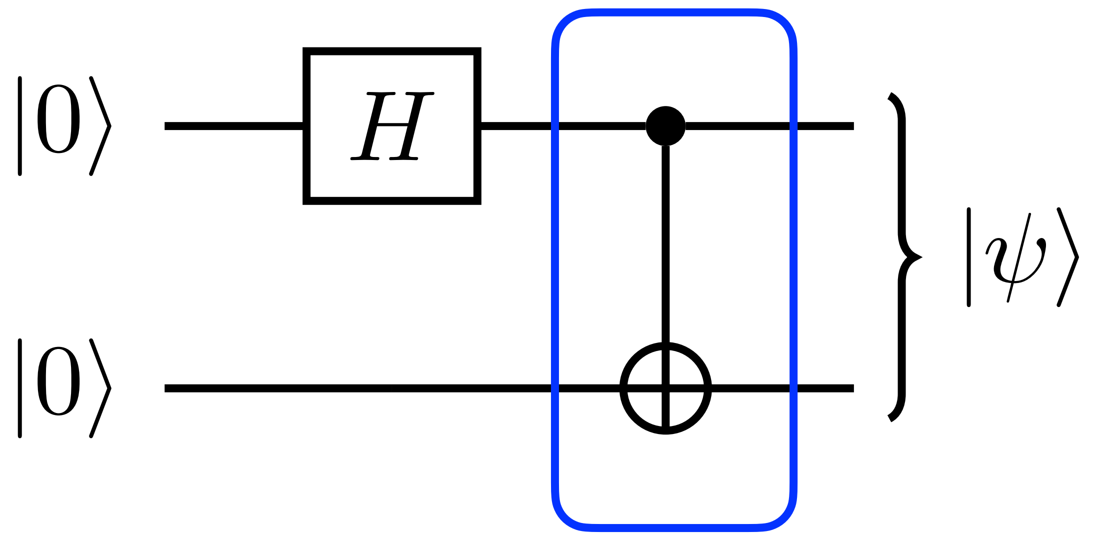
$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$:



Hadamard

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$:



CNOT

$$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

To prepare a generic n-qubit quantum state, need $\mathcal{O}(c^n)$ elementary gates ☹️

(Product states require few gates; entanglement makes preparation more difficult.)

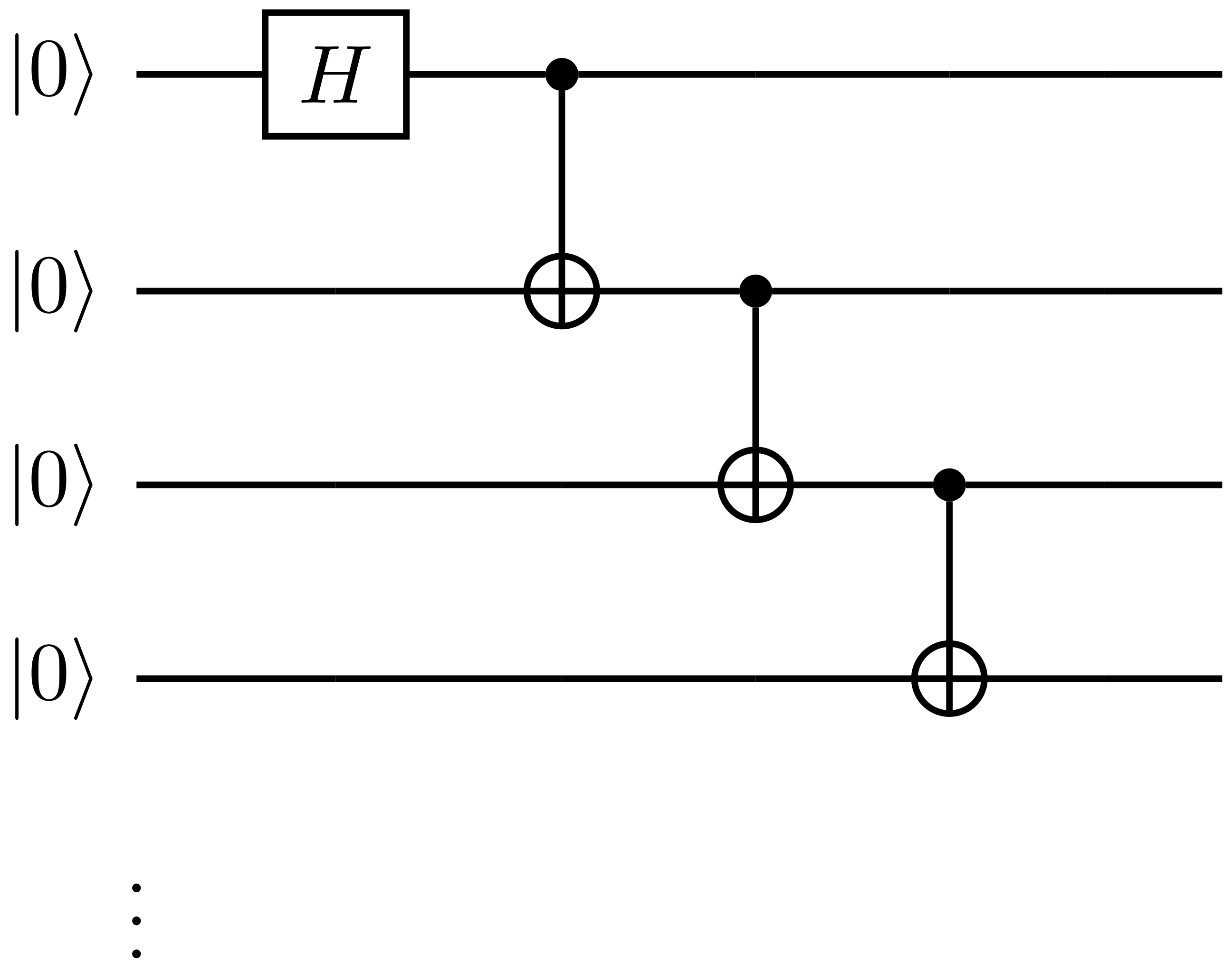
Certain entangled n-qubit states can nevertheless be prepared efficiently! 🙂

(size & depth of quantum circuit are *polynomial* in n)

GHZ (Greenberger-Horne-Zeilinger) states

$$|\mathrm{GHZ}_n\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n})$$

$n = 120$ IBM Javadi-Abhari et al. 2510.09520



- circuit size & depth $\mathcal{O}(n)$
- constant depth using intermediate measurements Smith et al. 2210.17548; Bäumer et al. 2308.13065; …

Dicke states

$|D_k^n\rangle$: equal-weight superposition of all n-qubit basis states with k $|1\rangle$'s

\# 1's

$\mathbin{/\!/}$

\# nonzero bits

$\mathbin{/\!/}$

fixed Hamming weight

permutation-invariant

Ex: $$|D_2^4\rangle = \frac{1}{\sqrt{6}}\left(|1100\rangle + |1010\rangle + |0110\rangle + |1001\rangle + |0101\rangle + |0011\rangle\right)$$

$$|D_k^n\rangle \propto \left(\mathbb{S}^-\right)^k |0\rangle^{\otimes n}$$

$$\mathbb{S}^- = \mathbb{S}^x - i\mathbb{S}^y$$

total spin lowering

$$\vec{\mathbb{S}} = \sum_{i=1}^n \vec{S}_i$$

total spin

$$\vec{S}_i = \mathbb{I}\otimes\ldots\otimes\mathbb{I}\otimes \overset{\overset{i}{\downarrow}}{\frac{\vec{\sigma}}{2}} \otimes\mathbb{I}\otimes\ldots\otimes\mathbb{I}$$

spin-1/2

How to prepare?

Recursive algorithm

Bärtschi, Eidenbenz 1904.07358

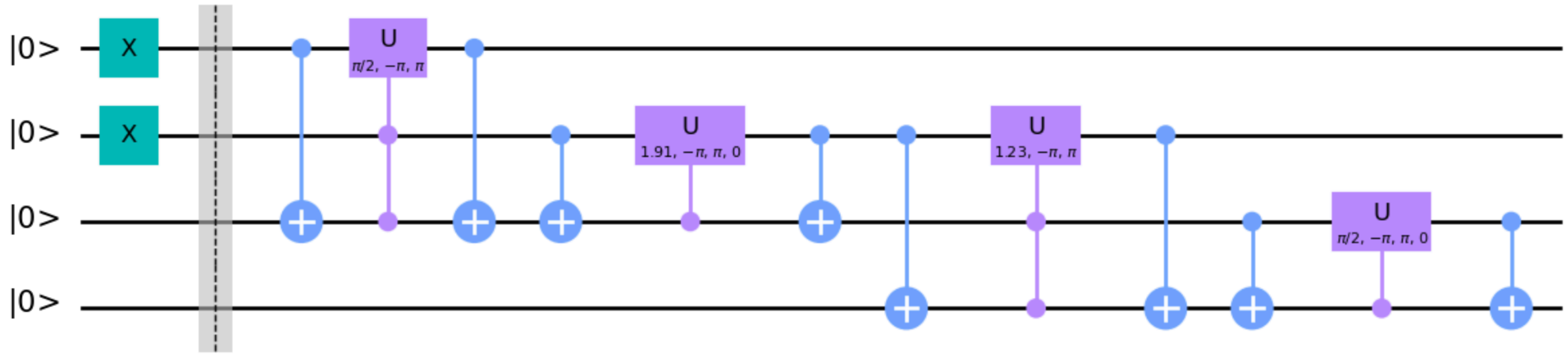
Ex:

$$|D_2^4\rangle = \frac{1}{\sqrt{6}}\Big(\underbrace{|1100\rangle + |1010\rangle + |0110\rangle}_{(|110\rangle + |101\rangle + |011\rangle)\otimes|0\rangle} + \underbrace{|1001\rangle + |0101\rangle + |0011\rangle}_{(|100\rangle + |010\rangle + |001\rangle)\otimes|1\rangle}\Big)$$

$$|D_2^4\rangle = \sqrt{\frac{1}{2}}\,|D_2^3\rangle \otimes |0\rangle + \sqrt{\frac{1}{2}}\,|D_1^3\rangle \otimes |1\rangle$$

- circuit size & depth $\mathcal{O}\left(k(n-k)\right)$

Ex: $|D_2^4\rangle$



- other approaches

Wang, Terhal 2104.14310; Buhrman et al. 2307.14840; Bond et al. 2312.05060; Piroli et al. 2403.07604; Yu et al. 2411. 03428; Liu et al. 2411. 04019; Vasconcelos, Joshi 2601.10693

- qudit (d-level) generalizations

$$|0\rangle = \begin{pmatrix}1\\0\\0\\\vdots\\0\end{pmatrix}, \quad |1\rangle = \begin{pmatrix}0\\1\\0\\\vdots\\0\end{pmatrix}, \ldots, \quad |d-1\rangle = \begin{pmatrix}0\\0\\\vdots\\0\\1\end{pmatrix}$$

"su(2) spin-s Dicke states" $\quad d = 2s+1 \quad s = \frac{1}{2}, 1, \frac{3}{2}, \ldots$

$$|D^{(s)}_{n,k}\rangle \propto \left(\mathbb{S}^-\right)^k |0\rangle^{\otimes n} \qquad \vec{\mathbb{S}} = \sum_{i=1}^{n} \vec{S}_i \qquad \vec{S}_i = \mathbb{I} \otimes \ldots \otimes \mathbb{I} \otimes \overset{\overset{i}{\downarrow}}{\vec{S}} \otimes \mathbb{I} \otimes \ldots \otimes \mathbb{I}$$

spin-s

Example with s=1: $|D^{(1)}_{3,2}\rangle = \frac{2}{\sqrt{15}}\left(|011\rangle + |101\rangle + |110\rangle\right) + \frac{1}{\sqrt{15}}\left(|002\rangle + |020\rangle + |200\rangle\right)$ digit sum = 2

d=3 (0,1, 2) ~~fixed Hamming weight~~ fixed digit sum = k

"su(d) Dicke states"

Example with d=3: $\frac{1}{\sqrt{6}}\left(|012\rangle + |021\rangle + |102\rangle + |120\rangle + |201\rangle + |210\rangle\right)$

N, Raveh 2301.04989; Raveh, N 2308.08392 ; N, Ravanini, Raveh 2402.03233 ; Raveh, N 2408.04729;
Kerzner, Galeazzi, N 2507.13308

Outline

I. Gray codes

II. Quantum computers & quantum state preparation

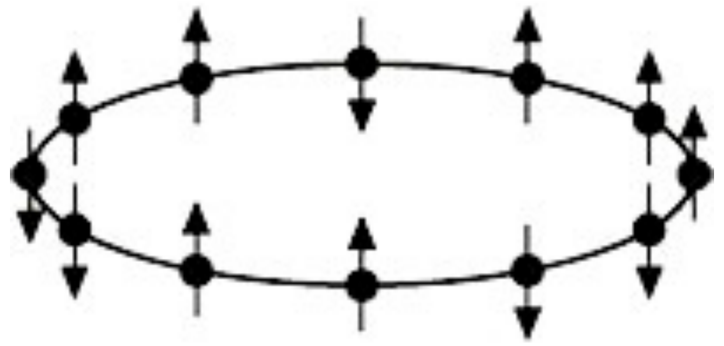
III. Bethe state preparation

- spin 1/2
- spin s
- nesting

new

IV. Conclusions

Bethe states



spin-1/2 = qubits

Ex: eigenstates $|B_k^n\rangle$ of periodic spin-1/2 Heisenberg (XXX) spin chain with n sites

Hamiltonian $$H = -\frac{1}{2}\sum_{i=1}^{n}(\vec{\sigma}_i \cdot \vec{\sigma}_{i+1} - I)$$

$\vec{\sigma} = (\sigma^x, \sigma^y, \sigma^z)$ Pauli spin matrices

quantum integrable

- $\mathcal{O}(n)$ local conserved quantities
- expressions for exact eigenstates

not true for generic Hamiltonians

coordinate Bethe ansatz:

Bethe 1931

k = # flipped spins = Hamming weight

$$|B_k^n\rangle = \sum_{1\le x_1<x_2<\ldots<x_k\le n} f(x_1,x_2,\ldots,x_k)\,\sigma_{x_1}^-\,\sigma_{x_2}^-\,\ldots\sigma_{x_k}^-\,|0\rangle^{\otimes n}$$

$$f(x_1,x_2,\ldots,x_k) = \sum_{P\in S_k} A_P(\vec{\mathbb{k}})\,e^{i\sum_{j=1}^k \mathbb{k}_{Pj}x_j}$$

$$A_P(\vec{\mathbb{k}}) = \prod_{j<l}\left(1-\frac{(e^{i\mathbb{k}_{Pj}}-1)(e^{i\mathbb{k}_{Pl}}-1)}{e^{i\mathbb{k}_{Pj}}-e^{i\mathbb{k}_{Pl}}}\right)$$

S_k : group of permutations of $1,\ldots,k$

$$e^{i\mathbb{k}_j} = \frac{u_j+\frac{i}{2}}{u_j-\frac{i}{2}}$$

"Bethe roots" $\{u_1,u_2,\ldots,u_k\}$

solutions of "Bethe equations" :

$$\left(\frac{u_j+\frac{i}{2}}{u_j-\frac{i}{2}}\right)^n = \prod_{\substack{l\neq j\\ l=1}}^{k}\frac{u_j-u_l+i}{u_j-u_l-i}\,,\qquad j=1,2,\ldots k$$

$$H\,|B_k^n\rangle = E\,|B_k^n\rangle \qquad E=\sum_{l=1}^k\frac{1}{u_l^2+\frac14}$$

Faddeev & Takhtadzhyan1981

$$\mathbb{S}^z\,|B_k^n\rangle = \left(\frac{n}{2}-k\right)|B_k^n\rangle \qquad \mathbb{S}^z=\sum_{i=1}^n S_i^z$$

$$\mathbb{S}^+\,|B_k^n\rangle = 0$$

SU(2) highest-weight states

$$\Rightarrow \qquad s=\frac{n}{2}-k$$

total spin

SU(2) symmetry $\Rightarrow \begin{cases} \text{degeneracy} & 2s+1 = n-2k+1 \\ s \geq 0 & k \leq \dfrac{n}{2} \end{cases}$ $\qquad s = \dfrac{n}{2} - k$

Example: $n = 4$ $\qquad k = 0, 1, 2$

k	u_j	E	degeneracy
0	-	0	5
1	$\frac{1}{2}$	2	3
1	$-\frac{1}{2}$	2	3
1	0	4	3
2	$\pm\frac{i}{2}$	2	1
2	$\pm\frac{1}{2\sqrt{3}}$	6	1

total: $16 = 2^4 = 2^n$ "complete" ✓

Assume Bethe roots $\{u_1, u_2, \ldots, u_k\}$ are known.

(Solving Bethe equations is generally hard, but we'll come back to this later.)

How to prepare the corresponding Bethe state $|B_k^n\rangle$?

Van Dyke, Barron, Mayhall, Barnes, Economou 2103.13388

Van Dyke, Barron, Barnes, Economou, N 2109.05607

Li, Okyay, N 2201.03021

Sopena, Hunter Gordon, García-Martín, Sierra, López 2202.04673

Ruiz, Sopena, Hunter Gordon, Sierra, López 2309.14430

Ruiz, Sopena, López, Sierra, Pozsgay 2411.02519

algebraic Bethe ansatz

Raveh, N 2403.03283

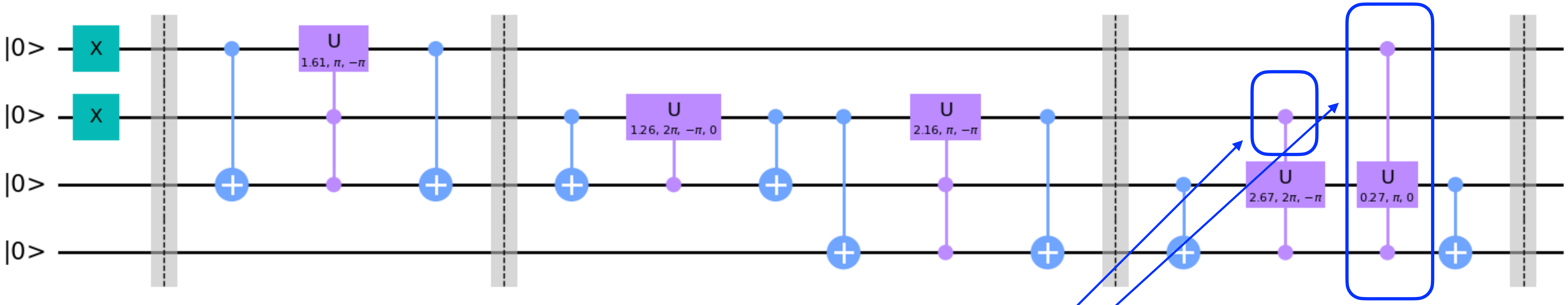
recursion (~ Dicke states)

Sahu, Vidal 2412.00923

Yeo, Kim, Sohn, Jeong 2501.02929

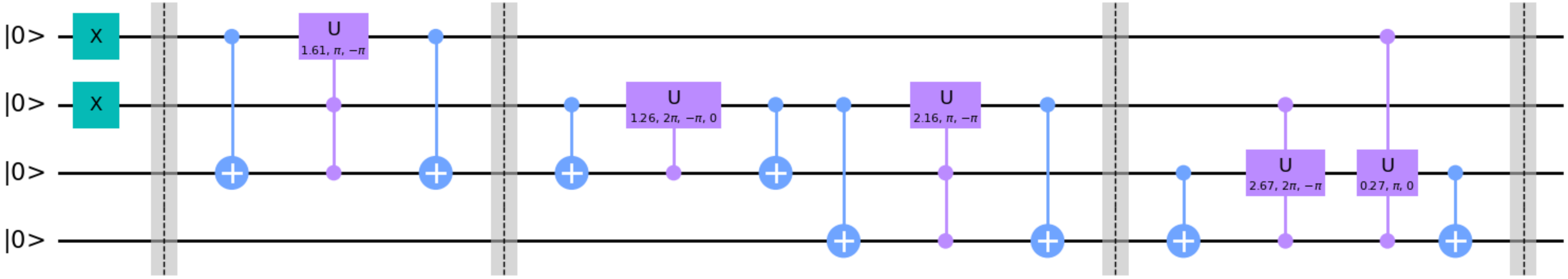
⋮

Ex: (n,k) = (4,2)



Same as circuit for Dicke state $|D_2^4\rangle$ except for angles & these gates

Ex: (n,k) = (4,2)



circuit size = $\mathcal{O}\left(\binom{n}{k}\right)$ For k=n/2 $\binom{n}{n/2} \sim \frac{2^n}{\sqrt{\pi n/2}}$ 🙁

- "Solve" Bethe equations on a quantum computer: Raveh, N 2404.18244

  variational method (VQE) using Bethe states as trial states, treating Bethe roots as variational parameters

- Algorithm can prepare *any* state with fixed Hamming weight k (not just Bethe states), as long as coefficients of computational basis states are known

Other algorithms for preparing *any* state with fixed Hamming weight:

Mao, Tian, Sun 2404.05147

Farias, Maciel, Camilo, Lin, Ramos-Calderer, Aolita 2405.20408 using Gray code

Li, Tian, He, Sun 2508.14470

Luo, Li 2508.17197

⋮

Using Gray code:

Ex: (n,k) = (3,1)

3 bits

Hamming weight 1

list of possible bitstrings:

100
001
010

"Gray property": only *two* bits *swap* at each step

Using Gray code:

Ex: (n,k) = (3,1)   list of possible bitstrings:

3 bits
Hamming weight 1

100
001
010

"Gray property": only *two* bits *swap* at each step

$\therefore$ Hamming weight is preserved

Using Gray code:

Ex: $(n,k) = (3,1)$ list of possible bitstrings:

3 bits

Hamming weight 1

100
001
010

"Gray property": only *two* bits *swap* at each step

$\therefore$ Hamming weight is preserved

Using Gray code:

Ex: (n,k) = (3,1) list of possible bitstrings: 100 Ehrlich 1970's

3 bits
Hamming weight 1

001
010

"Gray property": only *two* bits *swap* at each step

$\therefore$ Hamming weight is preserved

Farias, Maciel, Camilo, Lin, Ramos-Calderer, Aolita 2405.20408

Build quantum state by corresponding sequence of controlled 2-qubit rotations:

$|100\rangle$

Using Gray code:

Ex: (n,k) = (3,1) list of possible bitstrings:

100
001
010

Ehrlich 1970's

3 bits
Hamming weight 1

"Gray property": only *two* bits *swap* at each step

∴ Hamming weight is preserved

Farias, Maciel, Camilo, Lin, Ramos-Calderer, Aolita 2405.20408

Build quantum state by corresponding sequence of controlled 2-qubit rotations:

$$|100\rangle \mapsto \cos\theta_1 |100\rangle + \sin\theta_1 |001\rangle$$

Using Gray code:

Ex: (n,k) = (3,1)   list of possible bitstrings:

100
001
010

Ehrlich 1970's

3 bits
Hamming weight 1

"Gray property": only *two* bits *swap* at each step

$\therefore$ Hamming weight is preserved

Farias, Maciel, Camilo, Lin, Ramos-Calderer, Aolita 2405.20408

Build quantum state by corresponding sequence of controlled 2-qubit rotations:

$$|100\rangle \mapsto \cos\theta_1|100\rangle + \sin\theta_1|001\rangle$$

$$\mapsto \cos\theta_1|100\rangle + \sin\theta_1\left(\cos\theta_2|001\rangle + \sin\theta_2|010\rangle\right)$$

Using Gray code:

Ex: (n,k) = (3,1)   list of possible bitstrings: 100   Ehrlich 1970's
001
010

3 bits
Hamming weight 1

"Gray property": only *two* bits *swap* at each step

$\therefore$ Hamming weight is preserved

Farias, Maciel, Camilo, Lin, Ramos-Calderer, Aolita 2405.20408

Build quantum state by corresponding sequence of controlled 2-qubit rotations:

$$|100\rangle \mapsto \cos\theta_1|100\rangle + \sin\theta_1|001\rangle$$

$$\mapsto \cos\theta_1|100\rangle + \sin\theta_1\left(\cos\theta_2|001\rangle + \sin\theta_2|010\rangle\right) = a_0|100\rangle + a_1|001\rangle + a_2|010\rangle$$

Same circuit size & depth = $\mathcal{O}\left(\binom{n}{k}\right)$

… but can generalize this idea to prepare more complicated Bethe states!

Outline

I. Gray codes

II. Quantum computers & quantum state preparation

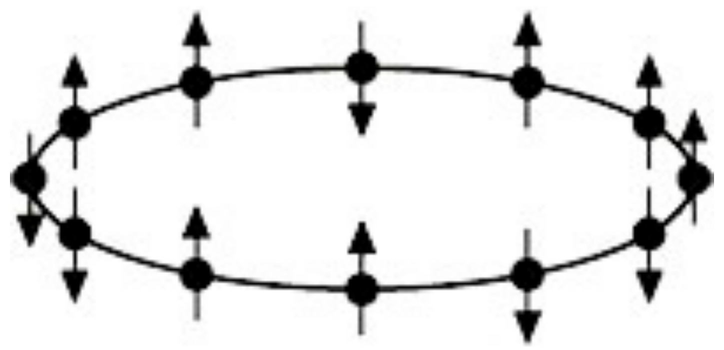
III. Bethe state preparation

- spin 1/2
- spin s
- nesting

new

IV. Conclusions

integrable spin-s XXX chain



spin-s = qudits $\qquad d = 2s+1$

Hamiltonian $\qquad H(s) = \sum_{i=1}^{n} h(x_i, s) \qquad x_i = \vec{S}_i \cdot \vec{S}_{i+1} \qquad \vec{S}_i = \mathbb{I} \otimes \ldots \otimes \mathbb{I} \otimes \overset{\overset{i}{\downarrow}}{\vec{S}} \otimes \mathbb{I} \otimes \ldots \otimes \mathbb{I}$ $\qquad$ Babujian 1983

spin-s

$$h(x,s) = 2 \sum_{i=1}^{2s} \left(\sum_{j=1}^{i} \frac{1}{j} \right) \prod_{\substack{l=0 \\ \neq i}}^{2s} \frac{x - x_l}{x_i - x_l} + \mathrm{const}$$

polynomial in x of degree 2s

$$x_l = \tfrac{1}{2}\left[l(l+1) - 2s(s+1) \right]$$

s	$h(x,s)$
$1/2$	$-\frac{1}{2} + 2x$
1	$\frac{1}{2}x - \frac{1}{2}x^2$
$3/2$	$-\frac{3}{4} - \frac{1}{8}x + \frac{1}{27}x^2 + \frac{2}{27}x^3$
2	$-\frac{1}{2} + \frac{13}{24}x + \frac{43}{432}x^2 - \frac{5}{216}x^3 - \frac{1}{144}x^4$

coordinate Bethe ansatz: Crampé, Ragoucy, Alonzi 2011 known

$$|B^{(s)}_{n,k}\rangle = \sum_{1\le x_1\le x_2\le\ldots\le x_k\le n} \boxed{f(x_1,x_2,\ldots,x_k)}\, e^-{}_{x_1}\, e^-{}_{x_2}\, \ldots e^-{}_{x_k}\, |0\rangle^{\otimes n}$$

k = # e— = digit sum

$$e^- = \sum_{j=0}^{2s-1} \sqrt{\frac{2s-j}{j+1}}\, |j+1\rangle\langle j|$$

can act more than once on same site!

$$f(x_1,x_2,\ldots,x_k) = \sum_{P\in S_k} A_P(\vec{\mathbb{k}})\, e^{i\sum_{j=1}^k \mathbb{k}_{Pj}x_j}$$

$$A_P(\vec{\mathbb{k}}) = \prod_{j<l}\left(1-\frac{1}{2s}\frac{(e^{i\mathbb{k}_{Pj}}-1)(e^{i\mathbb{k}_{Pl}}-1)}{e^{i\mathbb{k}_{Pj}}-e^{i\mathbb{k}_{Pl}}}\right)$$

"Bethe roots" $\{u_1,u_2,\ldots,u_k\}$

$$e^{i\mathbb{k}_j} = \frac{u_j+is}{u_j-is}$$

solutions of "Bethe equations" :

$$\left(\frac{u_j+is}{u_j-is}\right)^n = \prod_{\substack{l=1\\ \neq j}}^{k} \frac{u_j-u_l+i}{u_j-u_l-i}, \qquad j=1,\ldots,k$$

$$H(s)\,|B^{(s)}_{n,k}\rangle = E\,|B^{(s)}_{n,k}\rangle \qquad E = -\sum_{j=1}^{k}\frac{2s}{u_j^2+s^2}$$

$$\mathbb{S}^z\,|B^{(s)}_{n,k}\rangle = (sn-k)\,|B^{(s)}_{n,k}\rangle \qquad \mathbb{S}^z = \sum_{i=1}^{n} S^z_i$$

$$\mathbb{S}^+\,|B^n_k\rangle = 0$$

SU(2) highest-weight states

Assume Bethe roots $\{u_1, u_2, \ldots, u_k\}$ are known.

How to prepare the corresponding Bethe state $|B_{n,k}^{(s)}\rangle$?

Zare Harofteh, N 2601.14513

fixed digit sum $\Rightarrow$

Gray code for bounded integer compositions

Walsh 2000

Ex: (n,k,s) = (3,3,1)

3 dits each with 3 levels (0,1, 2)

digit sum = 3

list of possible ditstrings:

012
021
120
111
102
201
210

"Gray property": only *two* dits change at each step

one dit increases by 1, other dit decreases by 1 $\therefore$ digit sum is preserved

Assume Bethe roots $\{u_1, u_2, \ldots, u_k\}$ are known.

How to prepare the corresponding Bethe state $|B^{(s)}_{n,k}\rangle$?

Zare Harofteh, N 2601.14513

fixed digit sum $\Rightarrow$

Gray code for bounded integer compositions

Walsh 2000

Ex: (n,k,s) = (3,3,1)

3 dits each with 3 levels (0,1,2)

digit sum = 3

list of possible ditstrings:

012
021
120
111
102
201
210

"Gray property": only *two* dits change at each step

one dit increases by 1, other dit decreases by 1 $\therefore$ digit sum is preserved

Assume Bethe roots $\{u_1, u_2, \ldots, u_k\}$ are known.

How to prepare the corresponding Bethe state $|B_{n,k}^{(s)}\rangle$?

Zare Harofteh, N 2601.14513

fixed digit sum $\Rightarrow$

Gray code for bounded integer compositions

Walsh 2000

Ex: (n,k,s) = (3,3,1)

3 dits each with 3 levels (0,1,2)

digit sum = 3

list of possible ditstrings:

012
021
120
111
102
201
210

"Gray property": only *two* dits change at each step

one dit increases by 1, other dit decreases by 1 $\therefore$ digit sum is preserved

Assume Bethe roots $\{u_1, u_2, \ldots, u_k\}$ are known.

How to prepare the corresponding Bethe state $|B_{n,k}^{(s)}\rangle$?

Zare Harofteh, N 2601.14513

fixed digit sum $\Rightarrow$

Gray code for bounded integer compositions

Walsh 2000

Ex: (n,k,s) = (3,3,1)

3 dits each with 3 levels (0,1,2)

digit sum = 3

list of possible ditstrings:

012
021
120
111
102
201
210

"Gray property": only *two* dits change at each step

one dit increases by 1, other dit decreases by 1 $\therefore$ digit sum is preserved

Assume Bethe roots $\{u_1, u_2, \ldots, u_k\}$ are known.

How to prepare the corresponding Bethe state $|B^{(s)}_{n,k}\rangle$?

Zare Harofteh, N 2601.14513

fixed digit sum $\Rightarrow$

Gray code for bounded integer compositions

Walsh 2000

Ex: (n,k,s) = (3,3,1)

3 dits each with 3 levels (0,1,2)

digit sum = 3

list of possible ditstrings:

012
021
120
111
102
201
210

"Gray property": only *two* dits change at each step

one dit increases by 1, other dit decreases by 1 $\therefore$ digit sum is preserved

Assume Bethe roots $\{u_1, u_2, \ldots, u_k\}$ are known.

How to prepare the corresponding Bethe state $|B_{n,k}^{(s)}\rangle$?

Zare Harofteh, N 2601.14513

fixed digit sum $\Rightarrow$

Gray code for bounded integer compositions

Walsh 2000

Ex: (n,k,s) = (3,3,1)

3 dits each with 3 levels (0,1, 2)

digit sum = 3

list of possible ditstrings:

012
021
120
111
102
201
210

"Gray property": only *two* dits change at each step

one dit increases by 1, other dit decreases by 1 $\therefore$ digit sum is preserved

Assume Bethe roots $\{u_1, u_2, \ldots, u_k\}$ are known.

How to prepare the corresponding Bethe state $|B^{(s)}_{n,k}\rangle$?

Zare Harofteh, N 2601.14513

fixed digit sum $\Rightarrow$

Gray code for bounded integer compositions

Walsh 2000

Ex: (n,k,s) = (3,3,1)

list of possible ditstrings:

3 dits each with 3 levels (0,1, 2)

digit sum = 3

012
021
120
111
102
201
210

"Gray property": only *two* dits change at each step

one dit increases by 1, other dit decreases by 1 $\therefore$ digit sum is preserved

i j

$$\vec{m} \to \vec{m} + \hat{e}_i - \hat{e}_j$$

Assume Bethe roots $\{u_1, u_2, \ldots, u_k\}$ are known.

How to prepare the corresponding Bethe state $|B^{(s)}_{n,k}\rangle$?

Zare Harofteh, N 2601.14513

fixed digit sum $\Rightarrow$

Gray code for bounded integer compositions

Walsh 2000

Ex: (n,k,s) = (3,3,1)

3 dits each with 3 levels (0,1,2)

digit sum = 3

list of possible ditstrings:

012
021
120
111
102
201
210

"Gray property": only *two* dits change at each step

one dit increases by 1, other dit decreases by 1 $\therefore$ digit sum is preserved

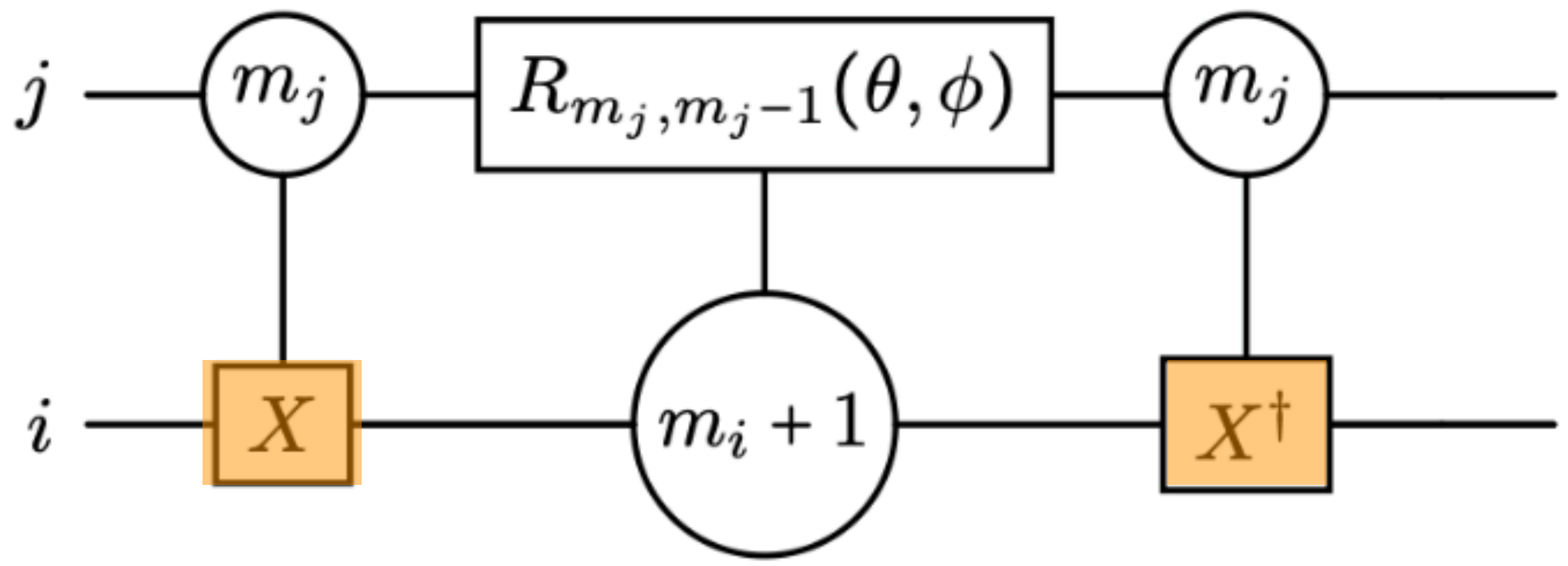
i j

$\vec{m} \to \vec{m} + \hat{e}_i - \hat{e}_j$

Build quantum state with fixed digit sum by corresponding sequence of controlled 2-qudit rotations

$$|\vec{m}\rangle \rightarrow \cos(\theta)\,|\vec{m}\rangle + e^{i\phi}\sin(\theta)\,|\vec{m'}\rangle \qquad \vec{m'} = \vec{m} + \hat{e}_i - \hat{e}_j$$

Gray gate

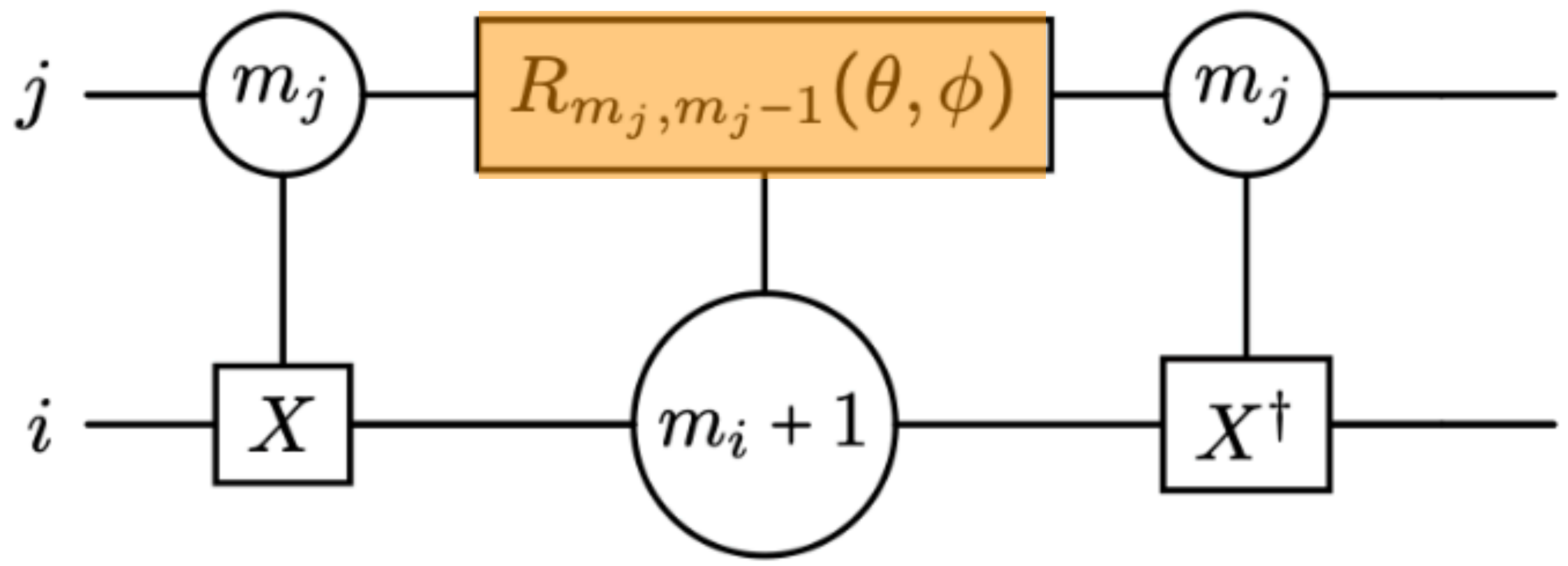


$$\mathrm{X}\,|\mu\rangle = |\mu+1\rangle\,, \quad \mathrm{X}^\dagger\,|\mu\rangle = |\mu-1\rangle \qquad \text{modulo } d = 2s+1$$

Build quantum state with fixed digit sum by corresponding sequence of controlled 2-qudit rotations

$$|\vec{m}\rangle \rightarrow \cos(\theta)\,|\vec{m}\rangle + e^{i\phi}\sin(\theta)\,|\vec{m'}\rangle \qquad \vec{m'} = \vec{m} + \hat{e}_i - \hat{e}_j$$

Gray gate



$$\mathrm{X}\,|\mu\rangle = |\mu+1\rangle\,, \quad \mathrm{X}^\dagger\,|\mu\rangle = |\mu-1\rangle \qquad \text{modulo } d = 2s+1$$

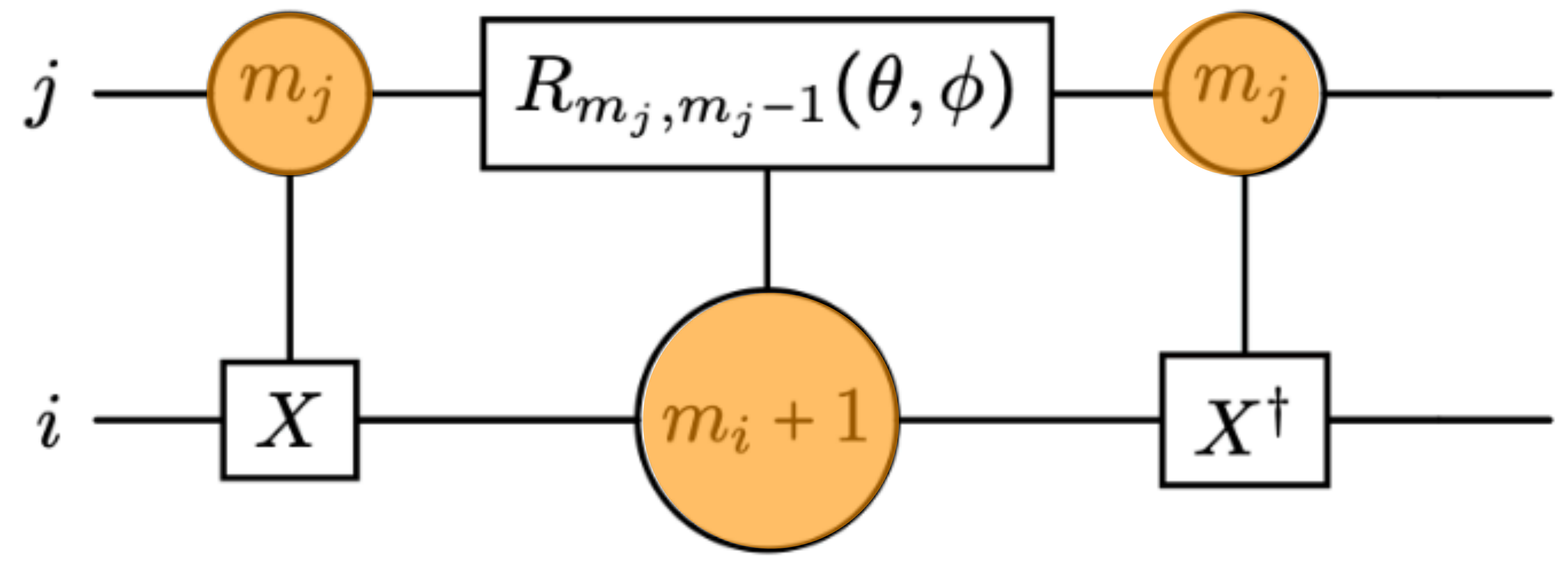
$$R_{m_j,m_j-1}(\theta,\phi) = \cos(\theta)|m_j\rangle\langle m_j| - \sin(\theta)|m_j\rangle\langle m_j-1|$$
$$+\, e^{i\phi}\sin(\theta)|m_j-1\rangle\langle m_j| + e^{i\phi}\cos(\theta)|m_j-1\rangle\langle m_j-1| + \sum_{\mu\neq m_j\,,m_j-1} |\mu\rangle\langle\mu|$$

$$= \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & e^{i\phi}\cos\theta & e^{i\phi}\sin\theta & & & \\ & & & -\sin\theta & \cos\theta & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{pmatrix}_{d\times d}$$

Build quantum state with fixed digit sum by corresponding sequence of controlled 2-qudit rotations

$$|\vec{m}\rangle \rightarrow \cos(\theta)\,|\vec{m}\rangle + e^{i\phi}\sin(\theta)\,|\vec{m'}\rangle \qquad \vec{m'} = \vec{m} + \hat{e}_i - \hat{e}_j$$

Gray gate



controls

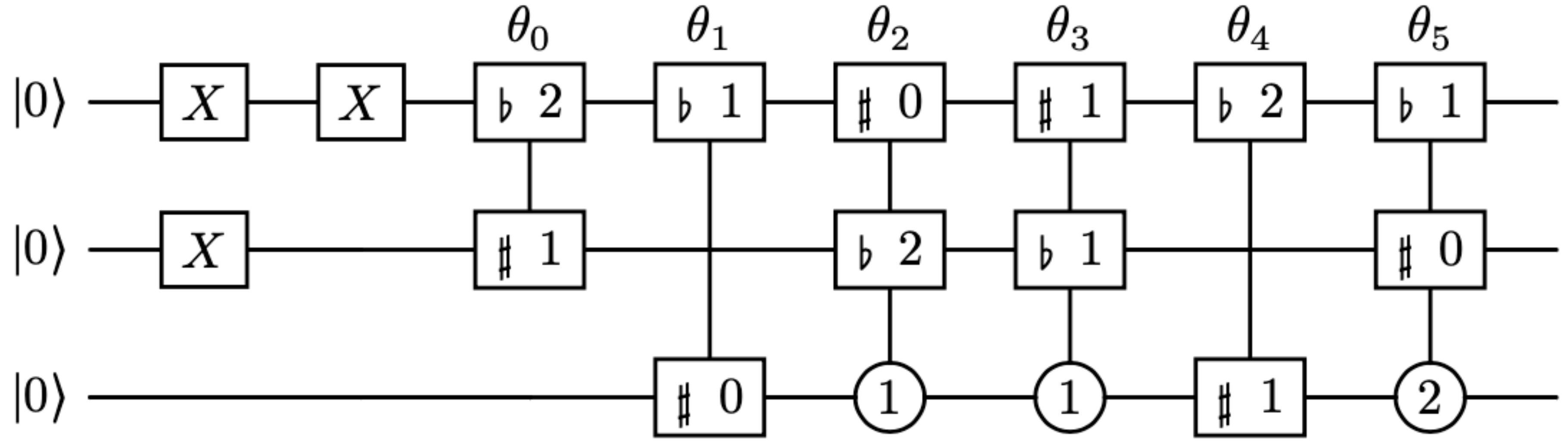
$$\mathrm{X}\,|\mu\rangle = |\mu+1\rangle\,, \quad \mathrm{X}^\dagger\,|\mu\rangle = |\mu-1\rangle \qquad \text{modulo } d = 2s+1$$

$$R_{m_j,m_j-1}(\theta,\phi) = \cos(\theta)|m_j\rangle\langle m_j| - \sin(\theta)|m_j\rangle\langle m_j-1| + e^{i\phi}\sin(\theta)|m_j-1\rangle\langle m_j| + e^{i\phi}\cos(\theta)|m_j-1\rangle\langle m_j-1| + \sum_{\mu\neq m_j\,,m_j-1}|\mu\rangle\langle\mu|$$

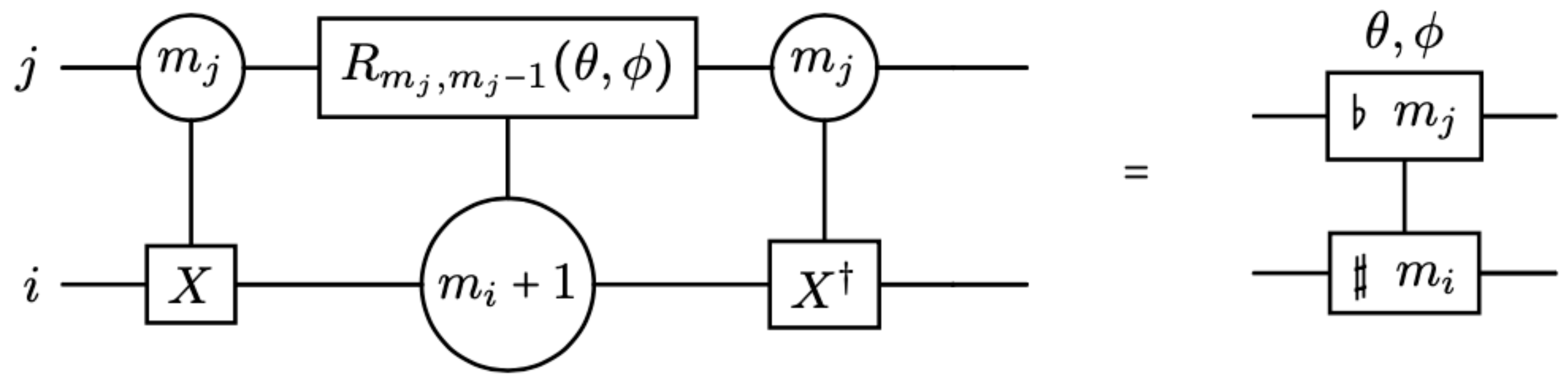
$$= \begin{pmatrix} 1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1 & & & & & \\ & & & e^{i\phi}\cos\theta & e^{i\phi}\sin\theta & & & \\ & & & -\sin\theta & \cos\theta & & & \\ & & & & & 1 & & \\ & & & & & & \ddots & \\ & & & & & & & 1 \end{pmatrix}_{d\times d}$$

Example: (n,k,s) = (3,3,1)

$$a_0|012\rangle + a_1|021\rangle + a_2|120\rangle + a_3|111\rangle + a_4|102\rangle + a_5|201\rangle + a_6|210\rangle$$

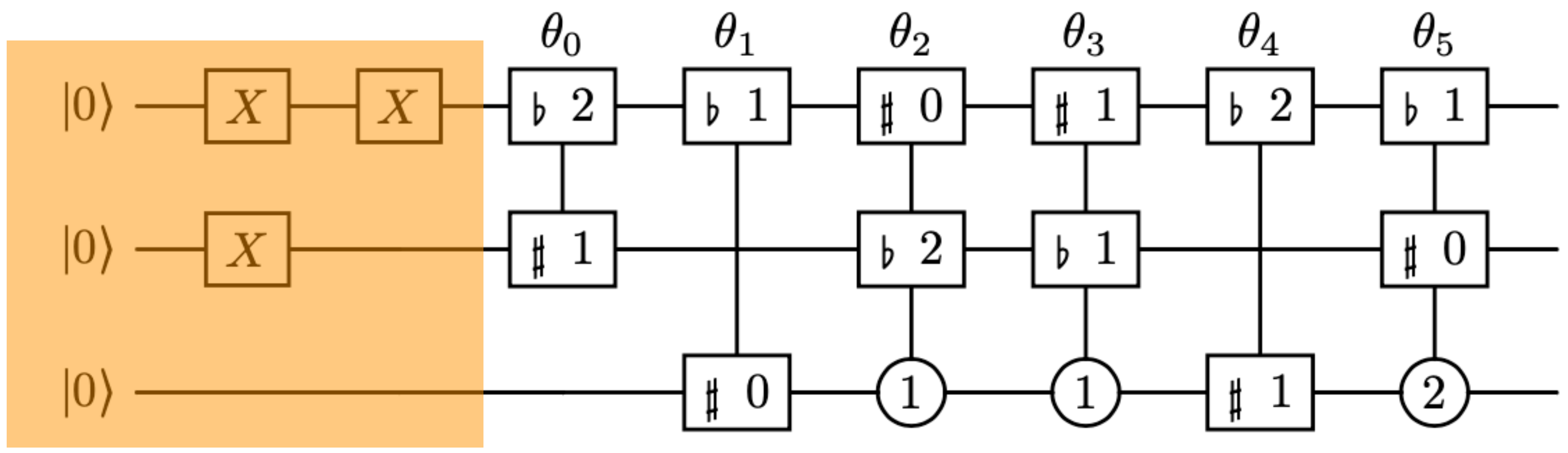


Gray gate

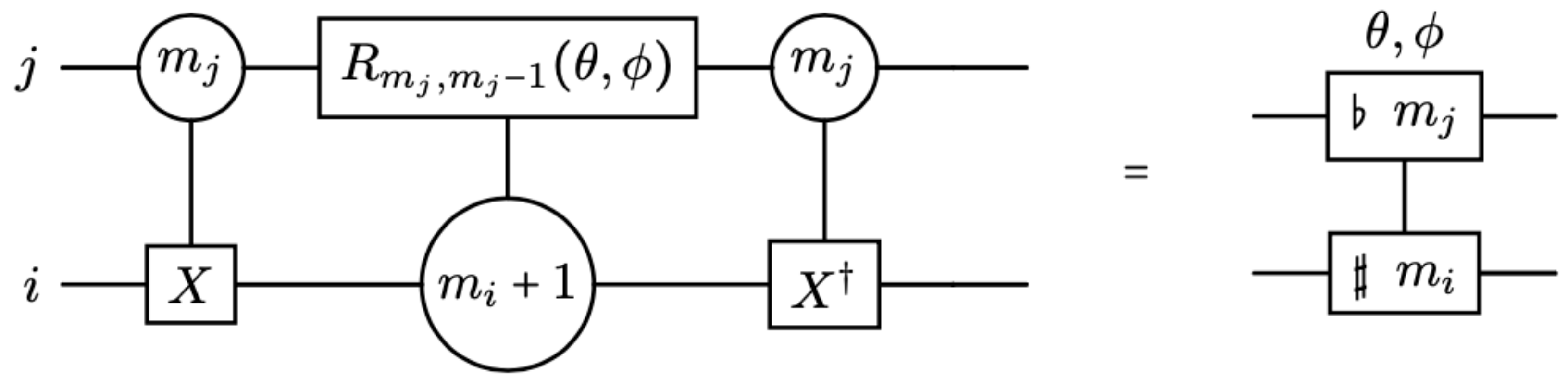


Example: (n,k,s) = (3,3,1)

$$a_0|012\rangle + a_1|021\rangle + a_2|120\rangle + a_3|111\rangle + a_4|102\rangle + a_5|201\rangle + a_6|210\rangle$$

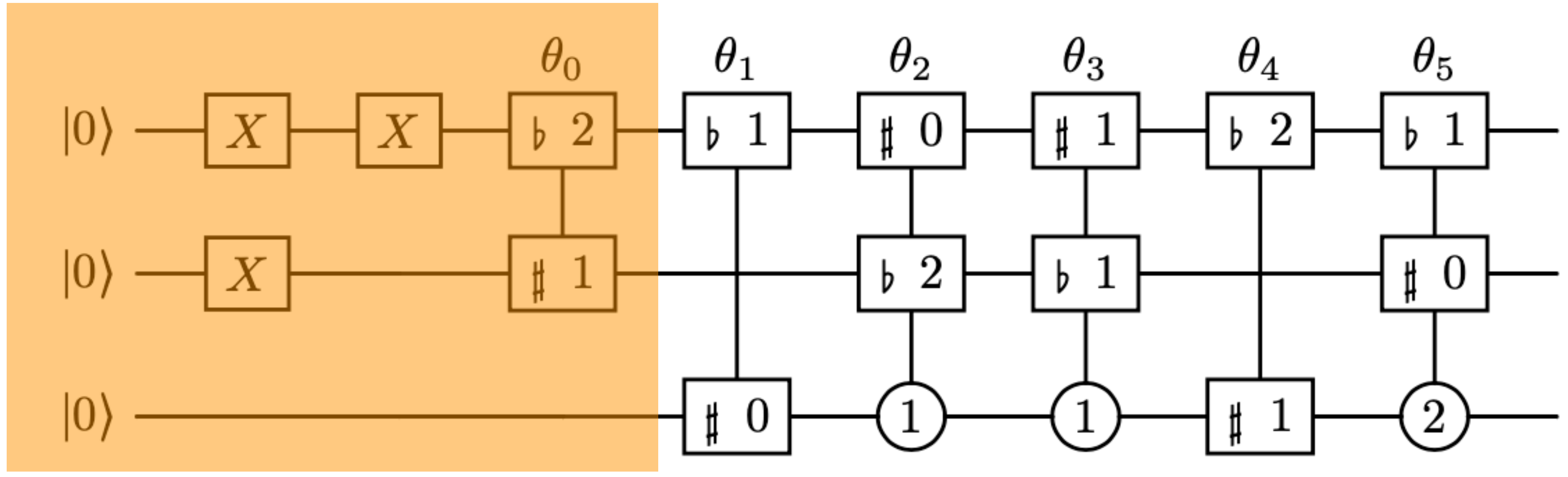


Gray gate

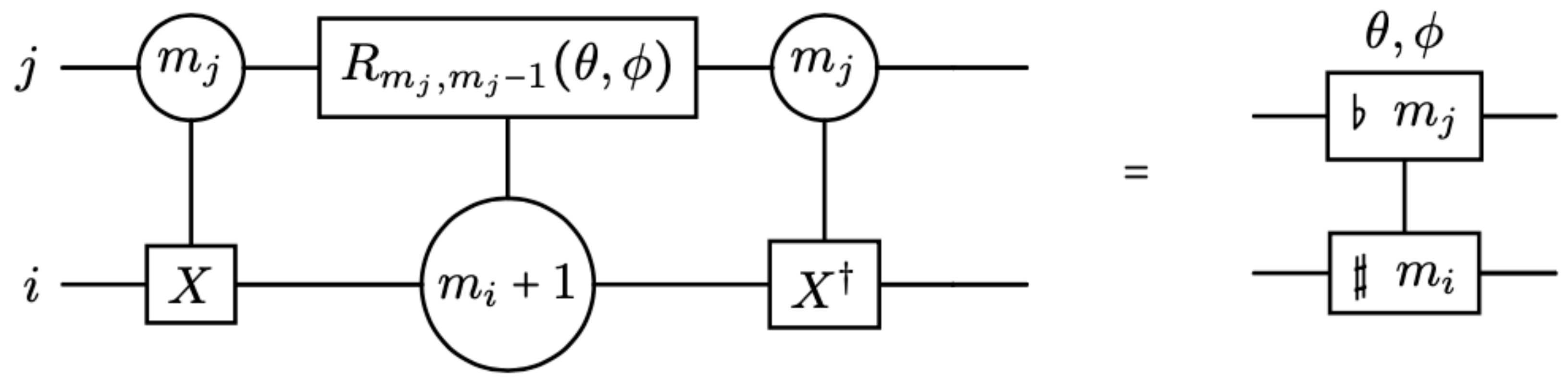


Example: (n,k,s) = (3,3,1)

$$a_0|012\rangle + a_1|021\rangle + a_2|120\rangle + a_3|111\rangle + a_4|102\rangle + a_5|201\rangle + a_6|210\rangle$$

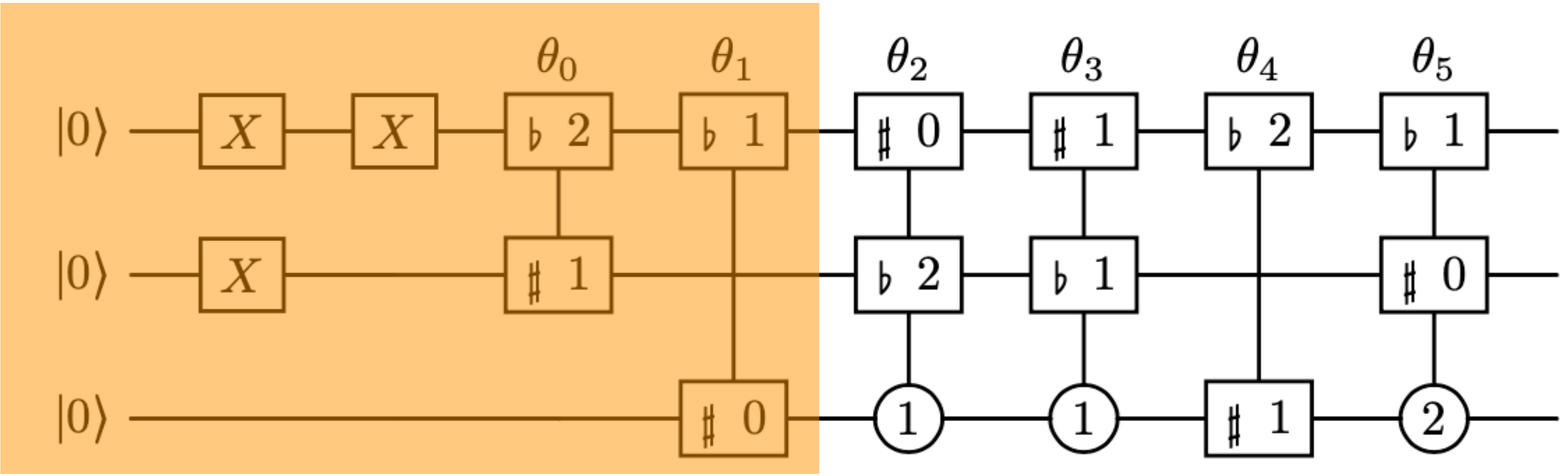


Gray gate

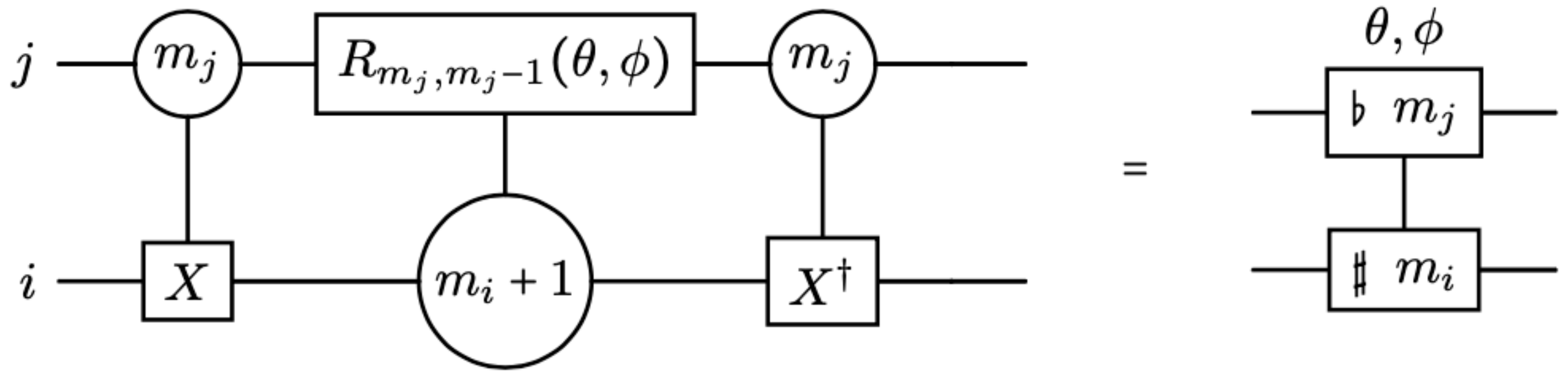


Example: (n,k,s) = (3,3,1)

$$a_0|012\rangle + a_1|021\rangle + a_2|120\rangle + a_3|111\rangle + a_4|102\rangle + a_5|201\rangle + a_6|210\rangle$$

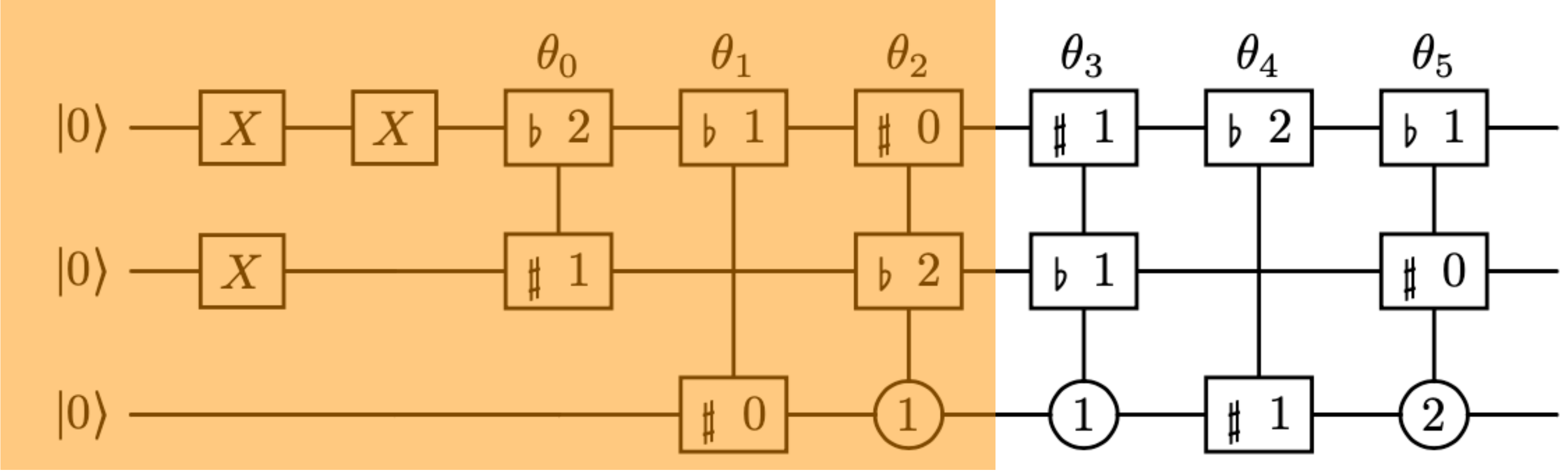


Gray gate

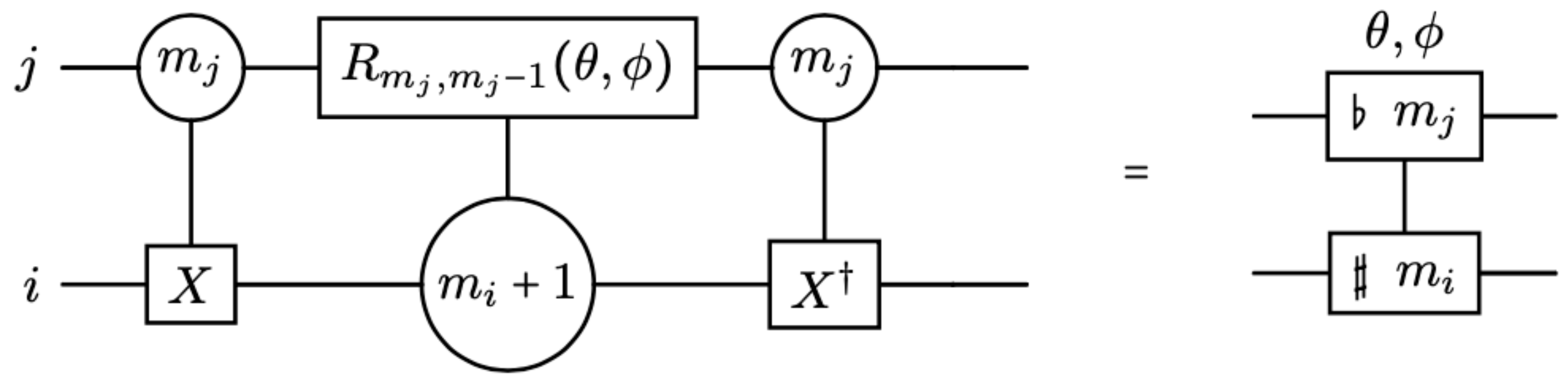


Example: (n,k,s) = (3,3,1)

$$a_0|012\rangle + a_1|021\rangle + a_2|120\rangle + a_3|111\rangle + a_4|102\rangle + a_5|201\rangle + a_6|210\rangle$$

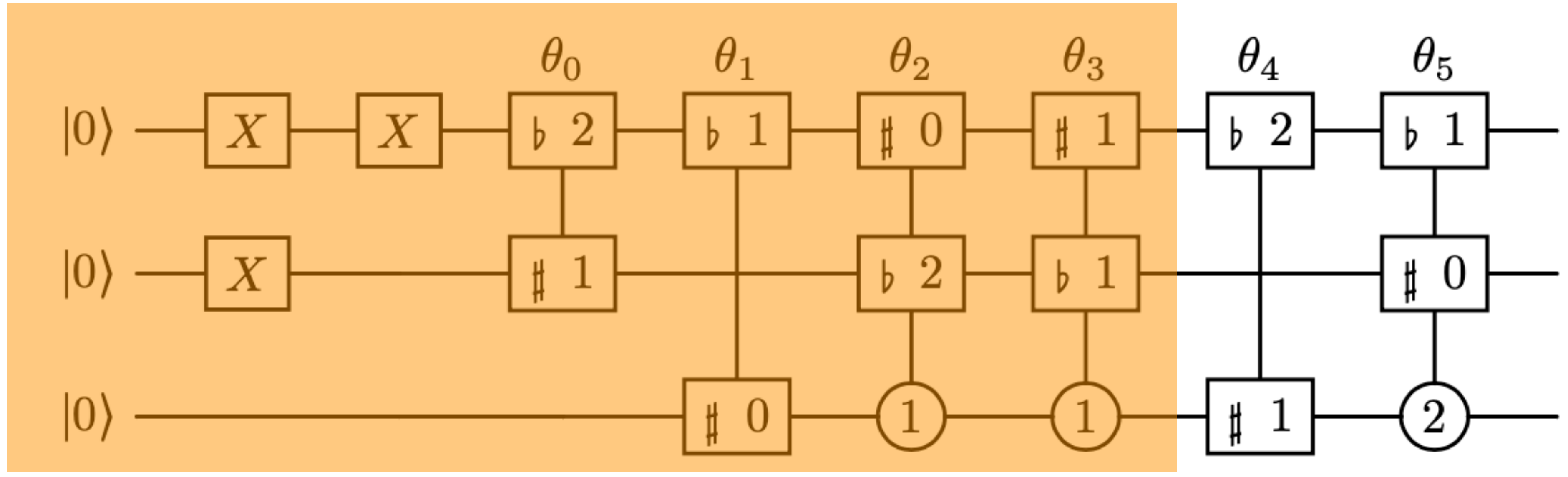


Gray gate

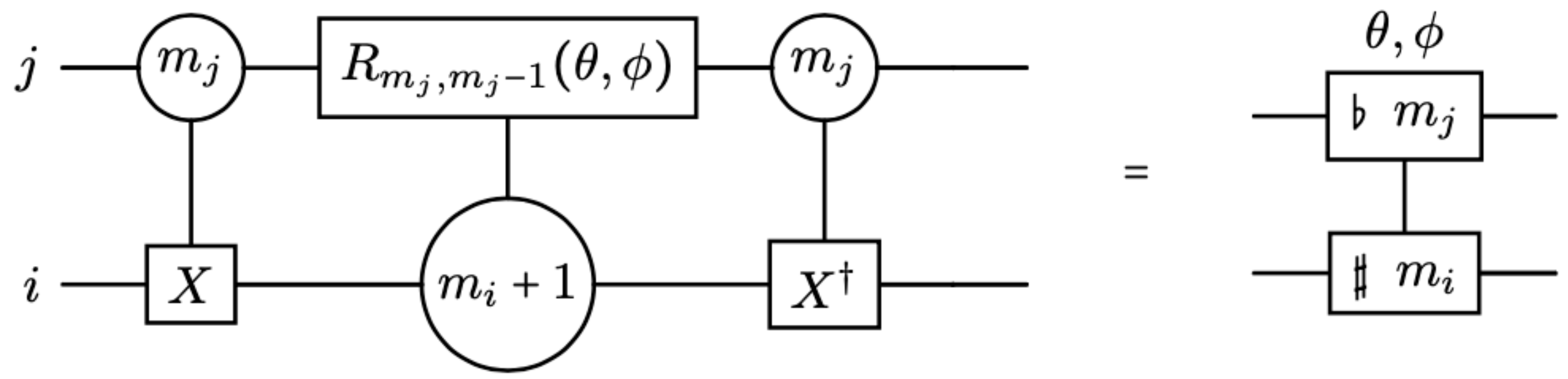


Example: (n,k,s) = (3,3,1)

$$a_0|012\rangle + a_1|021\rangle + a_2|120\rangle + a_3|111\rangle + a_4|102\rangle + a_5|201\rangle + a_6|210\rangle$$

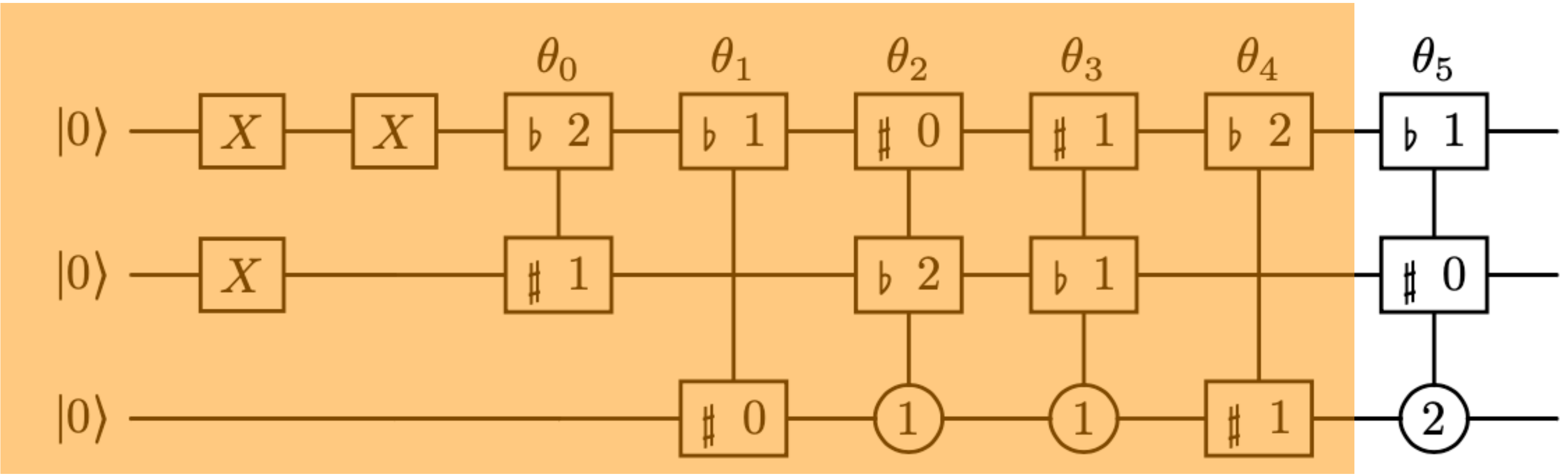


Gray gate

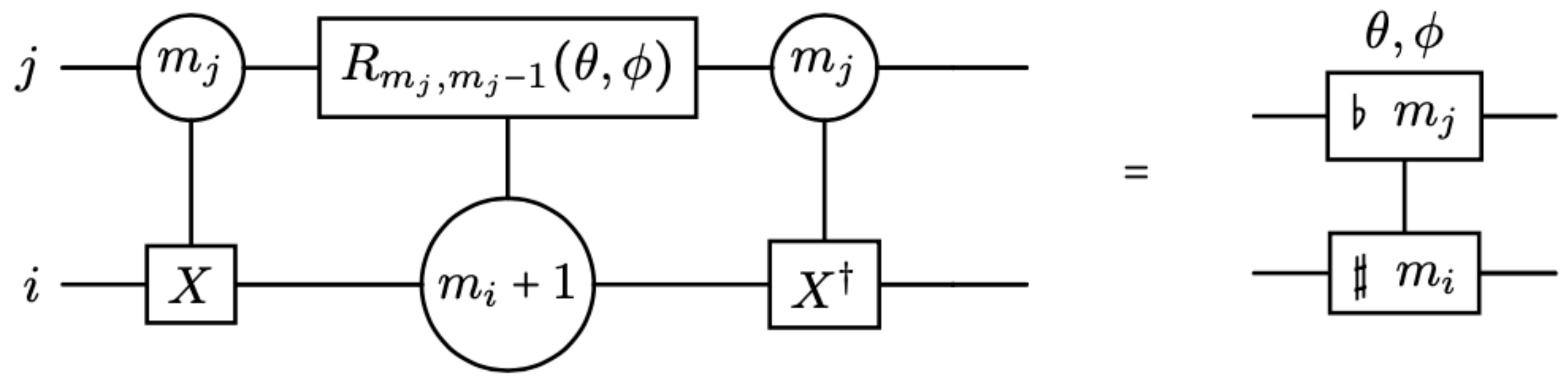


Example: (n,k,s) = (3,3,1)

$$a_0|012\rangle + a_1|021\rangle + a_2|120\rangle + a_3|111\rangle + a_4|102\rangle + a_5|201\rangle + a_6|210\rangle$$

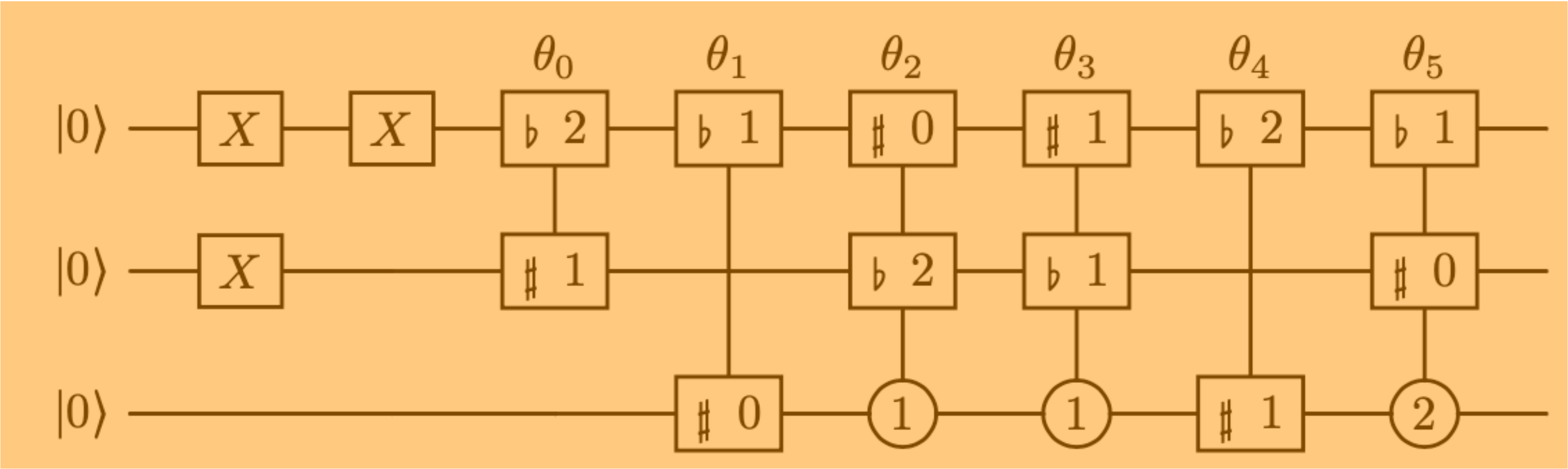


Gray gate

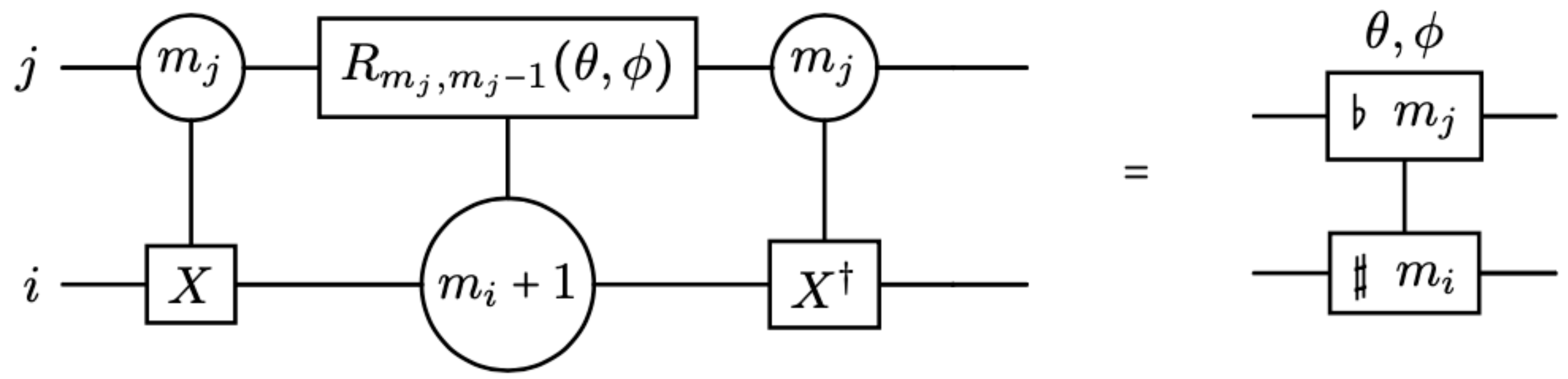


Example: (n,k,s) = (3,3,1)

$$a_0|012\rangle + a_1|021\rangle + a_2|120\rangle + a_3|111\rangle + a_4|102\rangle + a_5|201\rangle + a_6|210\rangle$$

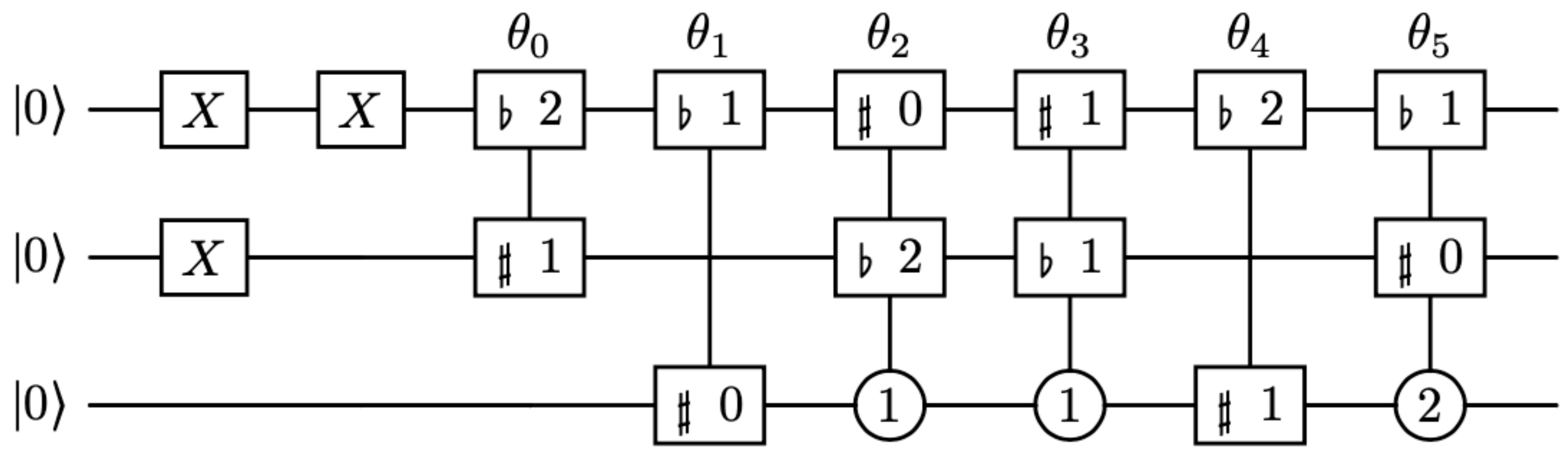


Gray gate

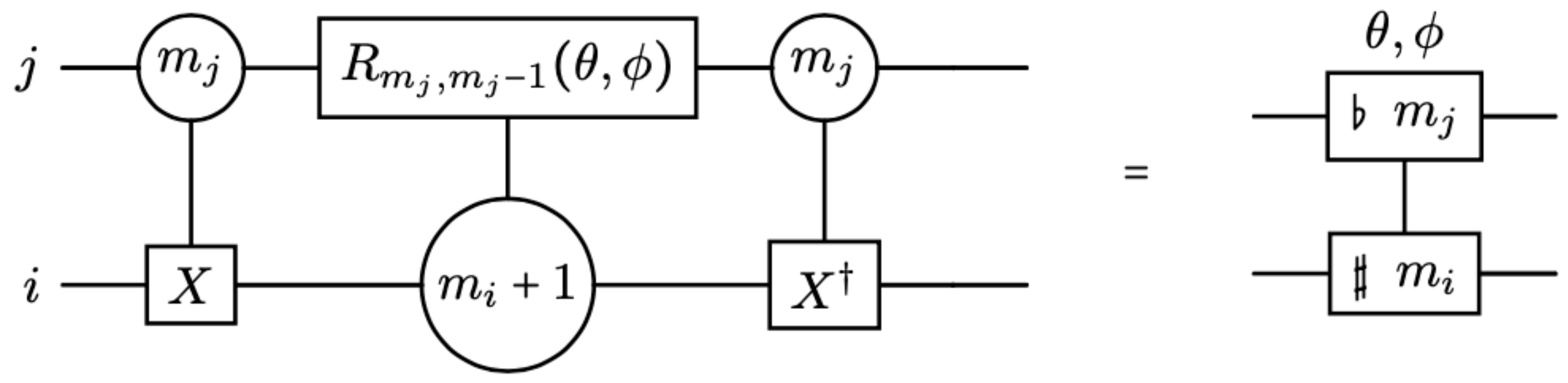


Example: (n,k,s) = (3,3,1)

$$a_0|012\rangle + a_1|021\rangle + a_2|120\rangle + a_3|111\rangle + a_4|102\rangle + a_5|201\rangle + a_6|210\rangle$$



Gray gate



circuit size & depth: $\mathcal{O}(n^k)$ for $n \gg k$

$\mathcal{O}((2s+1)^n)$ for worst case $k = sn$

Outline

I. Gray codes

II. Quantum computers & quantum state preparation

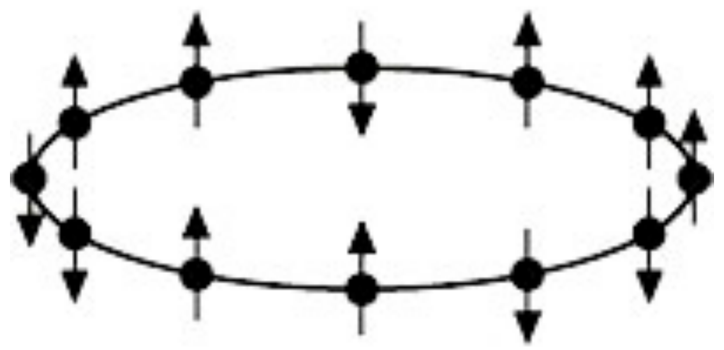
III. Bethe state preparation

- spin 1/2
- spin s
- nesting

new

IV. Conclusions

integrable SU(d)-invariant chain



d-level qudits

Hamiltonian $H = \sum_{i=1}^{n} (\mathcal{P}_{i,i+1} - \mathbb{I})$ Sutherland 1975

$\mathcal{P}$: permutation (swap) matrix on $\mathbb{C}^d \otimes \mathbb{C}^d$

nested coordinate Bethe ansatz for d>2 Yang 1967; Sutherland 1975; Gaudin 1983…

d=3:

“Bethe roots” $\{u_1, \ldots, u_{m_1}\}$, $\{v_1, \ldots, v_{m_2}\}$

solutions of “Bethe equations” :

$$\left(\frac{u_j - \frac{i}{2}}{u_j + \frac{i}{2}}\right)^n = \prod_{l \neq j}^{m_1} \frac{u_j - u_l - i}{u_j - u_l + i} \prod_{l=1}^{m_2} \frac{u_j - v_l + \frac{i}{2}}{u_j - v_l - \frac{i}{2}}, \qquad j = 1, \ldots, m_1$$

$$1 = \prod_{l=1}^{m_1} \frac{v_j - u_l + \frac{i}{2}}{v_j - u_l - \frac{i}{2}} \prod_{l \neq j}^{m_2} \frac{v_j - v_l - i}{v_j - v_l + i}, \qquad j = 1, \ldots, m_2$$

$$|\{u\};\{v\}\rangle = \sum_{1\le x_1<\ldots<x_{m_1}\le n}\ \sideset{}{'}\sum_{\alpha_1,\ldots,\alpha_{m_1}=1}^{2} f^{(\alpha_1,\ldots,\alpha_{m_1})}(x_1,\ldots,x_{m_1};\{u\},\{v\})\, e^{\alpha_1,0}_{x_1}\cdots e^{\alpha_{m_1},0}_{x_{m_1}}|0\rangle^{\otimes n}$$

$$\begin{aligned} f^{(\alpha_1,\ldots,\alpha_{m_1})}(x_1,\ldots,x_{m_1};\{u\},\{v\}) = \sum_{P\in\mathfrak{S}_{m_1}}\sum_{Q\in\mathfrak{S}_{m_2}} \Big\{&\varepsilon(P)\varepsilon(Q)A(u_{P(1)},\ldots,u_{P(m_1)})\,B(v_{Q(1)},\ldots,v_{Q(m_2)}) \\ &\times\prod_{j=1}^{m_1}\left(\frac{u_{P(j)}+\frac{i}{2}}{u_{P(j)}-\frac{i}{2}}\right)^{x_j}\prod_{l=1}^{m_2}\Phi(v_{Q(l)};u_{P(1)},\ldots,u_{P(y_l)})\Big\}\end{aligned}$$

$$\begin{aligned} A(u_1,\ldots,u_{m_1}) &= \prod_{1\le a<b\le m_1}(u_a-u_b+i)\,, \\ B(v_1,\ldots,v_{m_2}) &= \prod_{1\le a<b\le m_2}(v_a-v_b+i)\,, \\ \Phi(v;u_1,\ldots,u_y) &= \frac{1}{v-u_y-\frac{i}{2}}\prod_{l=1}^{y-1}\frac{v-u_l+\frac{i}{2}}{v-u_l-\frac{i}{2}}\end{aligned}$$

$$H\,|\{u\};\{v\}\rangle = E\,|\{u\};\{v\}\rangle \qquad E=-\sum_{j=1}^{m_1}\frac{1}{u_j^2+\frac{1}{4}}$$

SU(3) highest-weight states

Liashyk, Pakuliak, Ragoucy 2026

$$|\{u\};\{v\}\rangle = \sum_{1\le x_1<\ldots<x_{m_1}\le n} \sum_{\alpha_1,\ldots,\alpha_{m_1}=1}^{2}{}' \boxed{f^{(\alpha_1,\ldots,\alpha_{m_1})}(x_1,\ldots,x_{m_1};\{u\},\{v\})}\, e^{\alpha_1,0}_{x_1}\cdots e^{\alpha_{m_1},0}_{x_{m_1}}|0\rangle^{\otimes n}$$

known

$$f^{(\alpha_1,\ldots,\alpha_{m_1})}(x_1,\ldots,x_{m_1};\{u\},\{v\}) = \sum_{P\in\mathfrak{S}_{m_1}}\sum_{Q\in\mathfrak{S}_{m_2}} \Big\{\varepsilon(P)\varepsilon(Q)A(u_{P(1)},\ldots,u_{P(m_1)})\,B(v_{Q(1)},\ldots,v_{Q(m_2)})$$

$$\times\prod_{j=1}^{m_1}\left(\frac{u_{P(j)}+\frac{i}{2}}{u_{P(j)}-\frac{i}{2}}\right)^{x_j}\prod_{l=1}^{m_2}\Phi(v_{Q(l)};u_{P(1)},\ldots,u_{P(y_l)})\Big\}$$

$$A(u_1,\ldots,u_{m_1}) = \prod_{1\le a<b\le m_1}(u_a-u_b+i)\,,$$

$$B(v_1,\ldots,v_{m_2}) = \prod_{1\le a<b\le m_2}(v_a-v_b+i)\,,$$

$$\Phi(v;u_1,\ldots,u_y) = \frac{1}{v-u_y-\frac{i}{2}}\prod_{l=1}^{y-1}\frac{v-u_l+\frac{i}{2}}{v-u_l-\frac{i}{2}}$$

$$H\,|\{u\};\{v\}\rangle = E\,|\{u\};\{v\}\rangle \qquad E = -\sum_{j=1}^{m_1}\frac{1}{u_j^2+\frac{1}{4}}$$

SU(3) highest-weight states

Liashyk, Pakuliak, Ragoucy 2026

Assume Bethe roots $\{u\}, \{v\}$ are known.

How to prepare the corresponding Bethe state $|\{u\}, \{v\}\rangle$?

Zare Harofteh, N 2606.24659

0's $k_0 = n - m_1$

#1's $k_1 = m_1 - m_2$

2's $k_2 = m_2$

Gray code for multiset permutations

Ko & Ruskey 1992

Ex: $(k_0, k_1, k_2) = (2, 1, 1)$

list of possible ditstrings:

0012
0021
0120
0102
0201
0210
1200
1002
1020
2010
2001
2100

"Gray property": only *two* dits *swap* at each step

Assume Bethe roots $\{u\}, \{v\}$ are known.

How to prepare the corresponding Bethe state $|\{u\}, \{v\}\rangle$?

Zare Harofteh, N 2606.24659

\# 0's $\quad k_0 = n - m_1$

\#1's $\quad k_1 = m_1 - m_2$

\# 2's $\quad k_2 = m_2$

Gray code for multiset permutations

Ko & Ruskey 1992

Ex: $(k_0, k_1, k_2) = (2, 1, 1)$

list of possible ditstrings:

0012
0021
0120
0102
0201
0210
1200
1002
1020
2010
2001
2100

"Gray property": only *two* dits *swap* at each step

Assume Bethe roots $\{u\}, \{v\}$ are known.

How to prepare the corresponding Bethe state $|\{u\}, \{v\}\rangle$?

Zare Harofteh, N 2606.24659

0's $\quad k_0 = n - m_1$

#1's $\quad k_1 = m_1 - m_2$

2's $\quad k_2 = m_2$

Gray code for multiset permutations

Ko & Ruskey 1992

Ex: $(k_0, k_1, k_2) = (2, 1, 1)$

list of possible ditstrings:

0012
0021
0120
0102
0201
0210
1200
1002
1020
2010
2001
2100

"Gray property": only *two* dits *swap* at each step

Assume Bethe roots $\{u\}, \{v\}$ are known.

How to prepare the corresponding Bethe state $|\{u\}, \{v\}\rangle$?

Zare Harofteh, N 2606.24659

0's $k_0 = n - m_1$

#1's $k_1 = m_1 - m_2$

2's $k_2 = m_2$

Gray code for multiset permutations

Ko & Ruskey 1992

Ex: $(k_0, k_1, k_2) = (2, 1, 1)$

list of possible ditstrings:

0012
0021
0120
0102
0201
0210
1200
1002
1020
2010
2001
2100

"Gray property": only *two* dits *swap* at each step

Assume Bethe roots $\{u\}, \{v\}$ are known.

How to prepare the corresponding Bethe state $|\{u\}, \{v\}\rangle$?

Zare Harofteh, N 2606.24659

0's $k_0 = n - m_1$

#1's $k_1 = m_1 - m_2$

2's $k_2 = m_2$

Gray code for multiset permutations

Ko & Ruskey 1992

Ex: $(k_0, k_1, k_2) = (2, 1, 1)$

list of possible ditstrings:

0012
0021
0120
0102
0201
0210
1200
1002
1020
2010
2001
2100

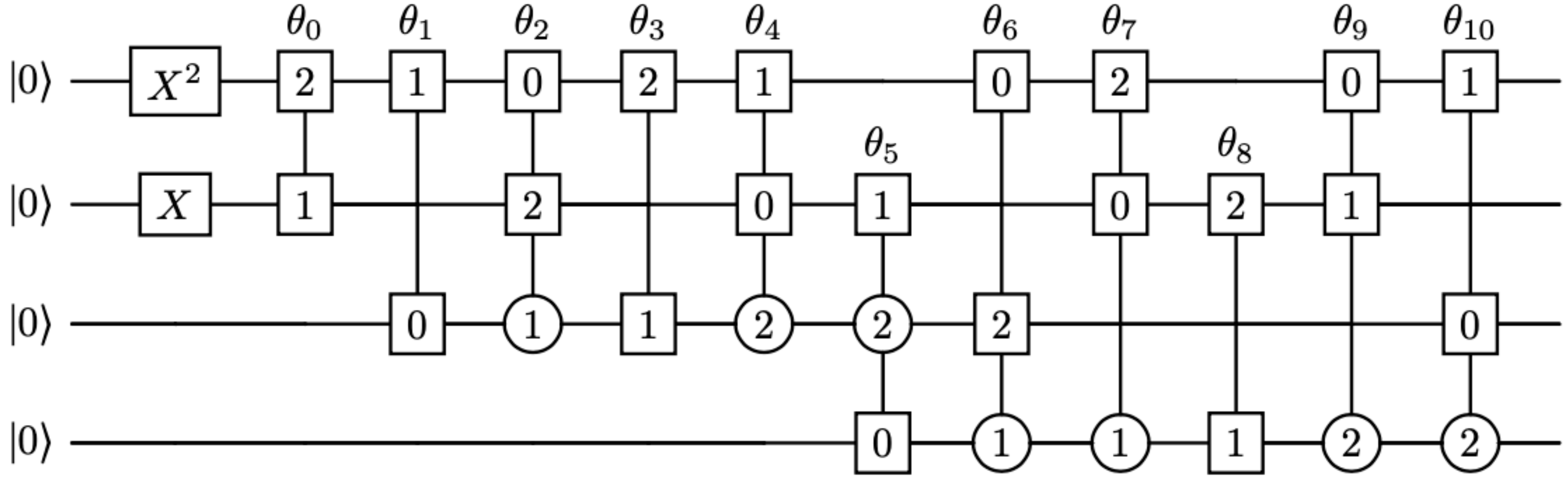
etc.

"Gray property": only *two* dits *swap* at each step

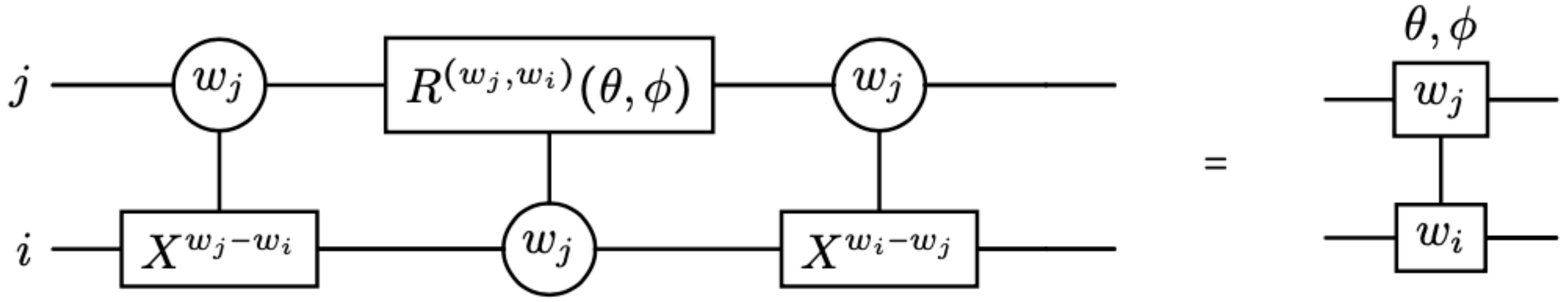
Build quantum state by corresponding sequence of controlled 2-qudit rotations

Ex: $(k_0, k_1, k_2) = (2, 1, 1)$

$$|\psi(2,1,1)\rangle = a_{0012}|0012\rangle + a_{0021}|0021\rangle + a_{0120}|0120\rangle + a_{0102}|0102\rangle + a_{0201}|0201\rangle + a_{0210}|0210\rangle$$
$$+ a_{1200}|1200\rangle + a_{1002}|1002\rangle + a_{1020}|1020\rangle + a_{2010}|2010\rangle + a_{2001}|2001\rangle + a_{2100}|2100\rangle$$



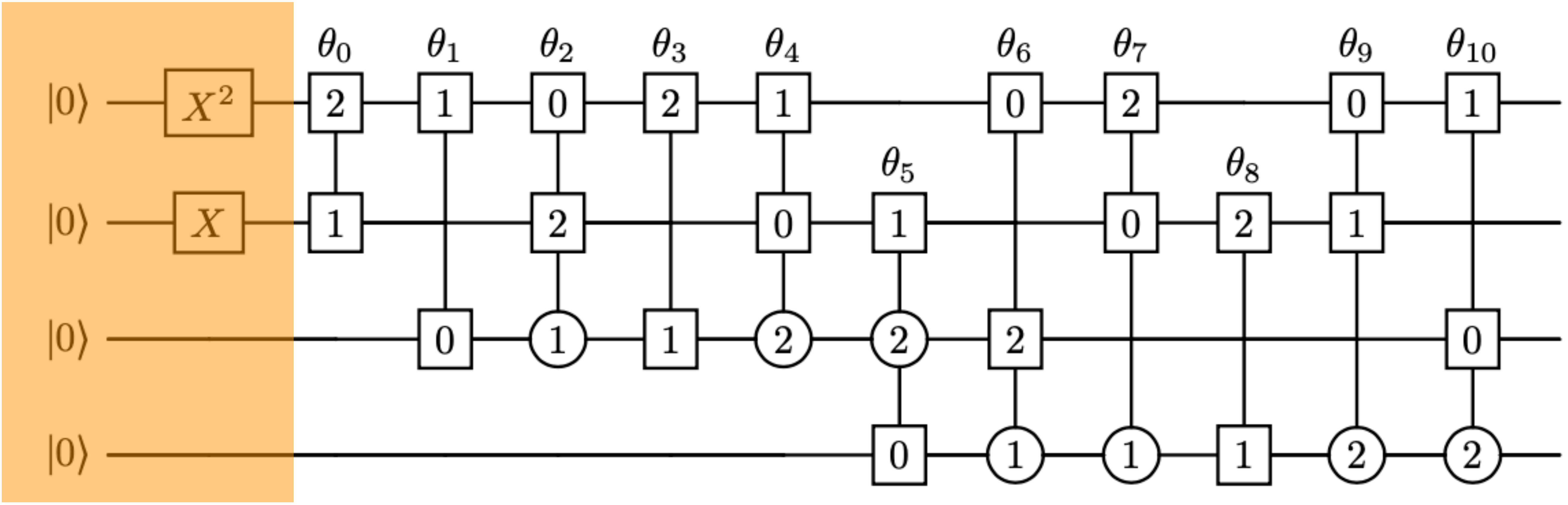
Gray gate



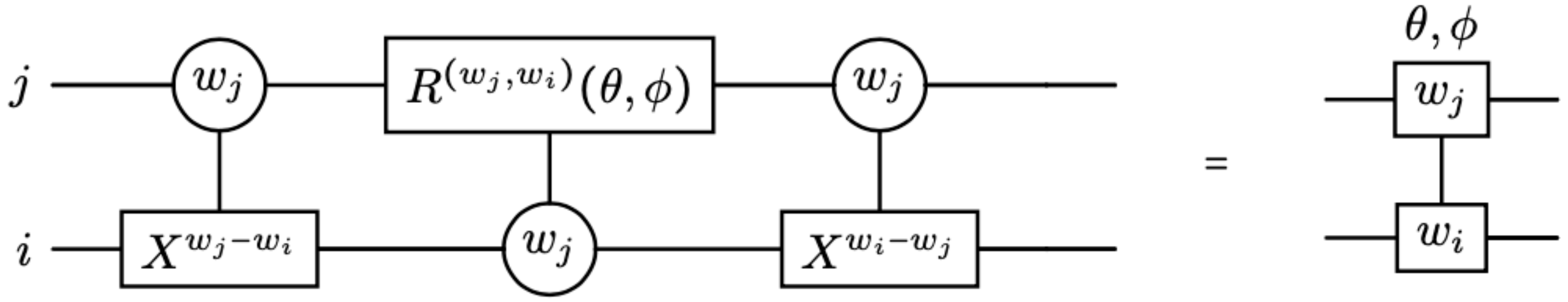
Build quantum state by corresponding sequence of controlled 2-qudit rotations

Ex: $(k_0, k_1, k_2) = (2, 1, 1)$

$$|\psi(2,1,1)\rangle = a_{0012}|0012\rangle + a_{0021}|0021\rangle + a_{0120}|0120\rangle + a_{0102}|0102\rangle + a_{0201}|0201\rangle + a_{0210}|0210\rangle$$
$$+ a_{1200}|1200\rangle + a_{1002}|1002\rangle + a_{1020}|1020\rangle + a_{2010}|2010\rangle + a_{2001}|2001\rangle + a_{2100}|2100\rangle$$



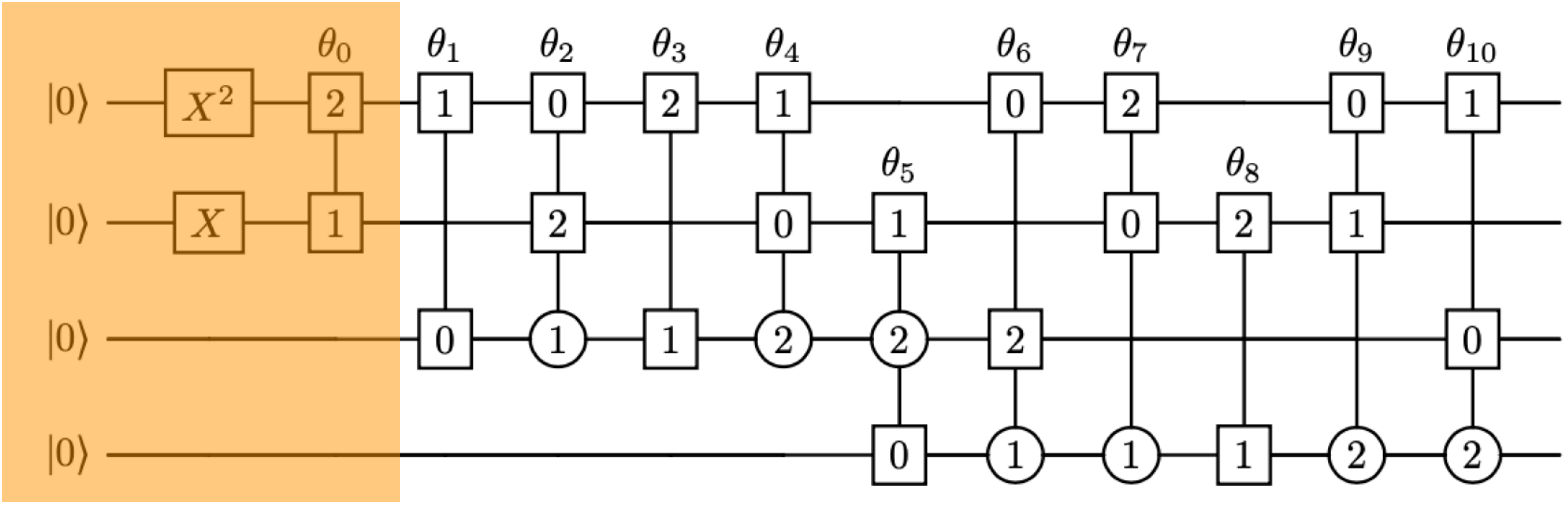
Gray gate



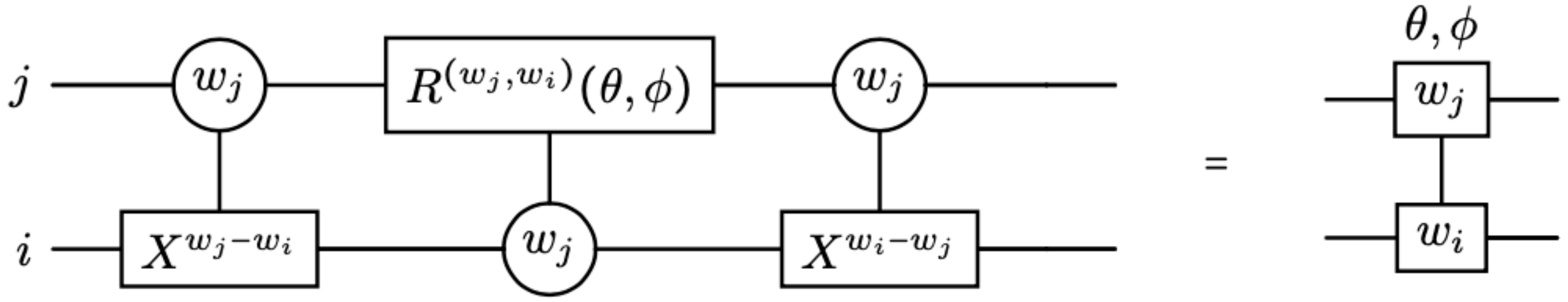
Build quantum state by corresponding sequence of controlled 2-qudit rotations

Ex: $(k_0, k_1, k_2) = (2, 1, 1)$

$$|\psi(2,1,1)\rangle = a_{0012}|0012\rangle + a_{0021}|0021\rangle + a_{0120}|0120\rangle + a_{0102}|0102\rangle + a_{0201}|0201\rangle + a_{0210}|0210\rangle$$
$$+ a_{1200}|1200\rangle + a_{1002}|1002\rangle + a_{1020}|1020\rangle + a_{2010}|2010\rangle + a_{2001}|2001\rangle + a_{2100}|2100\rangle$$



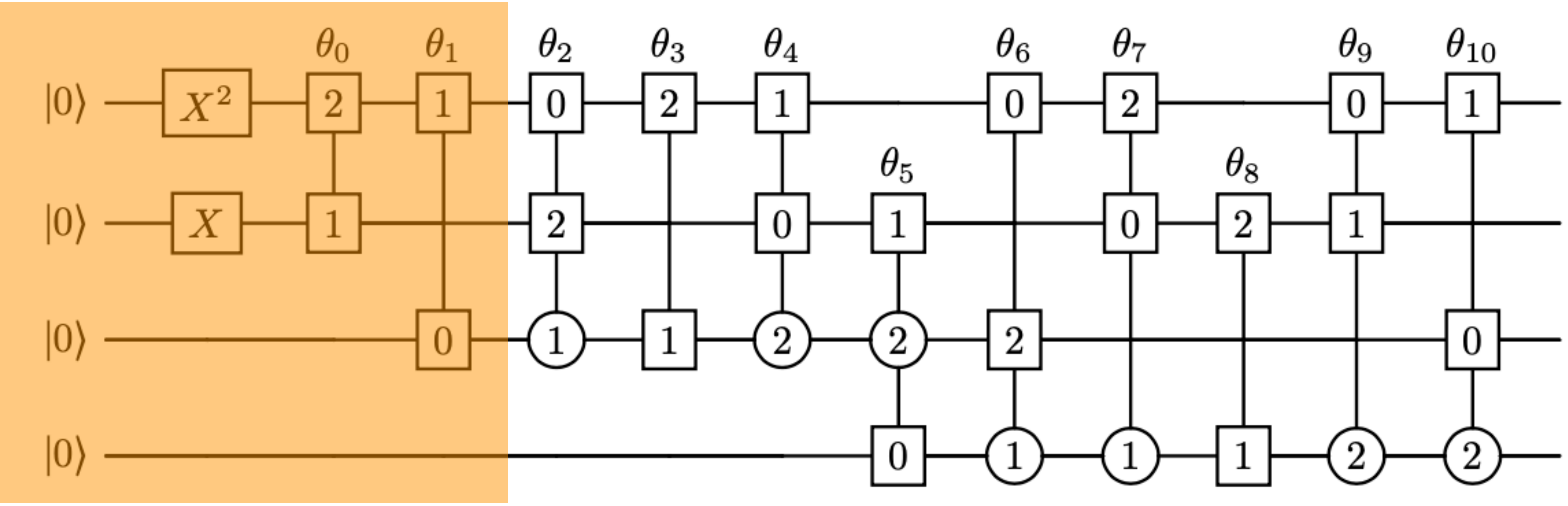
Gray gate



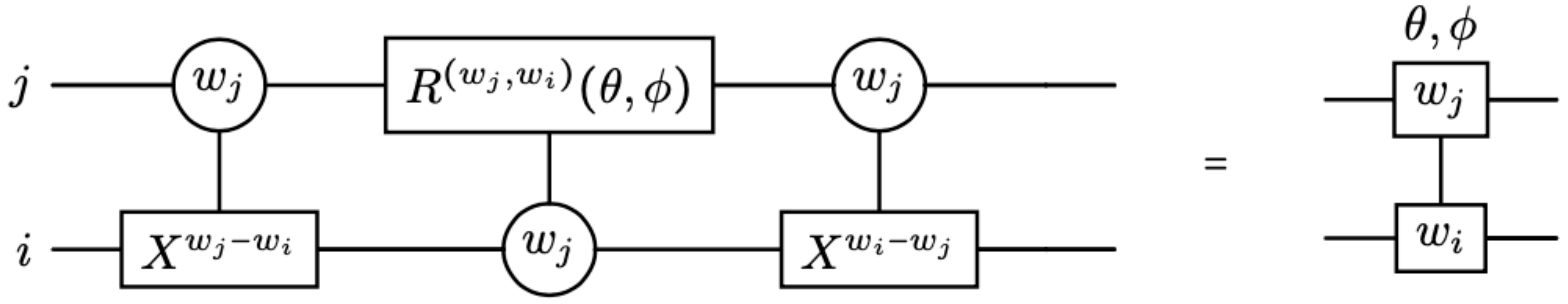
Build quantum state by corresponding sequence of controlled 2-qudit rotations

Ex: $(k_0, k_1, k_2) = (2, 1, 1)$

$$|\psi(2,1,1)\rangle = a_{0012}|0012\rangle + a_{0021}|0021\rangle + a_{0120}|0120\rangle + a_{0102}|0102\rangle + a_{0201}|0201\rangle + a_{0210}|0210\rangle + a_{1200}|1200\rangle + a_{1002}|1002\rangle + a_{1020}|1020\rangle + a_{2010}|2010\rangle + a_{2001}|2001\rangle + a_{2100}|2100\rangle$$



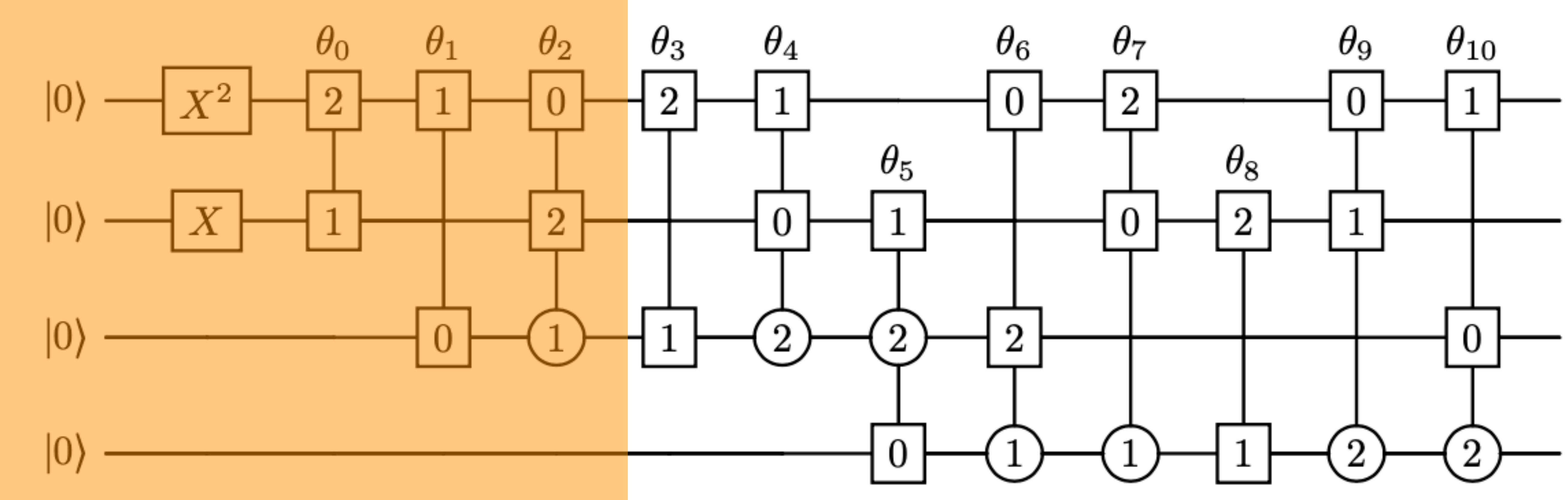
Gray gate



Build quantum state by corresponding sequence of controlled 2-qudit rotations

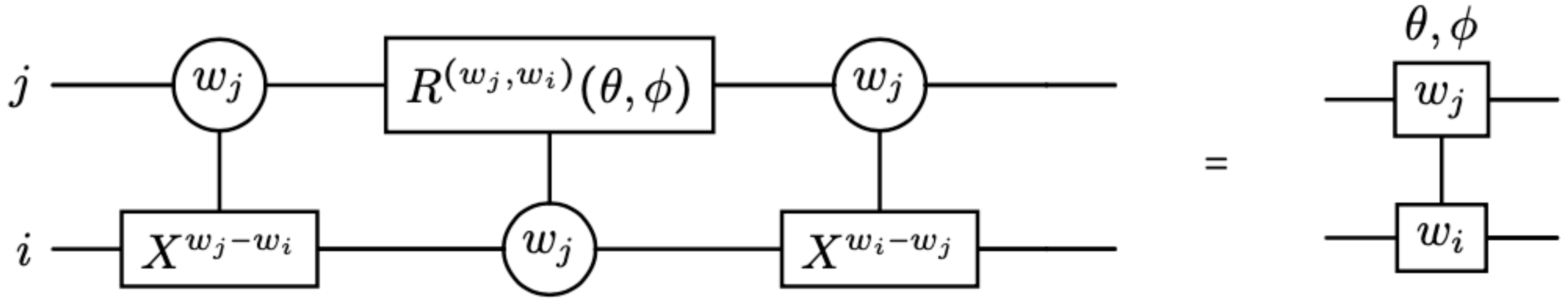
Ex: $(k_0, k_1, k_2) = (2, 1, 1)$

$$|\psi(2,1,1)\rangle = a_{0012}|0012\rangle + a_{0021}|0021\rangle + a_{0120}|0120\rangle + a_{0102}|0102\rangle + a_{0201}|0201\rangle + a_{0210}|0210\rangle$$
$$+ a_{1200}|1200\rangle + a_{1002}|1002\rangle + a_{1020}|1020\rangle + a_{2010}|2010\rangle + a_{2001}|2001\rangle + a_{2100}|2100\rangle$$



etc.

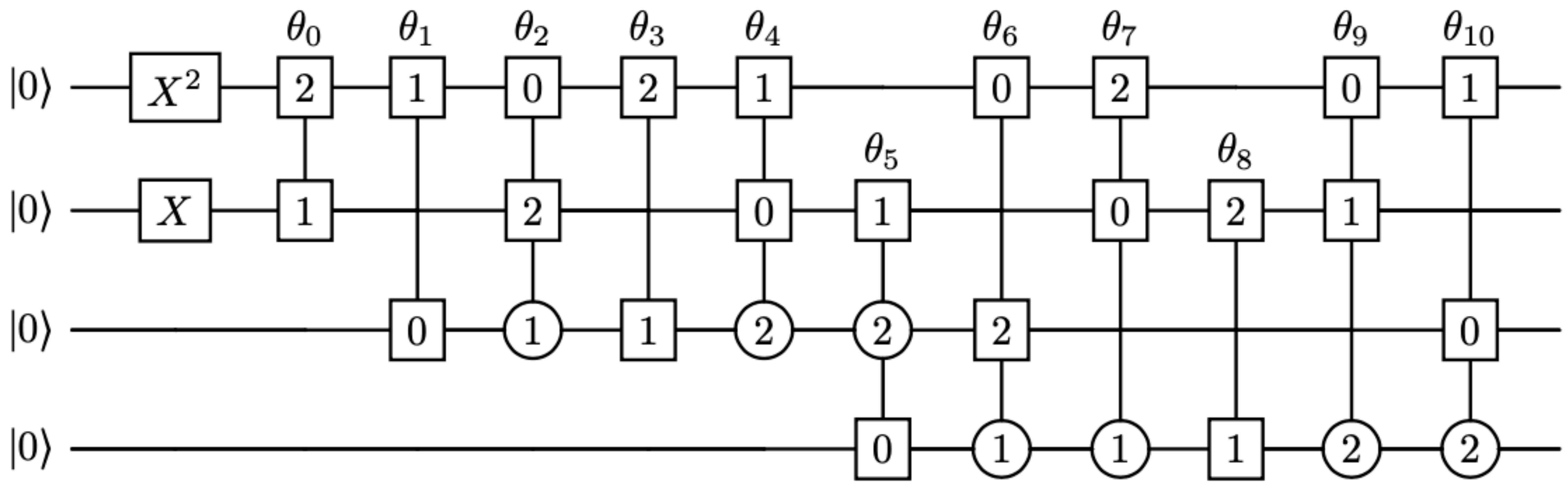
Gray gate



Build quantum state by corresponding sequence of controlled 2-qudit rotations

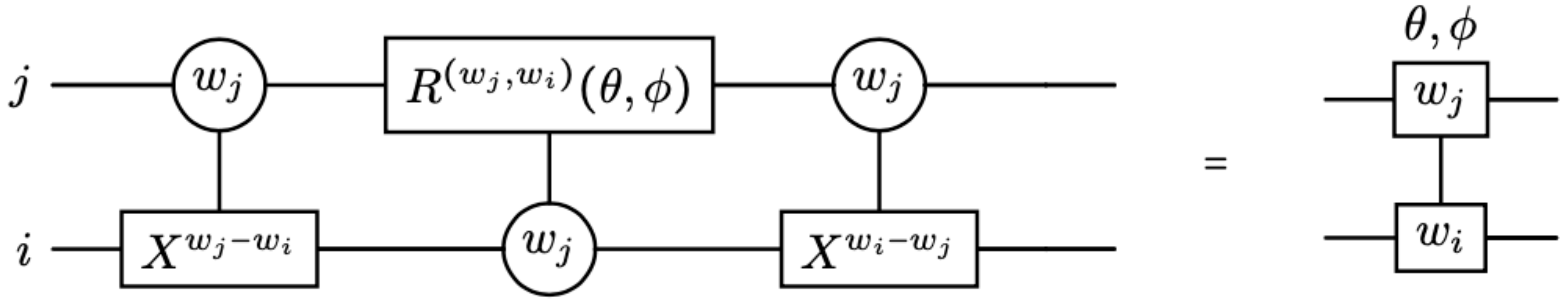
Ex: $(k_0, k_1, k_2) = (2, 1, 1)$

$$|\psi(2,1,1)\rangle = a_{0012}|0012\rangle + a_{0021}|0021\rangle + a_{0120}|0120\rangle + a_{0102}|0102\rangle + a_{0201}|0201\rangle + a_{0210}|0210\rangle$$
$$+ a_{1200}|1200\rangle + a_{1002}|1002\rangle + a_{1020}|1020\rangle + a_{2010}|2010\rangle + a_{2001}|2001\rangle + a_{2100}|2100\rangle$$



etc.

Gray gate



circuit size & depth: $\mathcal{O}(n\mathcal{D})$

$$\mathcal{D} = \binom{n}{\vec{k}} = \frac{n!}{k_0!\, k_1!\, k_2!} = \mathcal{O}(3^n) \quad \text{for worst case} \quad \vec{k} = (\tfrac{n}{3}, \tfrac{n}{3}, \tfrac{n}{3})$$

Outline

I. Gray codes

II. Quantum computers & quantum state preparation

III. Bethe state preparation

- spin 1/2
- spin s
- nesting

new

IV. Conclusions

Conclusions

- Quantum state preparation is a fundamental but challenging task in quantum computing
- Certain entangled multi-qubit (and multi-qudit) states *can* be prepared efficiently
- Gray codes provide a simple — but costly — option for preparing general Bethe states

"Take-home message"

Homework:

Can integrability be exploited to prepare Bethe states more efficiently?

Thank you for your attention!