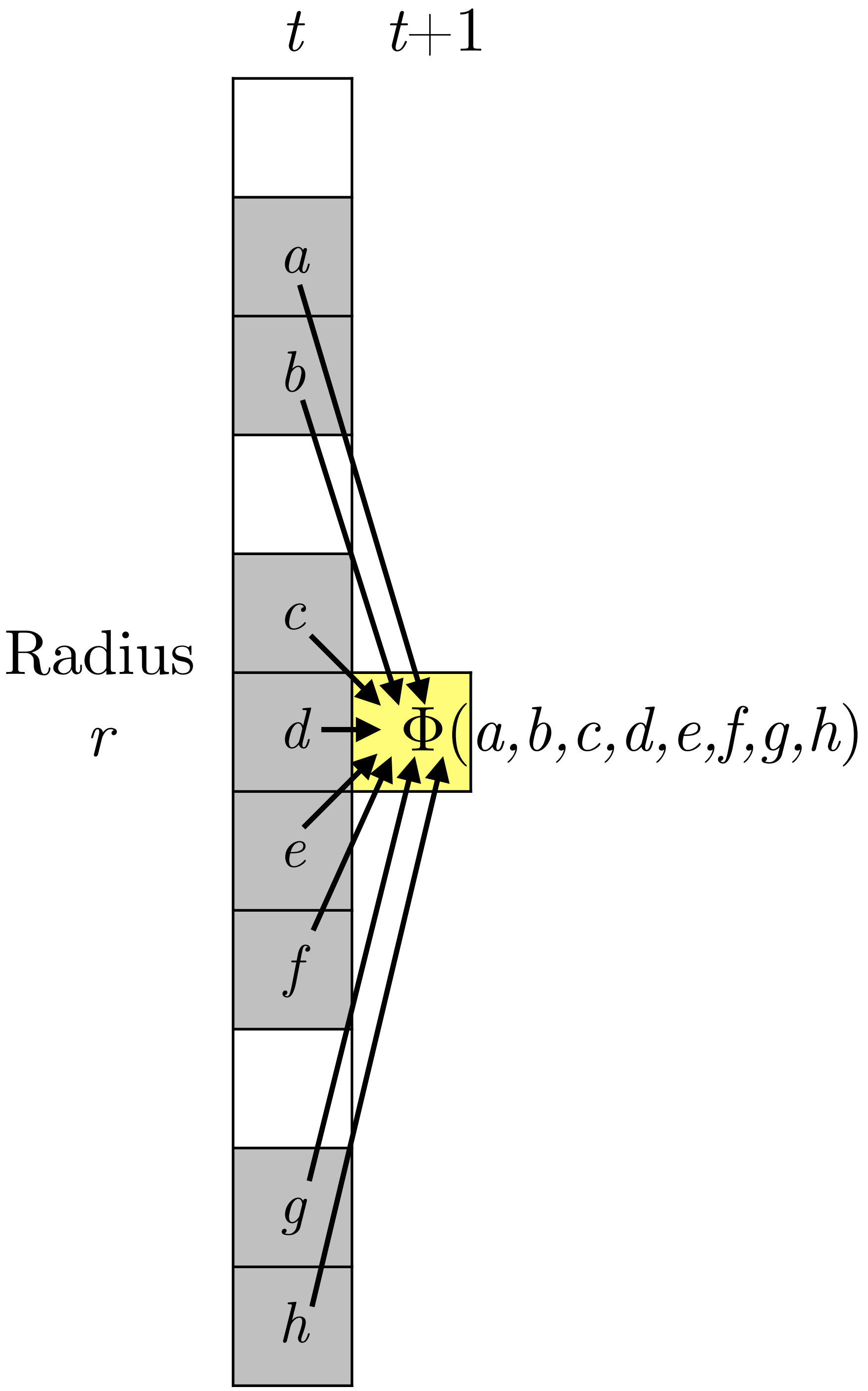
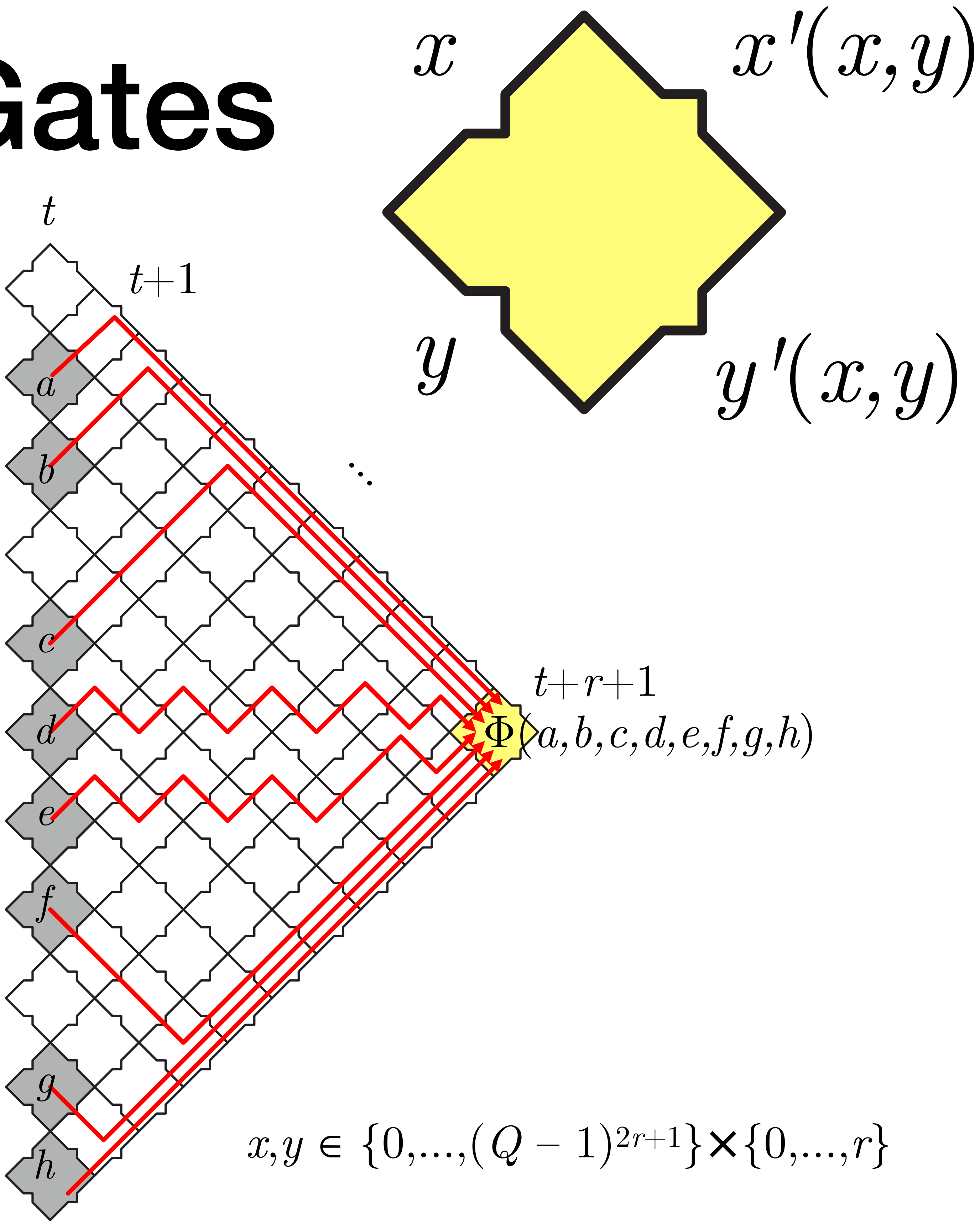
Our new result:
Intrinsic Simulation of 1D Cellular Automata

from 1D CA to 2-in 2-out Gates

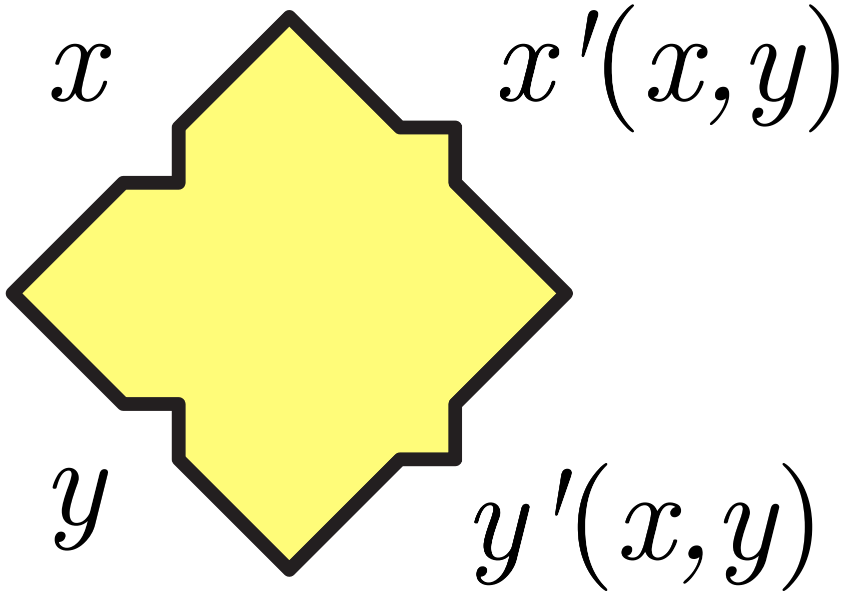


$a,b,c,d,e,f,g,h \in \{0,\ldots,Q{-}1\}$



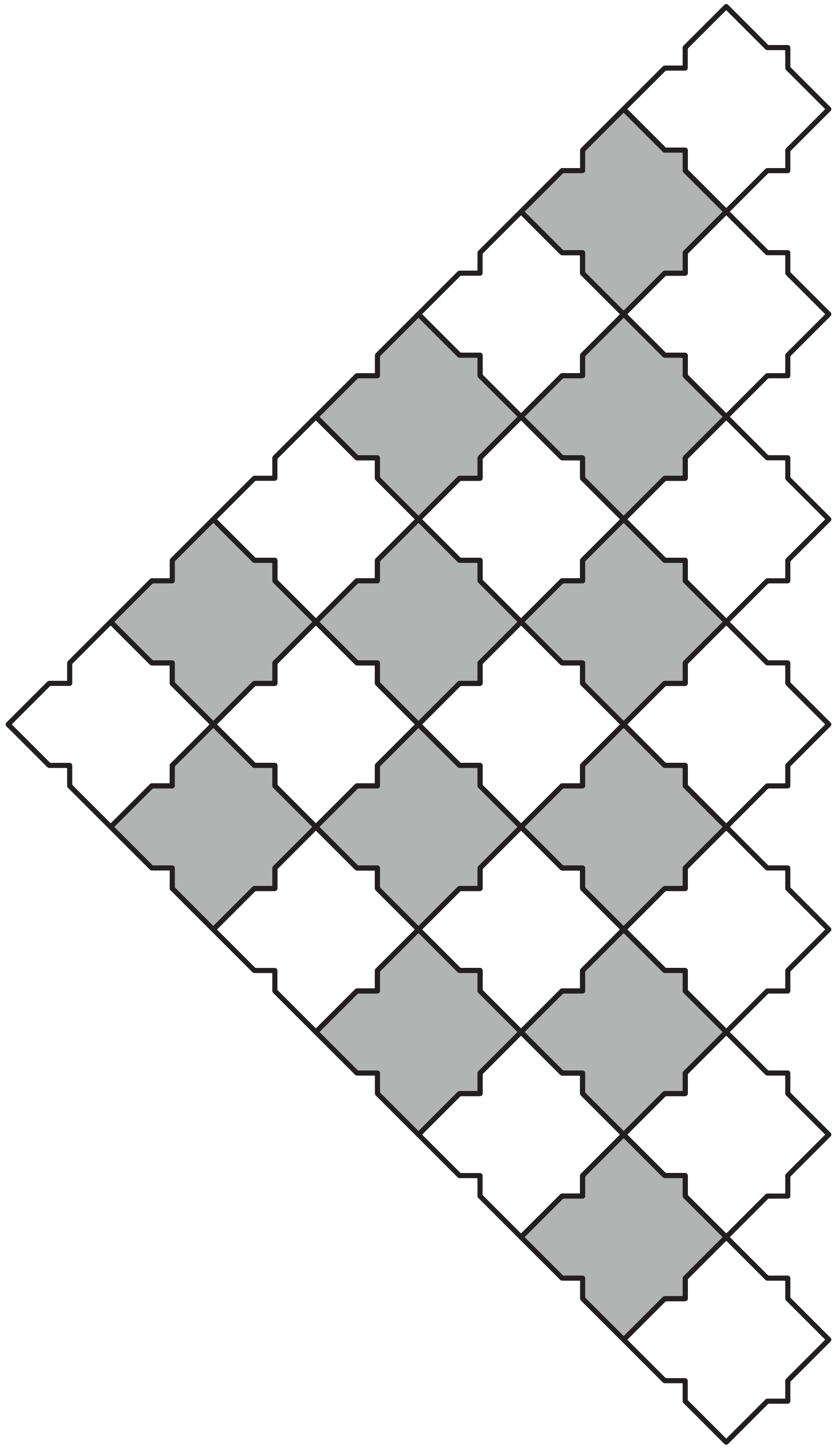
$x,y \in \{0,\ldots,(Q-1)^{2r+1}\}\times\{0,\ldots,r\}$

Oritatami system simulating 2-in 2-out gates

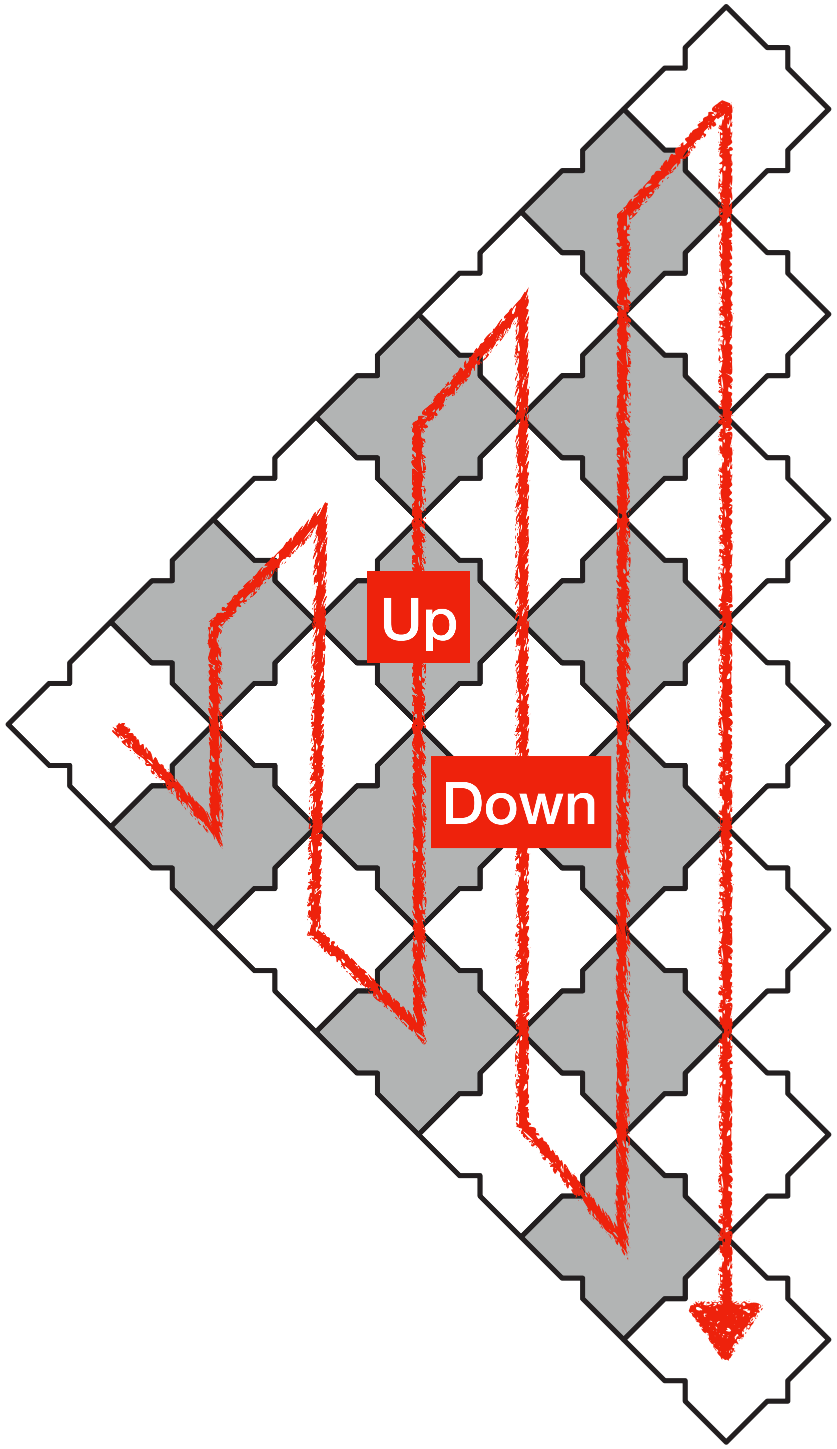


where $x, y \in \{0, \ldots, Q-1\}$

Oritatami system simulating 2-in 2-out gates



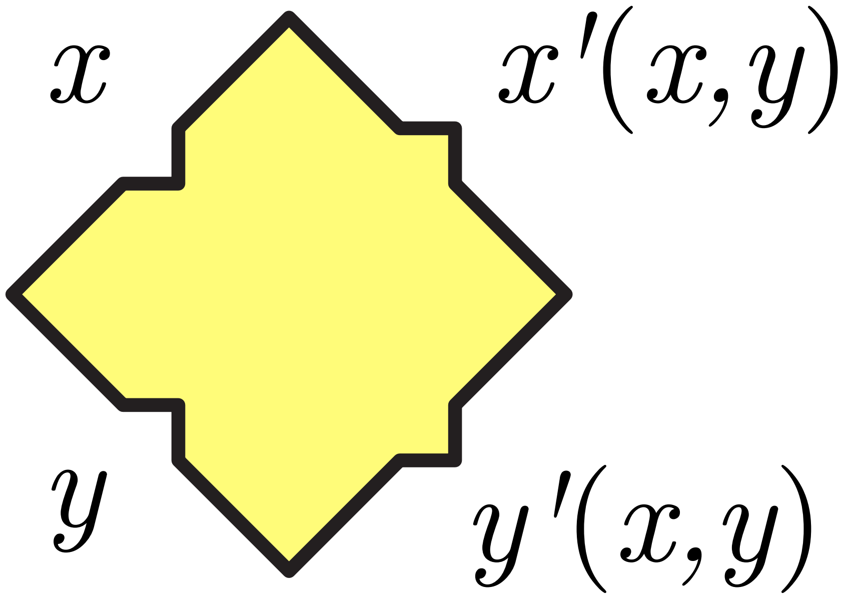
Oritatami system simulating 2-in 2-out gates



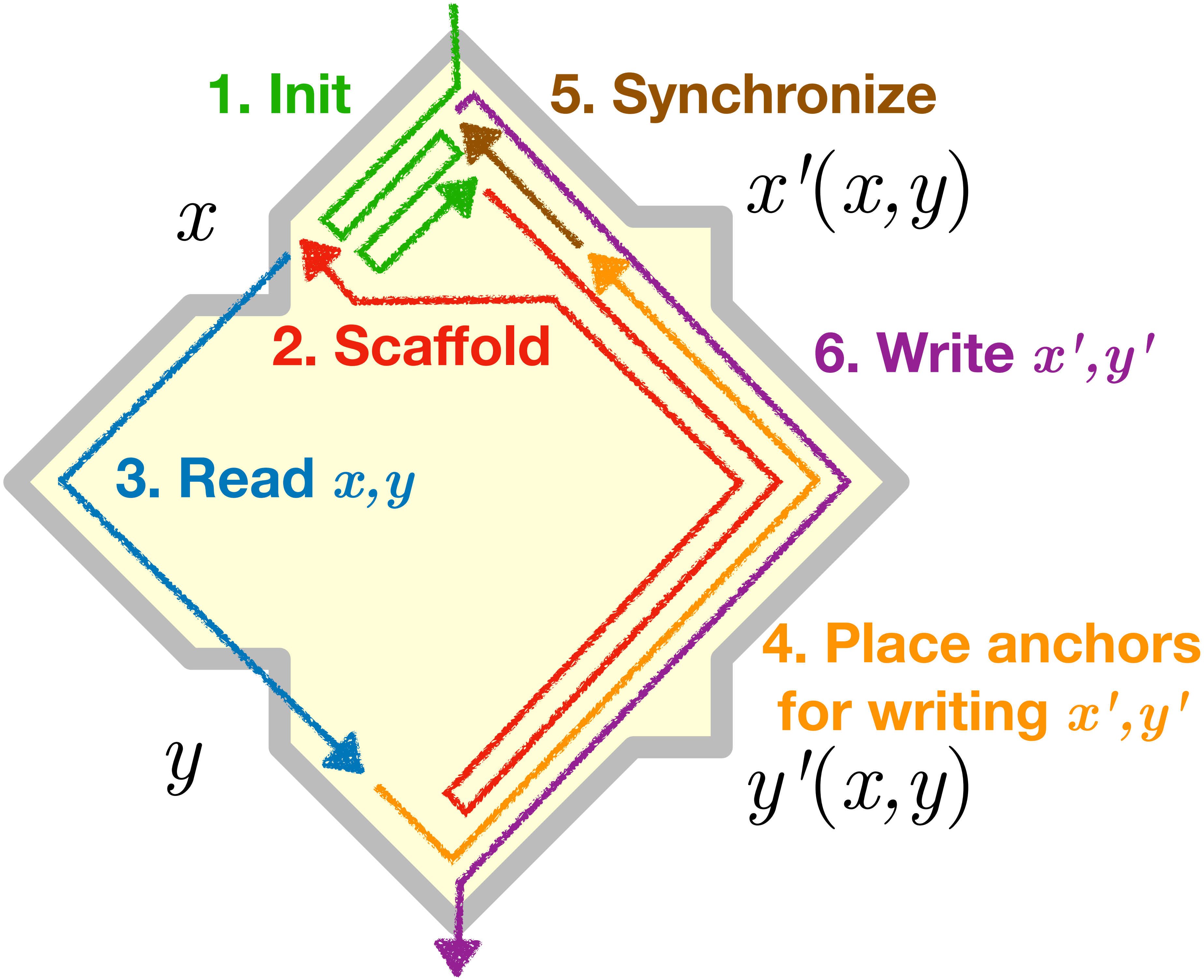
Up & Down paths are just mirrored of each other

→ Just need to add an extra Up/Down-state and to mirror the transition function:

$$X'(x,y,\downarrow) := (x'(x,y),\uparrow)$$
$$X'(x,y,\uparrow) := (y'(y,x),\downarrow)$$



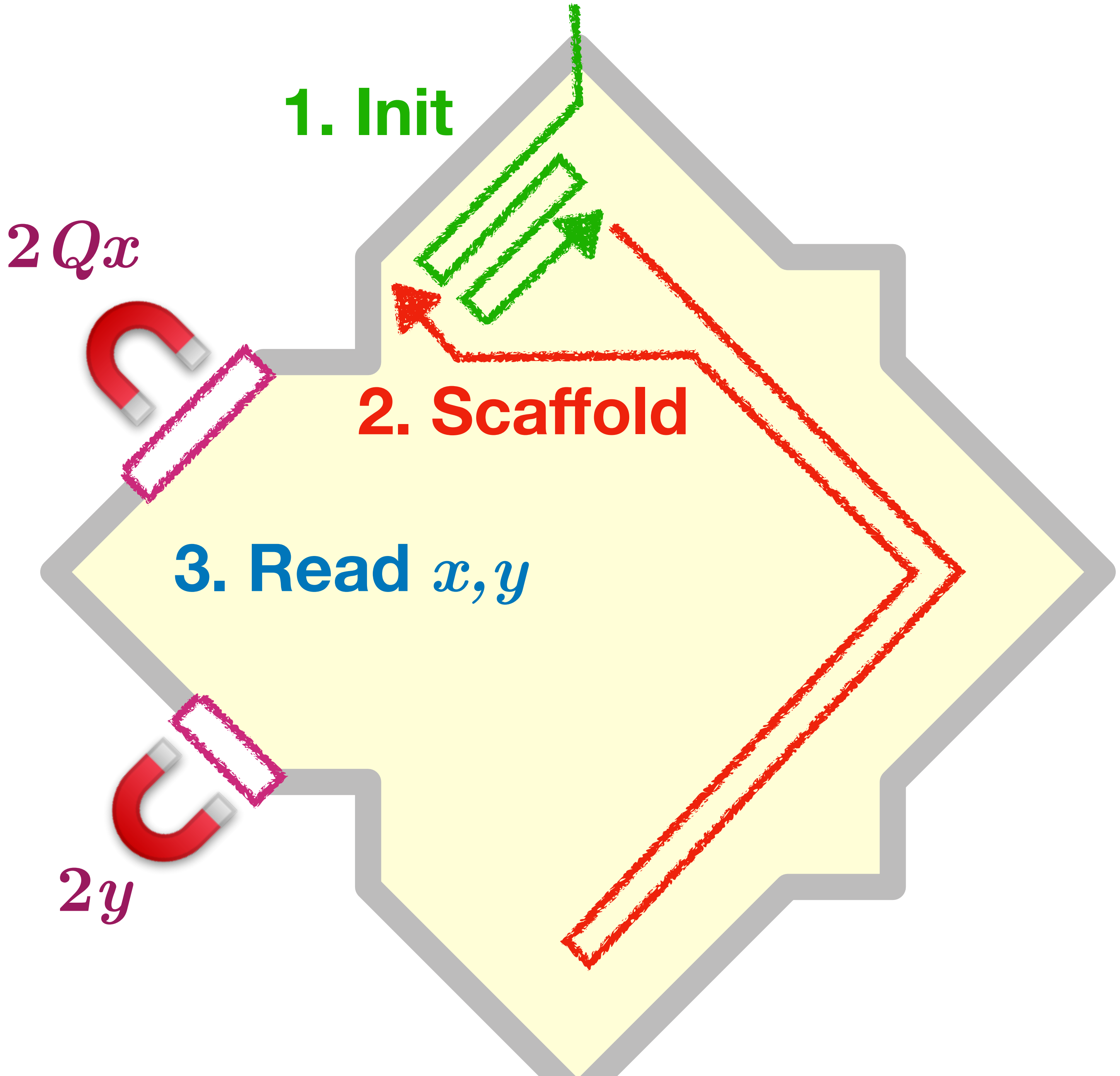
Let's simulate one 2-in 2-out gate



Read-Write mechanism

Writing x, y = Placing magnets of length $2Qx$ and $2y$

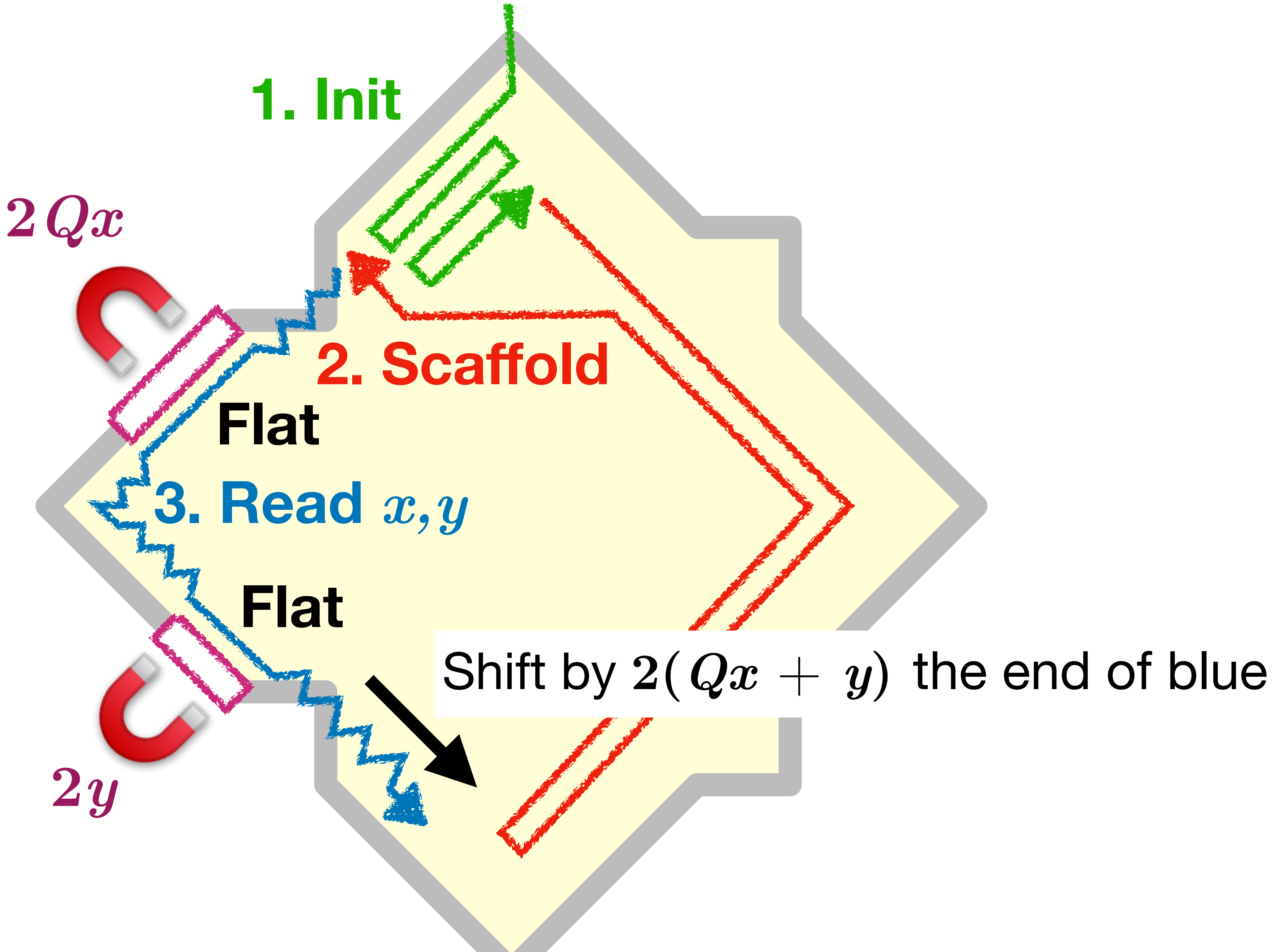
Reading x, y = Create an offset of $2(Qx + y)$



Read-Write mechanism

Writing x, y = Placing magnets of length $2Qx$ and $2y$

Reading x, y = Create an offset of $2(Qx + y)$

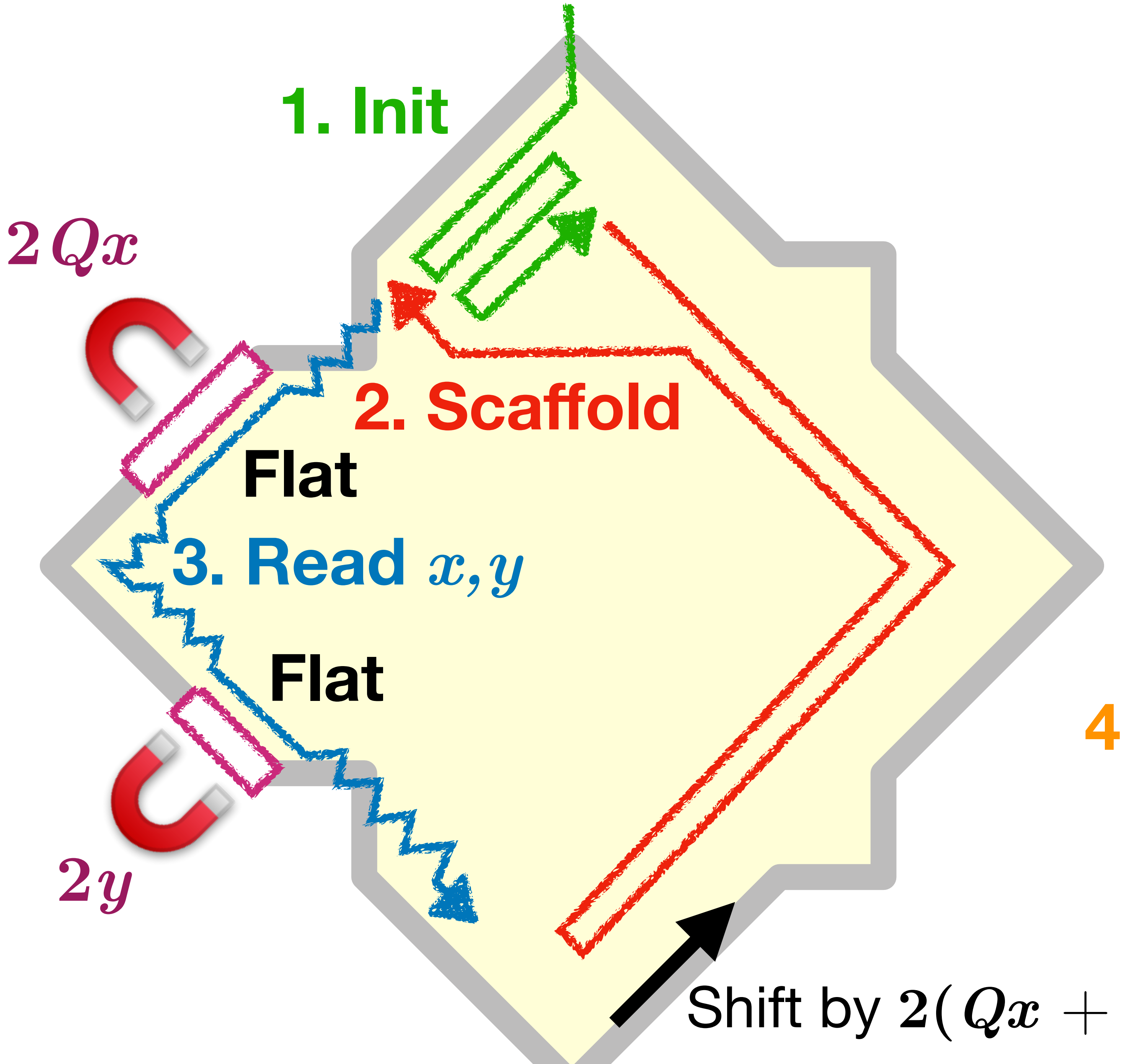


Shift by $2(Qx + y)$ the end of blue

Read-Write mechanism

Writing x, y = Placing magnets of length $2Qx$ and $2y$

Reading x, y = Create an offset of $2(Qx + y)$



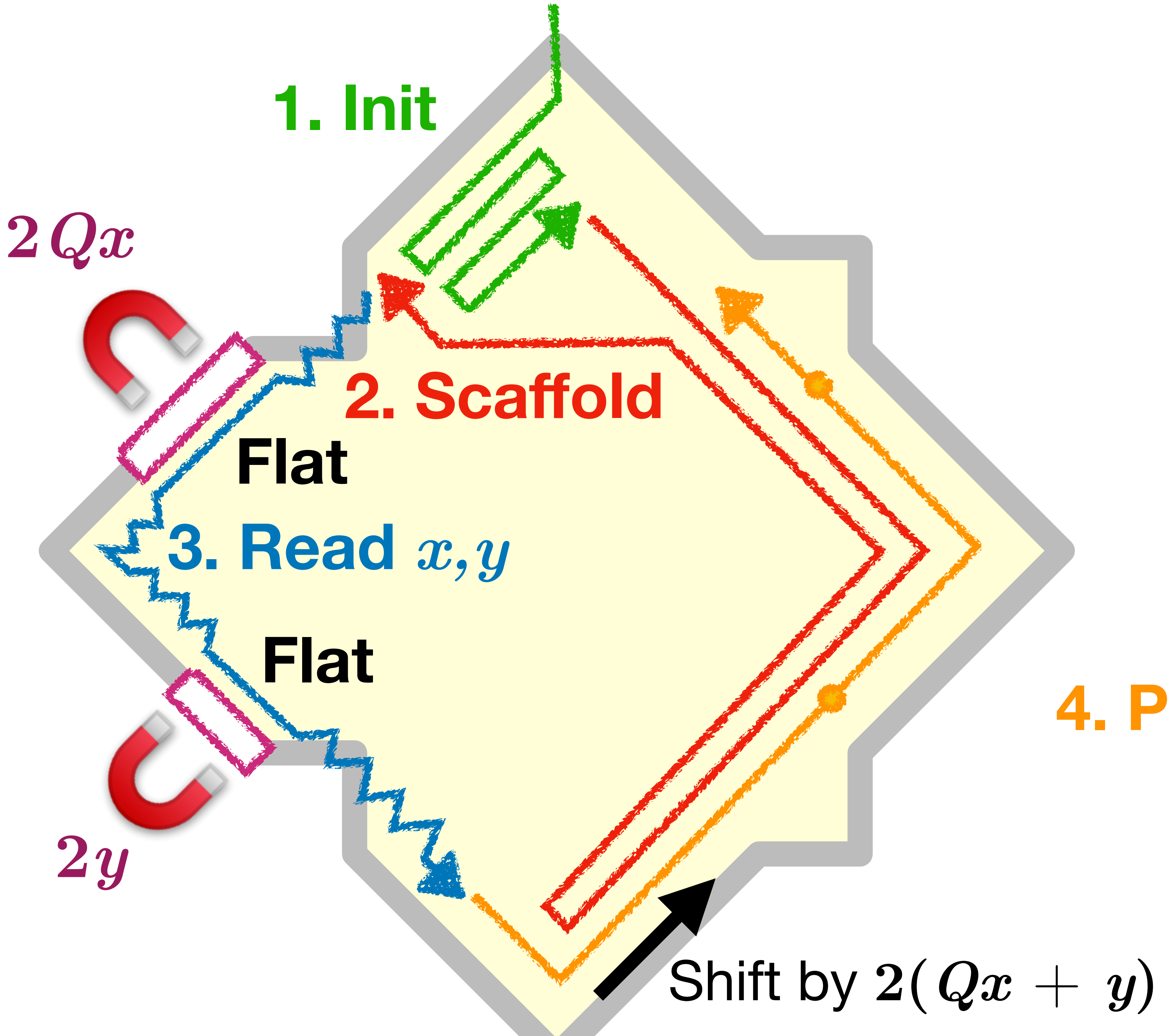
4. Place anchors shifted by $2(Qx + y)$

Shift by $2(Qx + y)$

Read-Write mechanism

Writing x, y = Placing magnets of length $2Qx$ and $2y$

Reading x, y = Create an offset of $2(Qx + y)$

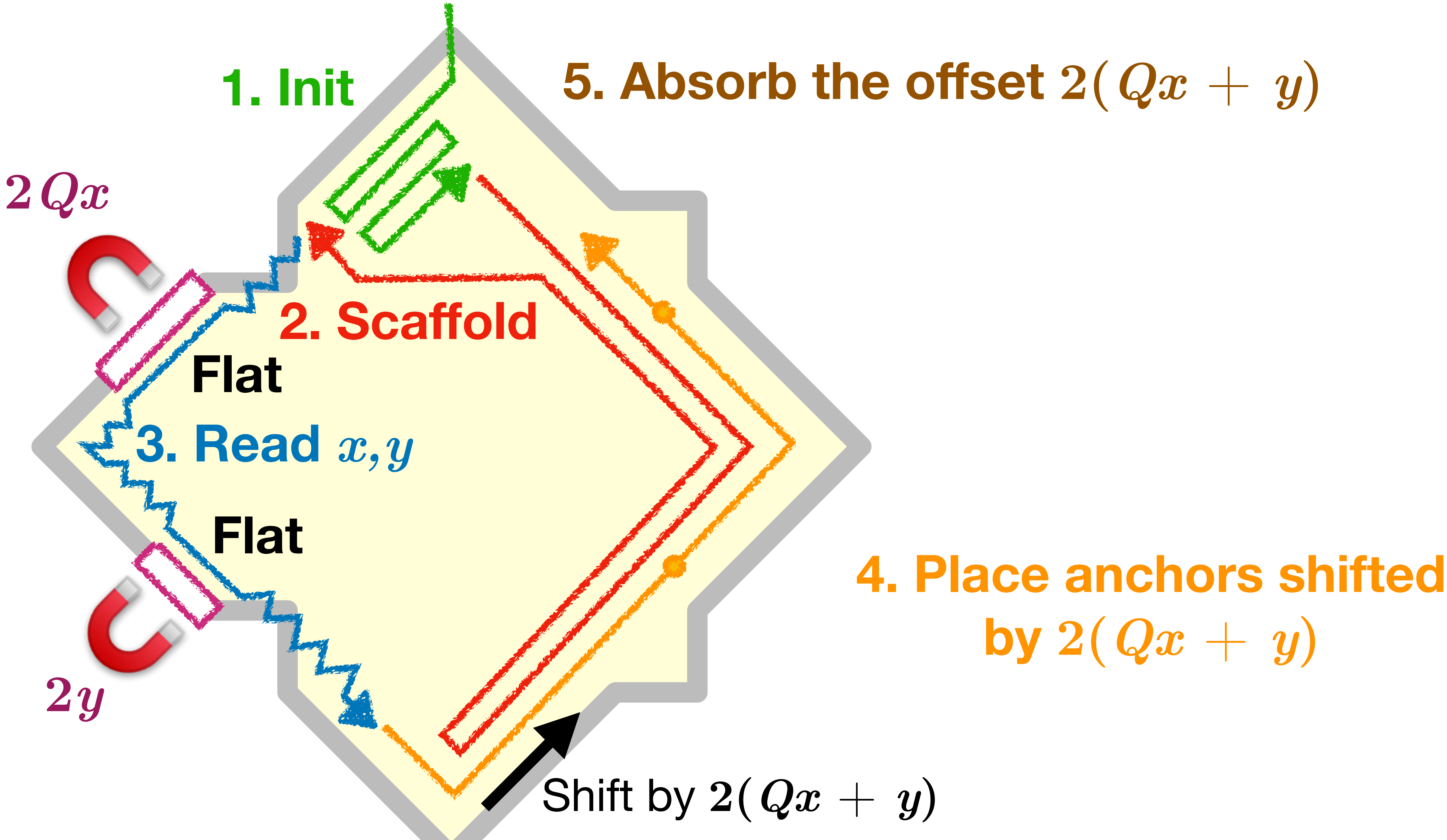


4. Place anchors shifted by $2(Qx + y)$

Read-Write mechanism

Writing x, y = Placing magnets of length $2Qx$ and $2y$

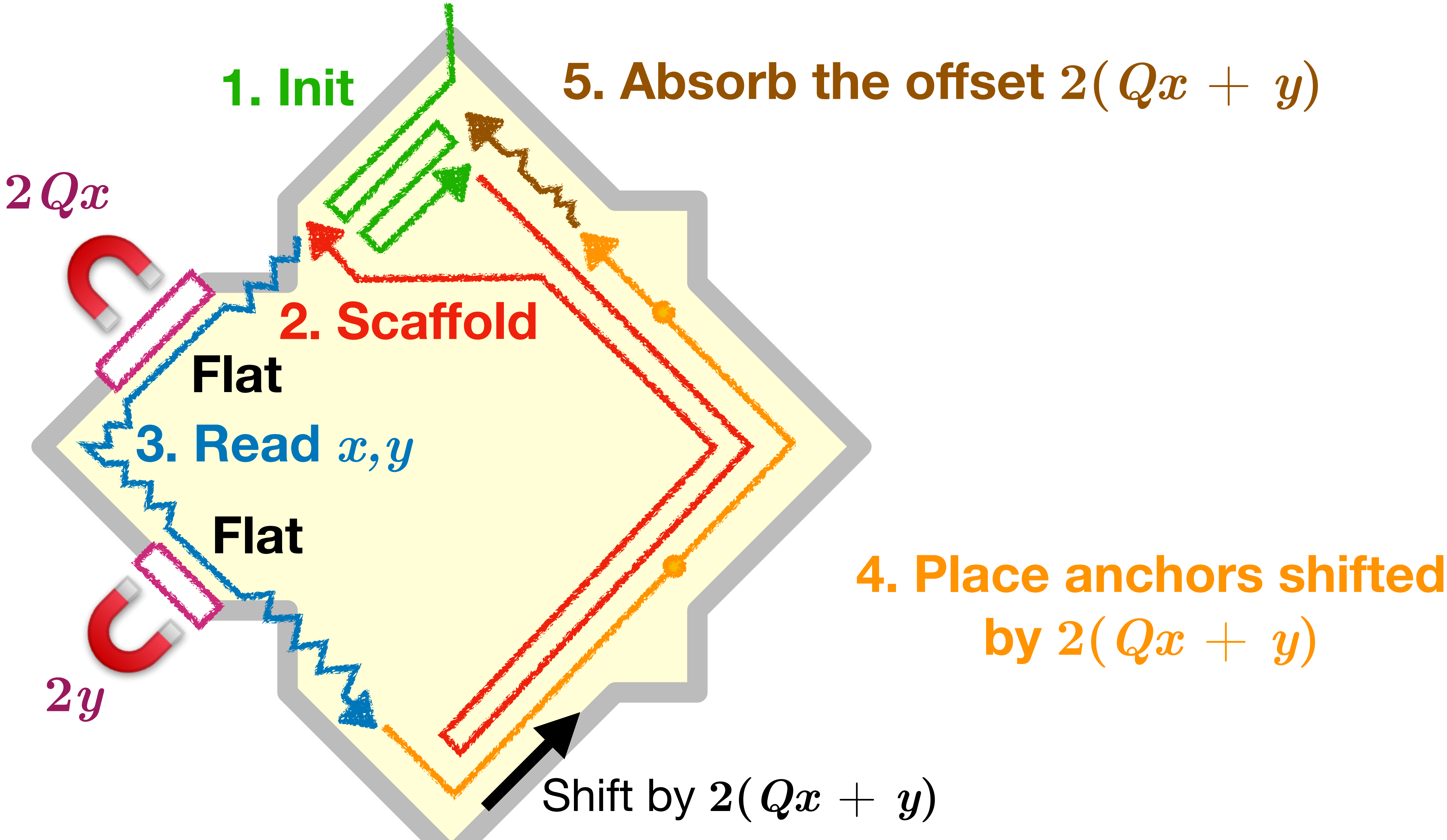
Reading x, y = Create an offset of $2(Qx + y)$


1. Init
5. Absorb the offset $2(Qx + y)$
$2Qx$
2. Scaffold
Flat
3. Read x, y
Flat
4. Place anchors shifted by $2(Qx + y)$
$2y$
Shift by $2(Qx + y)$

Read-Write mechanism

Writing x, y = Placing magnets of length $2Qx$ and $2y$

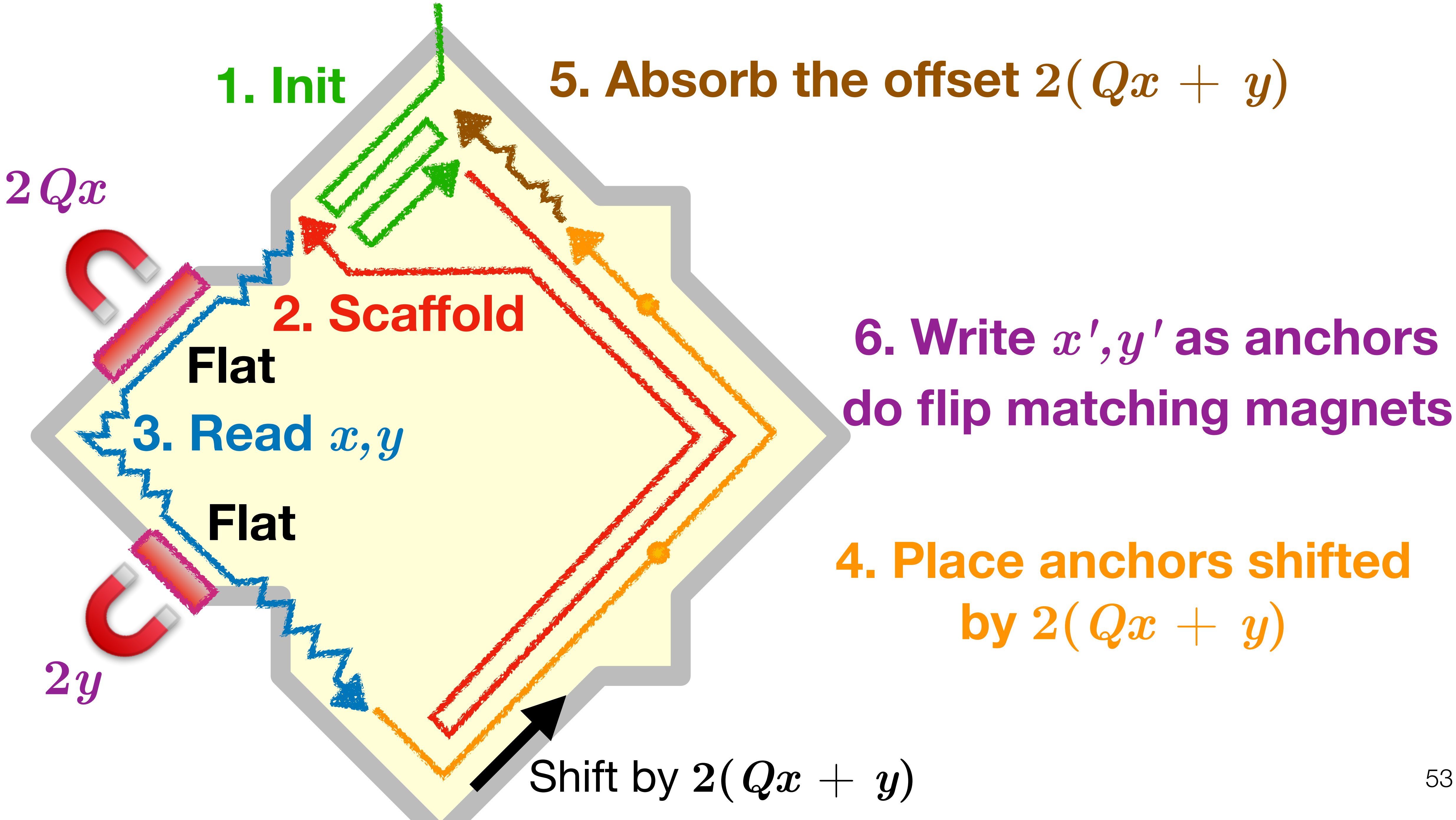
Reading x, y = Create an offset of $2(Qx + y)$


1. Init
5. Absorb the offset $2(Qx + y)$
$2Qx$
2. Scaffold
Flat
3. Read x, y
Flat
$2y$
4. Place anchors shifted by $2(Qx + y)$
Shift by $2(Qx + y)$

Read-Write mechanism

Writing x, y = Placing magnets of length $2Qx$ and $2y$

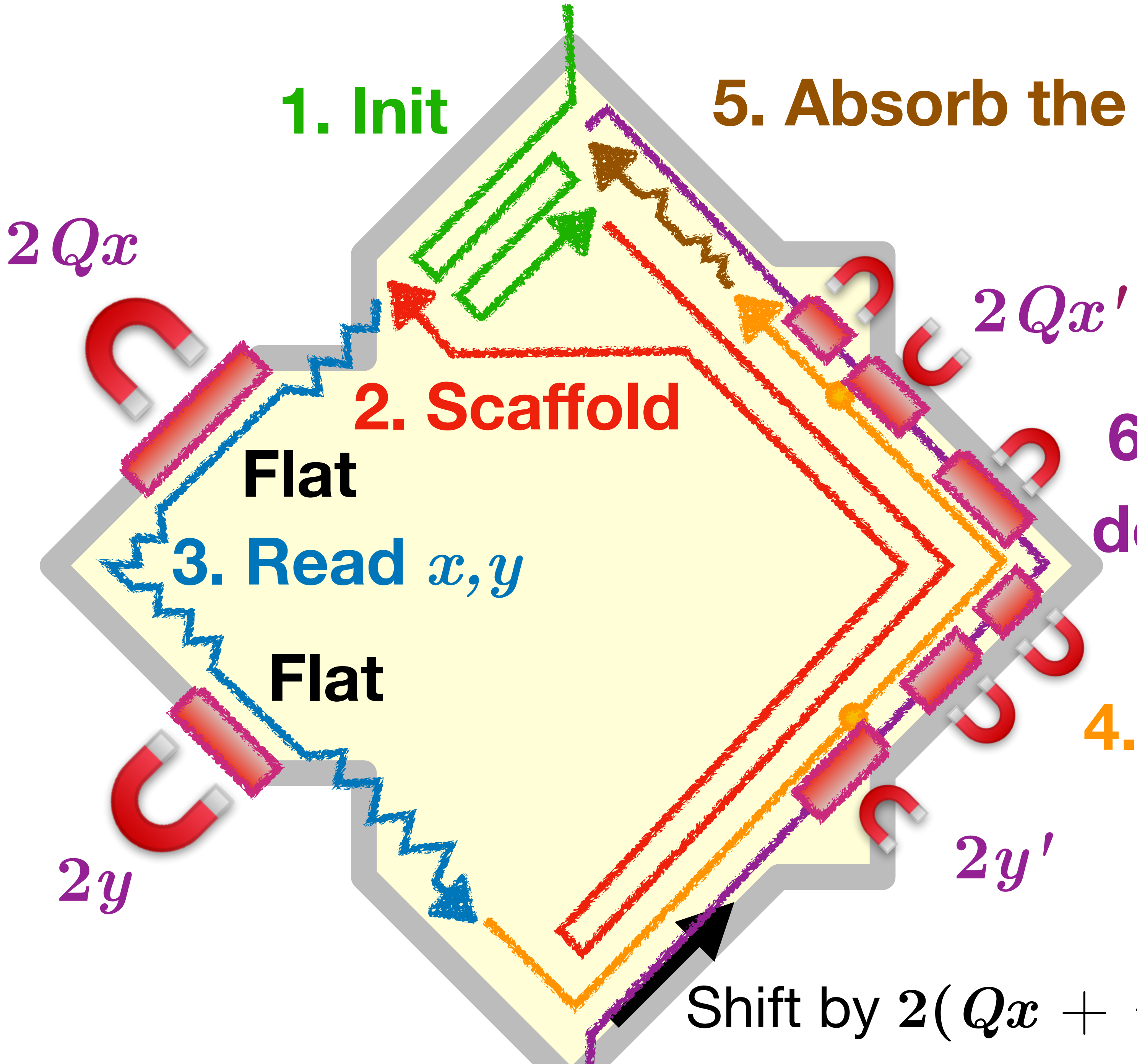
Reading x, y = Create an offset of $2(Qx + y)$



Read-Write mechanism

Writing x, y = Placing magnets of length $2Qx$ and $2y$

Reading x, y = Create an offset of $2(Qx + y)$



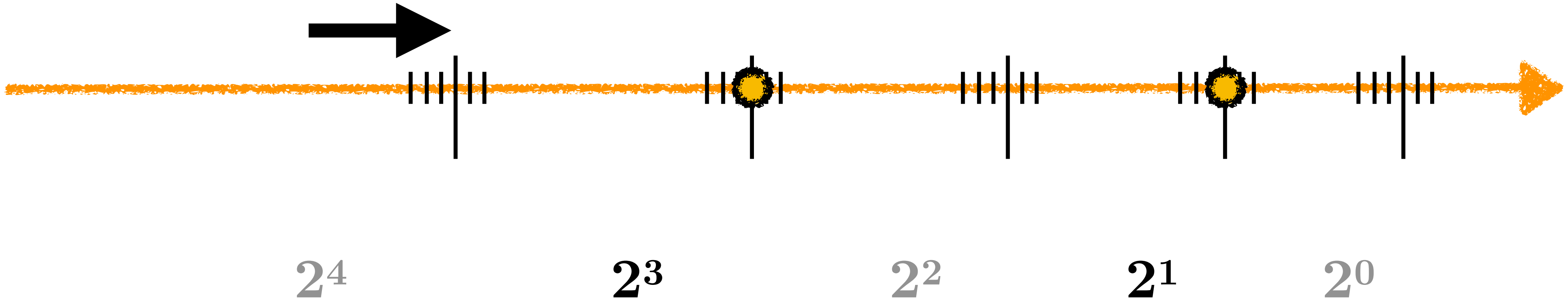
1. Init
5. Absorb the offset $2(Qx + y)$
$2Qx$
$2Qx'$
2. Scaffold
Flat
6. Write x', y' as anchors do flip matching magnets
3. Read x, y
Flat
4. Place anchors shifted by $2(Qx + y)$
$2y$
$2y'$
Shift by $2(Qx + y)$

Read-Write mechanism

Place the anchors as follows:

Consider $x'(x, y) = 10 = 2^3 + 2^1$

Shift by $2(Qx + y)$



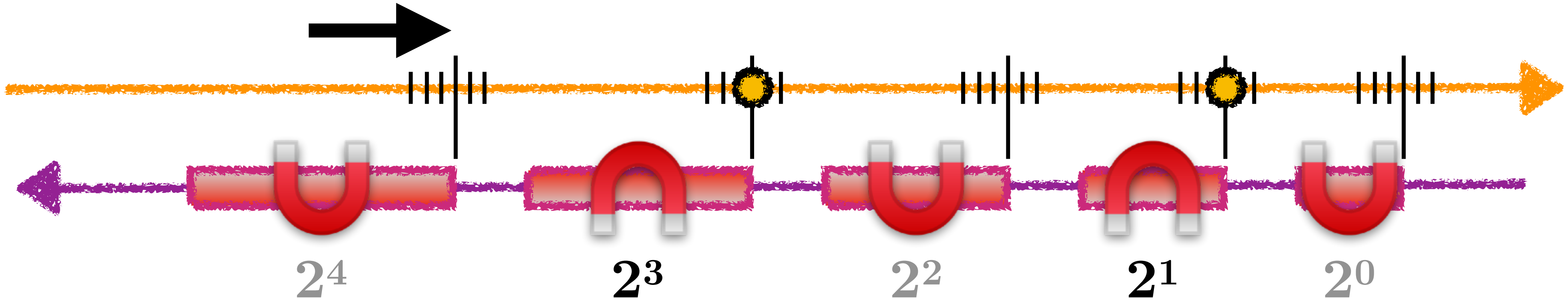
Place anchors at 2^3 and 2^1 at positions shifted by by $2(Qx + y)$

Read-Write mechanism

Place the anchors as follows:

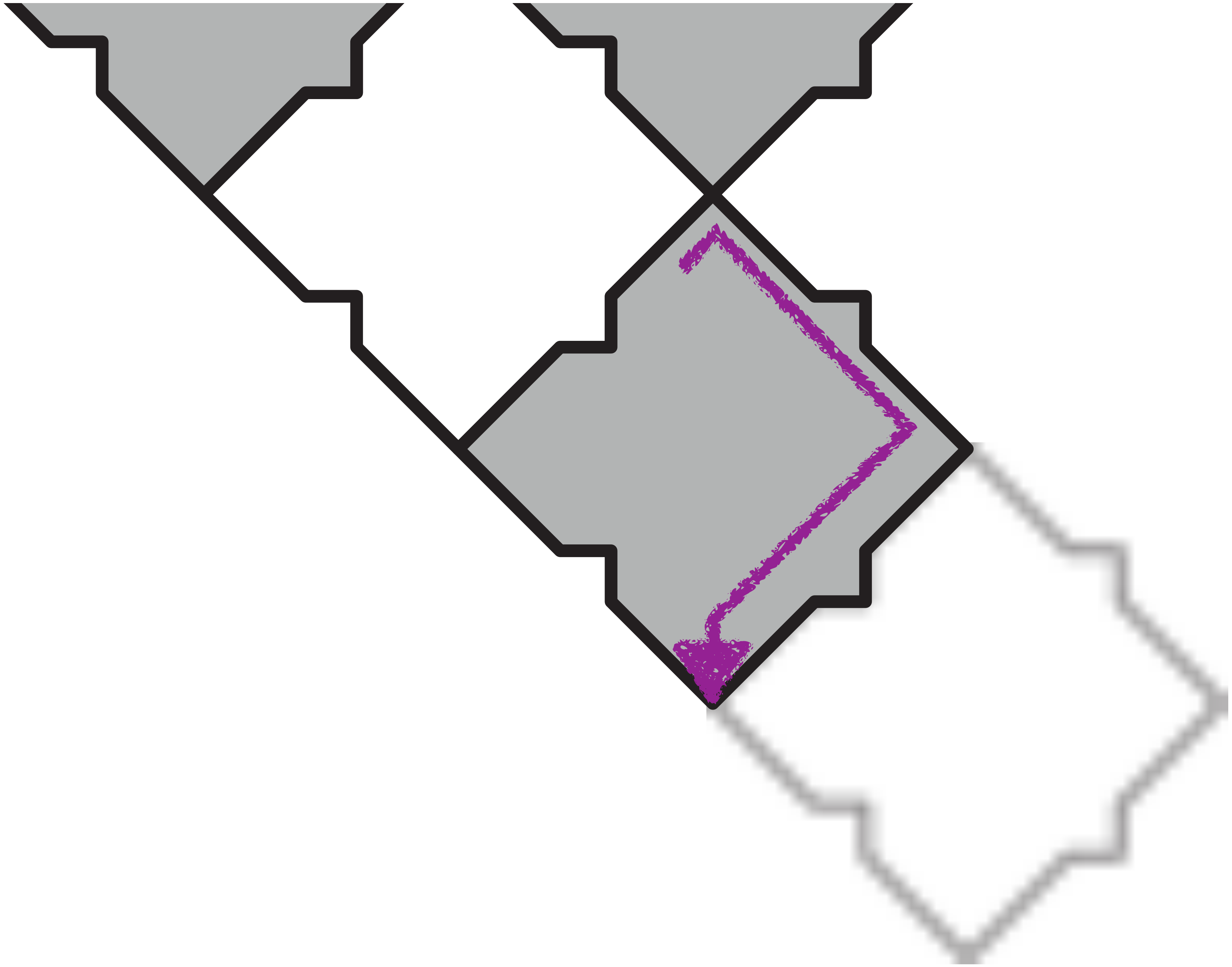
Consider $x'(x,y) = \mathbf{10} = \mathbf{2^3} + \mathbf{2^1}$

Shift by $\mathbf{2}(Qx + y)$



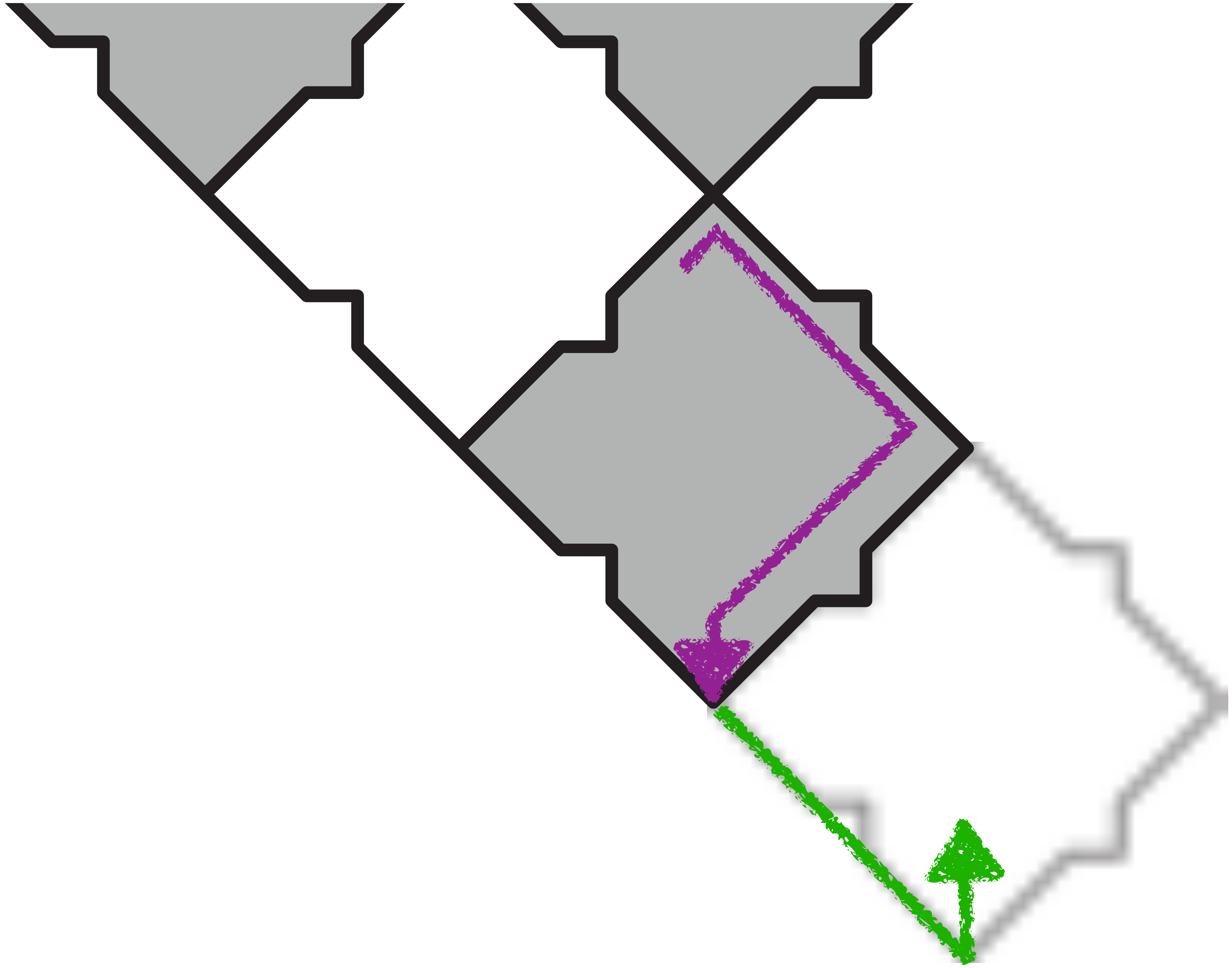
Place anchors at 2^3 and 2^1 at positions shifted by by $2(Qx + y)$

Expanding the configuration



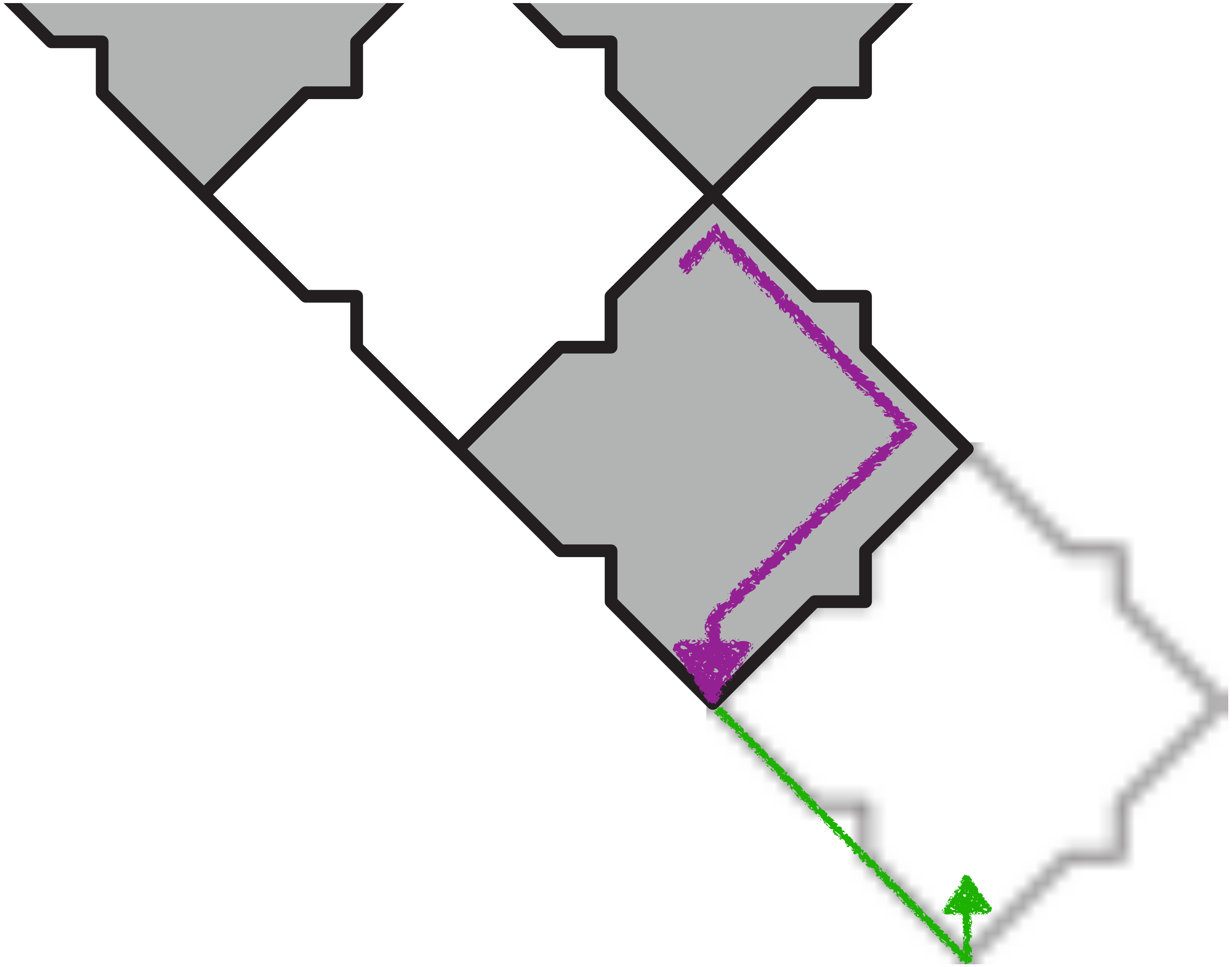
1. "Init" unfolds to build the new cell and miror direction

Expanding the configuration



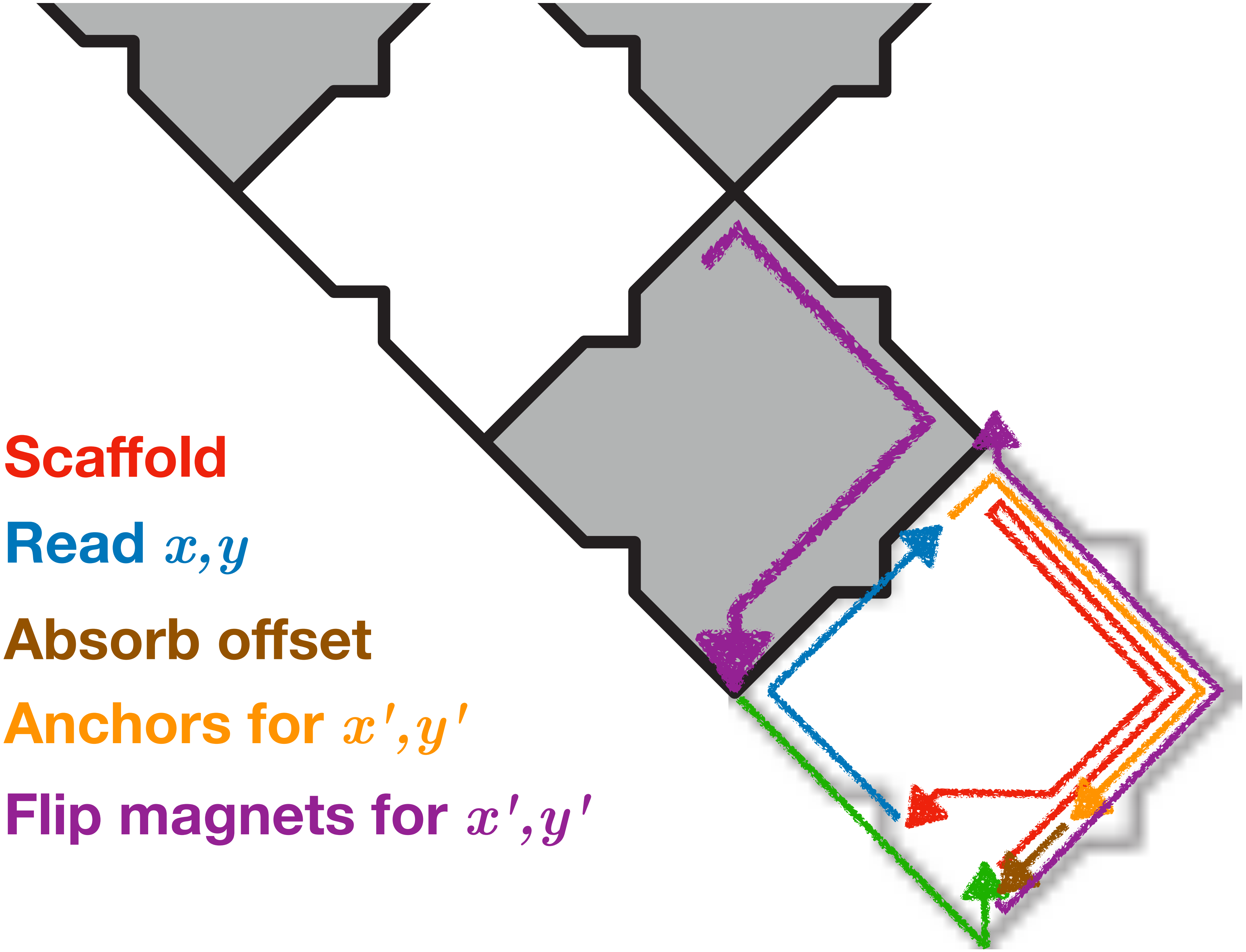
1. "Init" unfolds to build the new cell and miror direction

Expanding the configuration



1. "Init" unfolds to build the new cell and miror direction

Expanding the configuration

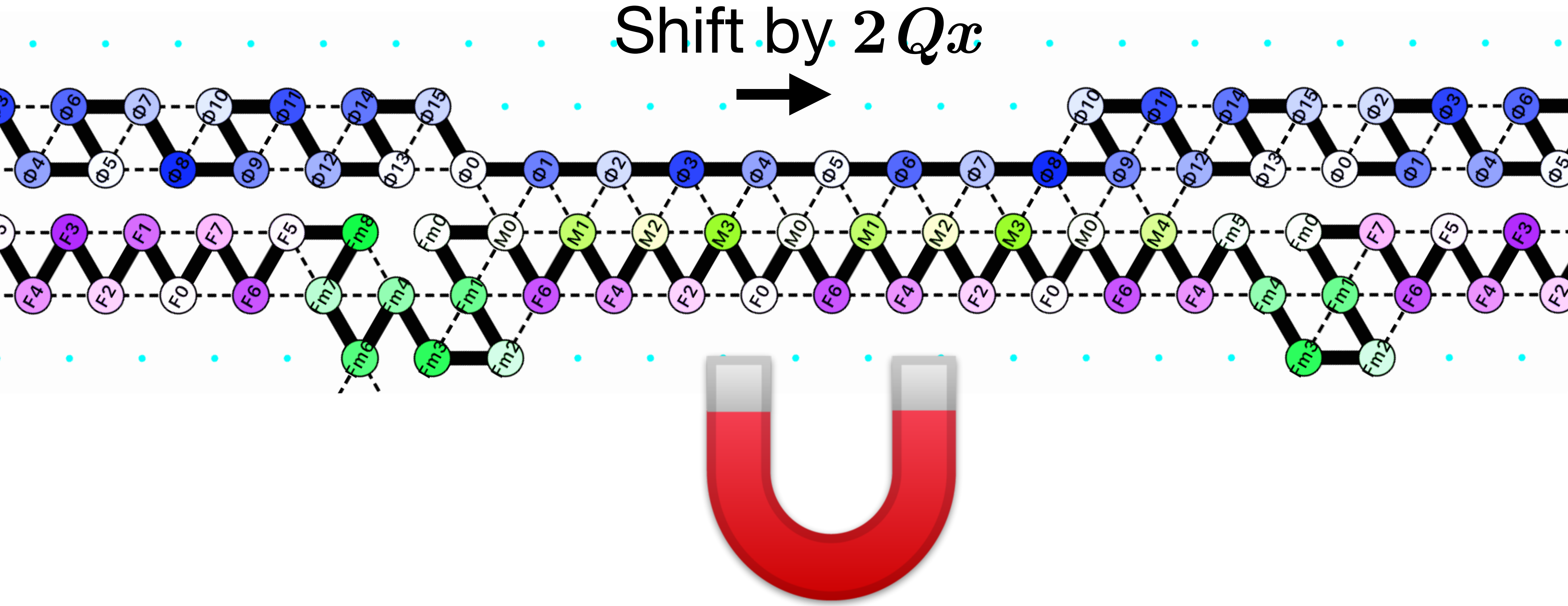


2. Scaffold
3. Read x,y
5. Absorb offset
4. Anchors for x',y'
6. Flip magnets for x',y'

1. "Init" unfolds to build the new cell and miror direction

Getting hands dirty:

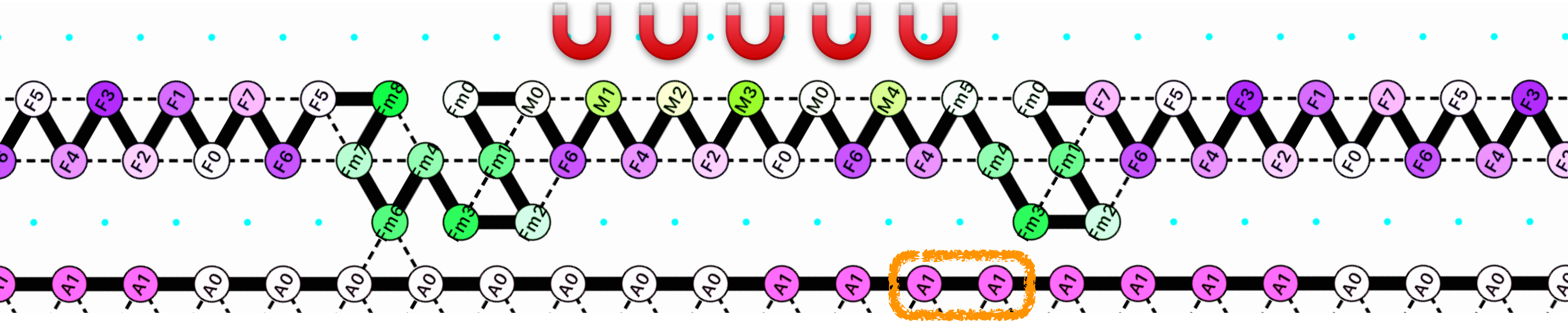
Read > Offset


Shift by $2Qx$ →

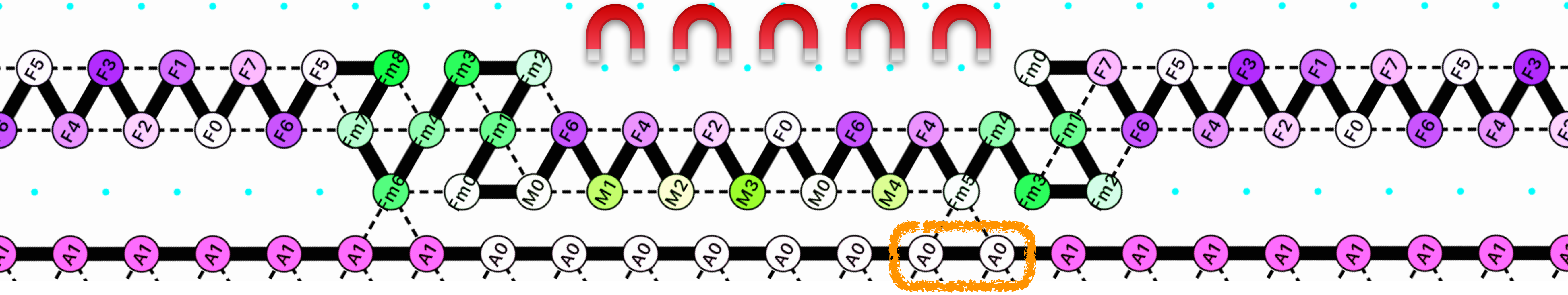
Φ6 Φ7 Φ10 Φ11 Φ14 Φ15 Φ10 Φ11 Φ14 Φ15 Φ2 Φ3 Φ6
Φ4 Φ5 Φ8 Φ9 Φ12 Φ13 Φ0 Φ1 Φ2 Φ3 Φ4 Φ5 Φ6 Φ7 Φ8 Φ9 Φ12 Φ13 Φ0 Φ1 Φ4 Φ5
F3 F1 F7 F5 Fm8 Fm0 M0 M1 M2 M3 M0 M1 M2 M3 M0 M4 Fm5 Fm0 F7 F5 F3
F4 F2 F0 F6 Fm7 Fm4 Fm1 F6 F4 F2 F0 F6 F4 F2 F0 F6 F4 Fm4 Fm1 F6 F4 F2
Fm6 Fm3 Fm2 Fm3 Fm2

Getting hands dirty:

Write: A0-Anchor flips the magnet


F5 F3 F1 F7 F5 Fm8 Fm0 M0 M1 M2 M3 M0 M4 Fm5 Fm0 F7 F5 F3 F1 F7 F5 F3
F4 F2 F0 F6 Fm7 Fm4 Fm1 F6 F4 F2 F0 F6 F4 Fm4 Fm1 F6 F4 F2 F0 F6 F4
Fm6 Fm3 Fm2 Fm3 Fm2
A1 A1 A0 A0 A0 A0 A0 A0 A0 A0 A1 A1 A1 A1 A1 A1 A1 A1 A0 A0 A0

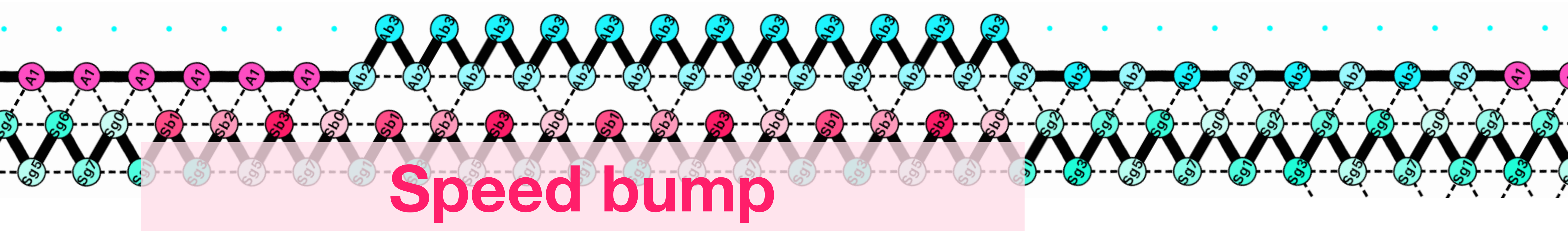
No anchor


F5 F3 F1 F7 F5 Fm8 Fm3 Fm2 Fm0 F7 F5 F3 F1 F7 F5 F3
F4 F2 F0 F6 Fm7 Fm4 Fm1 F6 F4 F2 F0 F6 F4 Fm4 Fm1 F6 F4 F2 F0 F6 F4
Fm6 Fm0 M0 M1 M2 M3 M0 M4 Fm5 Fm3 Fm2
A1 A1 A1 A1 A1 A1 A0 A0 A0 A0 A0 A0 A0 A0 A1 A1 A1 A1 A1 A1 A1

Anchor

Getting hands dirty:
Absorbing Offset

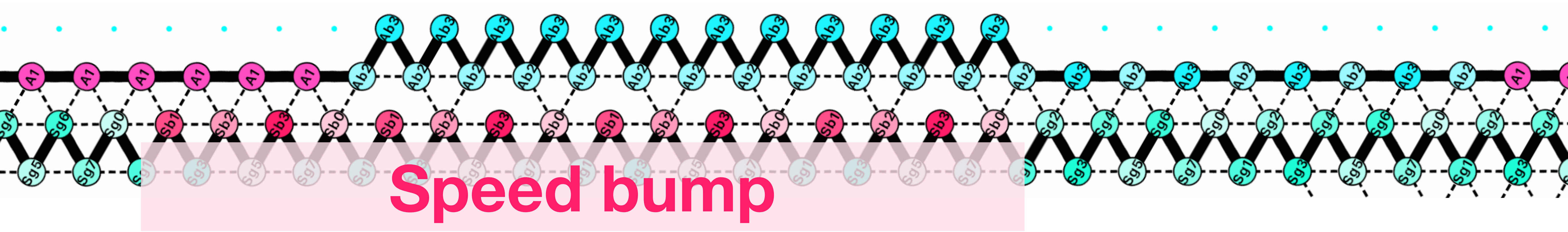
Offset divided by 2



Speed bump

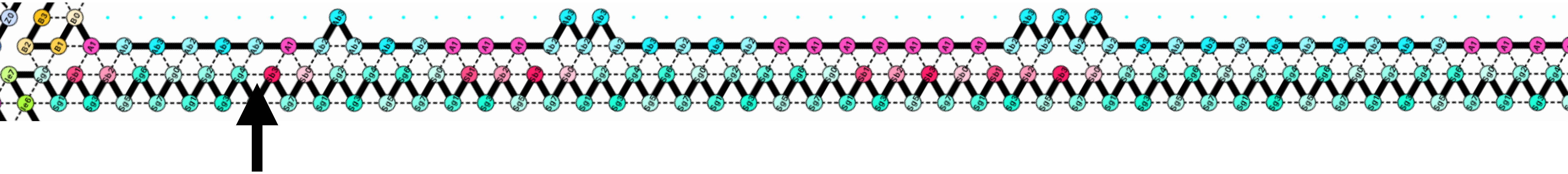
Getting hands dirty:
Absorbing Offset

Offset divided by 2



Speed bump

Repeat log(Max offset) times!

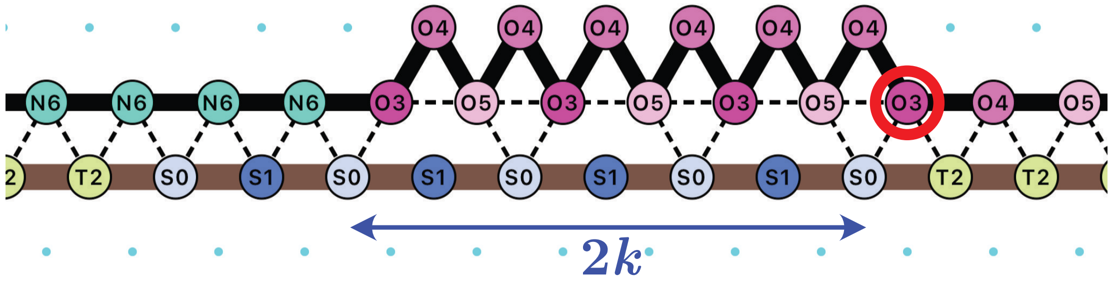


Synchronized!

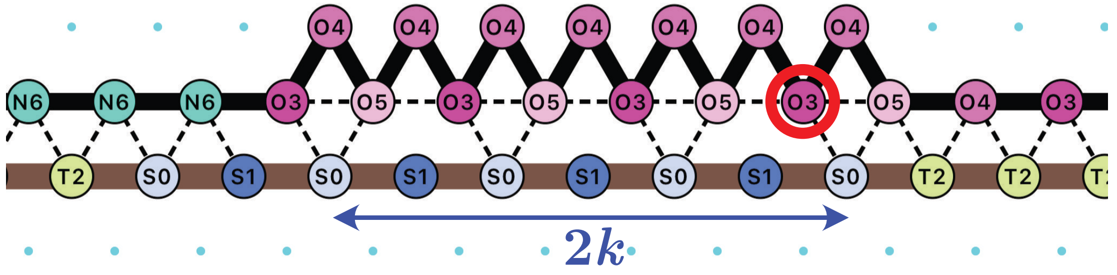
Getting hands dirty:
Absorbing Offset

Here, $k = 3$

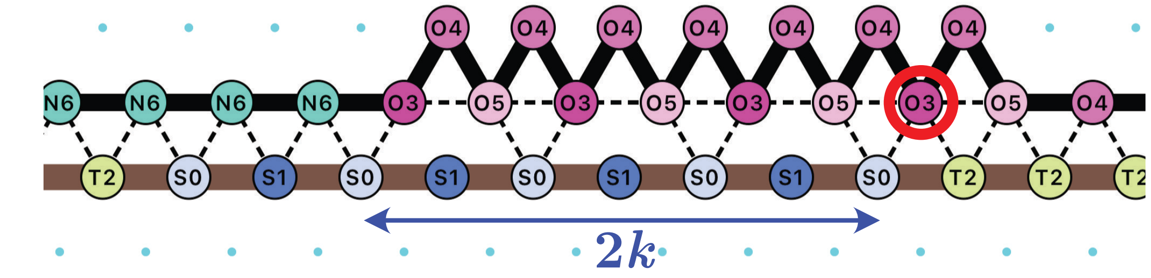
$\Delta = 4k + 0$ then $\Delta' = \Delta - 2k = \Delta/2$


O4 O4 O4 O4 O4 O4
N6 N6 N6 N6 O3 O5 O3 O5 O3 O5 O3 O4 O5
2 T2 S0 S1 S0 S1 S0 S1 S0 S1 S0 T2 T2
$2k$

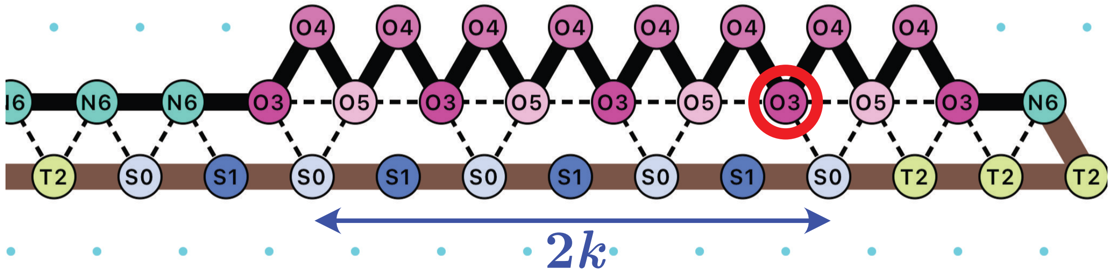
$\Delta = 4k + 2$ then $\Delta' = \Delta - 2k - 1 = \Delta/2$


O4 O4 O4 O4 O4 O4 O4
N6 N6 N6 O3 O5 O3 O5 O3 O5 O3 O5 O4 O3
T2 S0 S1 S0 S1 S0 S1 S0 S1 S0 T2 T2 T
$2k$

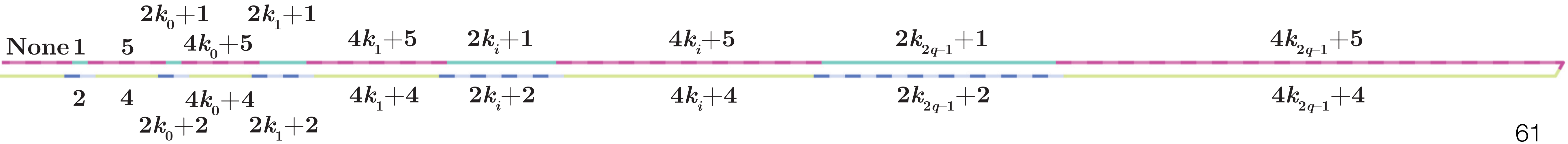
$\Delta = 4k + 1$ then $\Delta' = \Delta - 2k - 1 = [\Delta/2]$


O4 O4 O4 O4 O4 O4 O4
N6 N6 N6 N6 O3 O5 O3 O5 O3 O5 O3 O5 O4
T2 S0 S1 S0 S1 S0 S1 S0 S1 S0 T2 T2 T2
$2k$

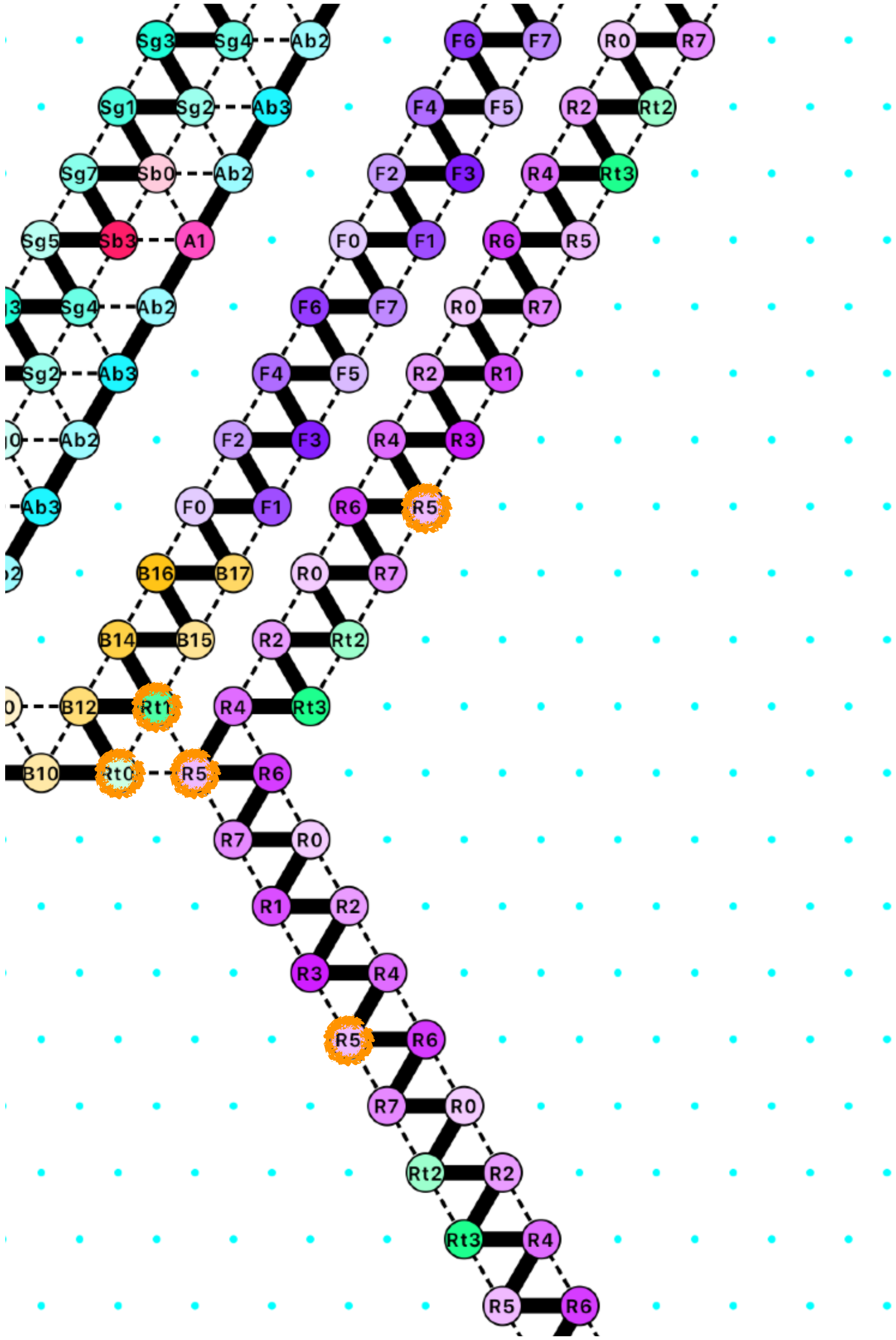
$\Delta = 4k + 3$ then $\Delta' = \Delta - 2k - 2 = [\Delta/2]$


O4 O4 O4 O4 O4 O4 O4 O4
N6 N6 N6 O3 O5 O3 O5 O3 O5 O3 O5 O3 N6
T2 S0 S1 S0 S1 S0 S1 S0 S1 S0 T2 T2 T2
$2k$

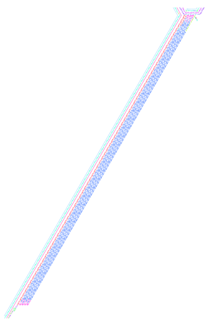
$$k_0 = 0 \text{ and } k_{i+1} = 2k_i + 1 = 2^i - 1$$



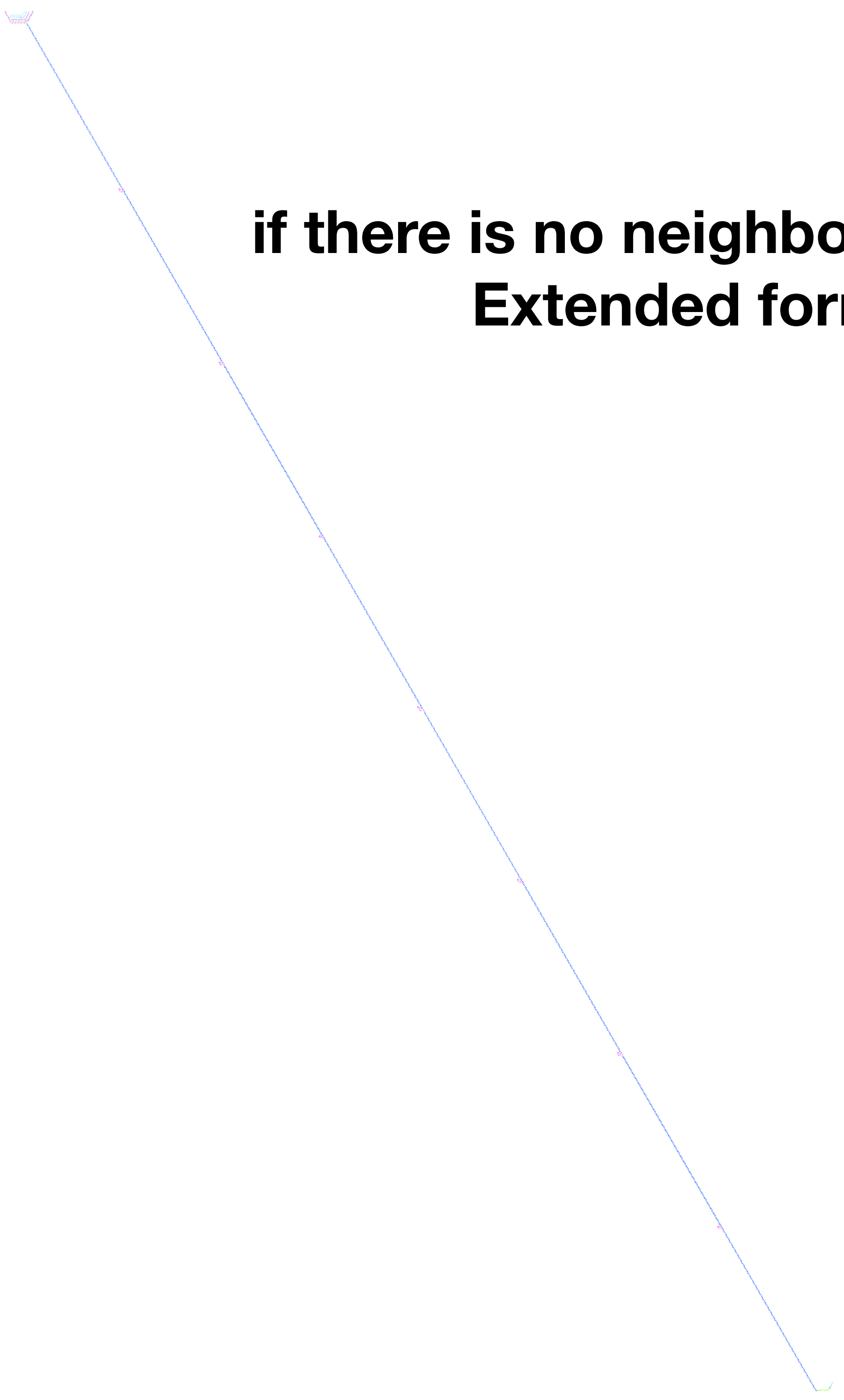
Getting hands dirty:
Turning Scaffold



Getting hands dirty:
Init: unfolding glider

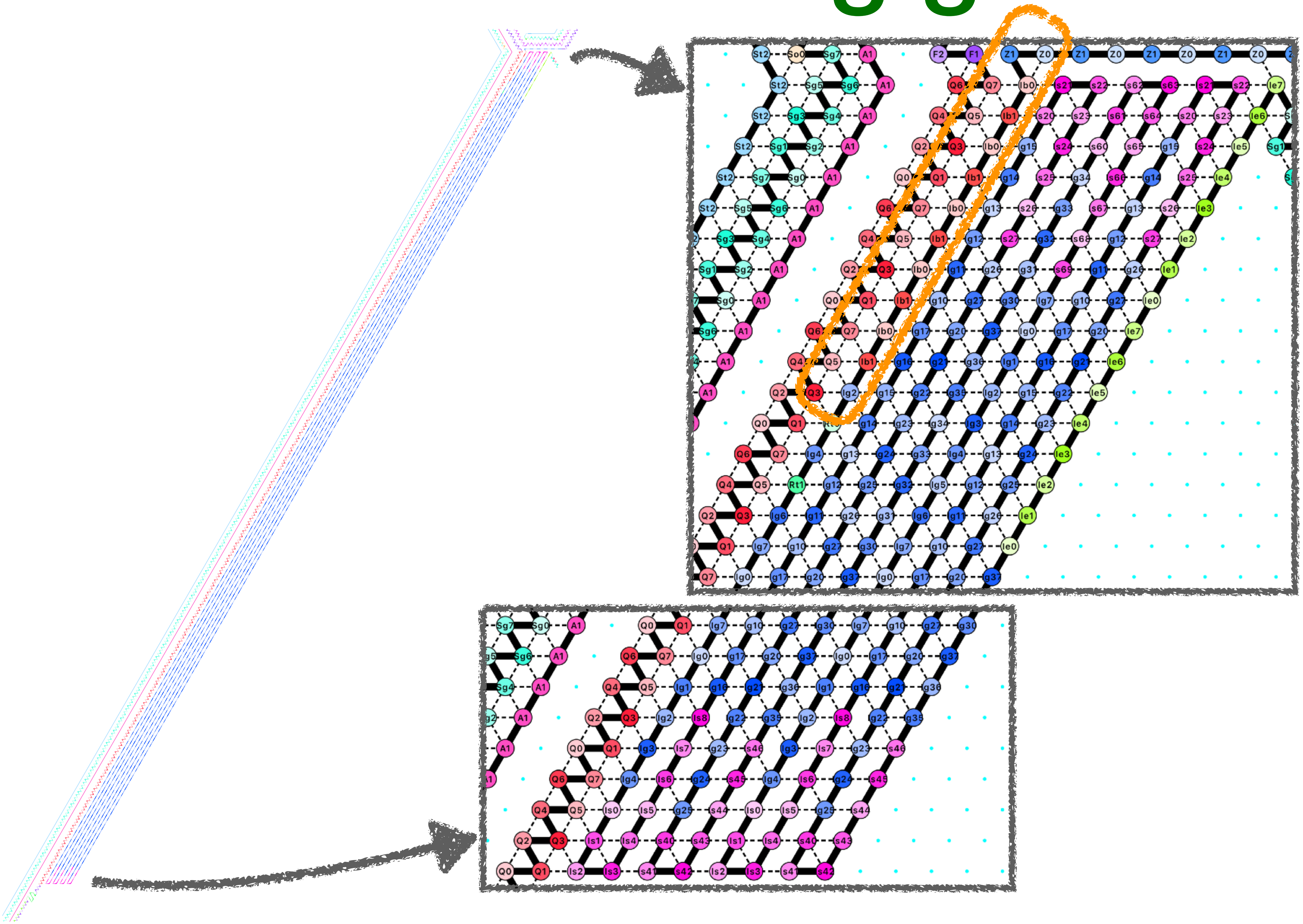


if there is a neighboring cell :
Compact form

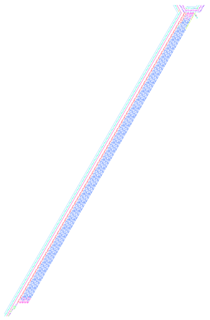


if there is no neighboring cell :
Extended form

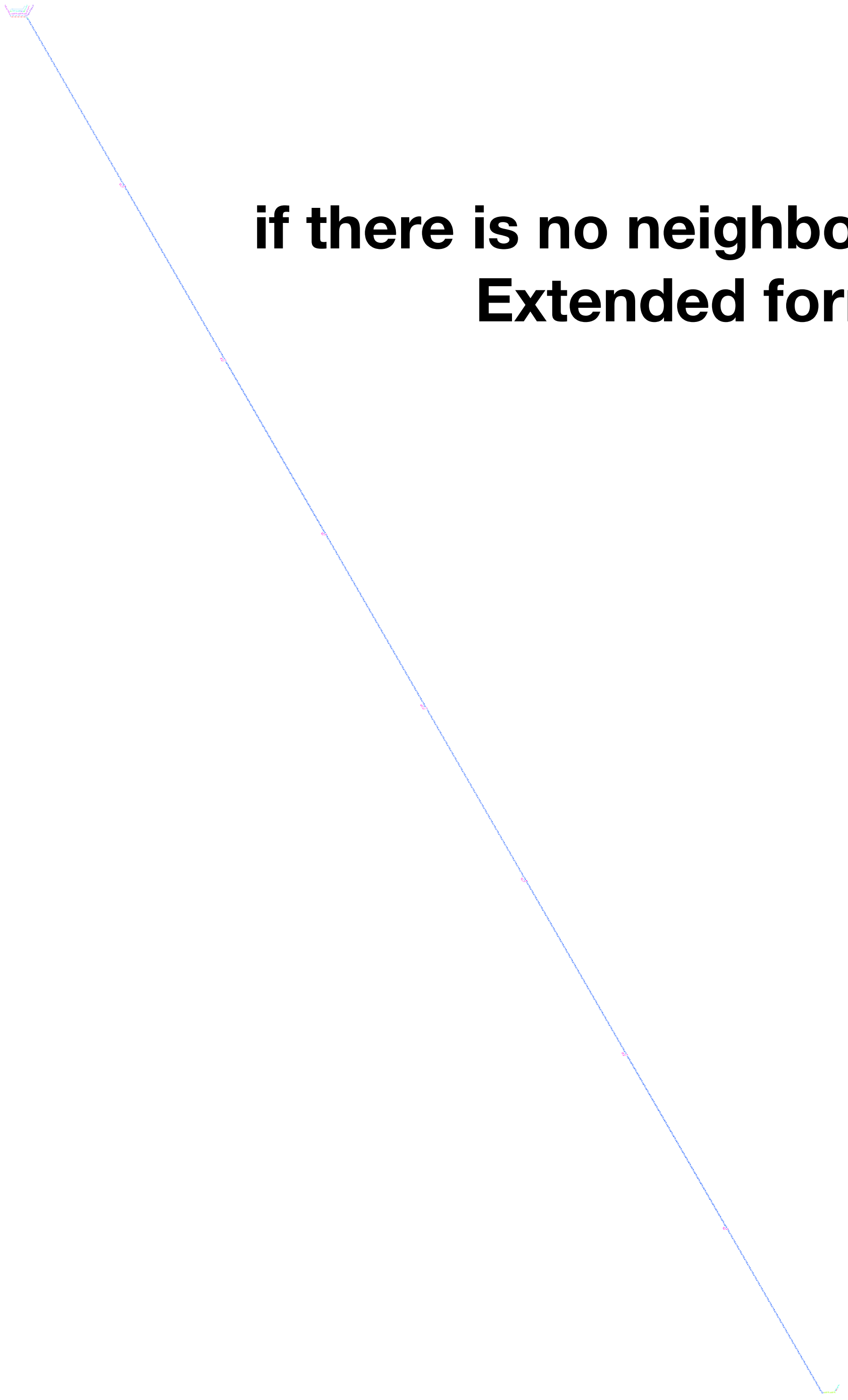
Getting hands dirty:
Init: unfolding glider



Getting hands dirty:
Init: unfolding glider

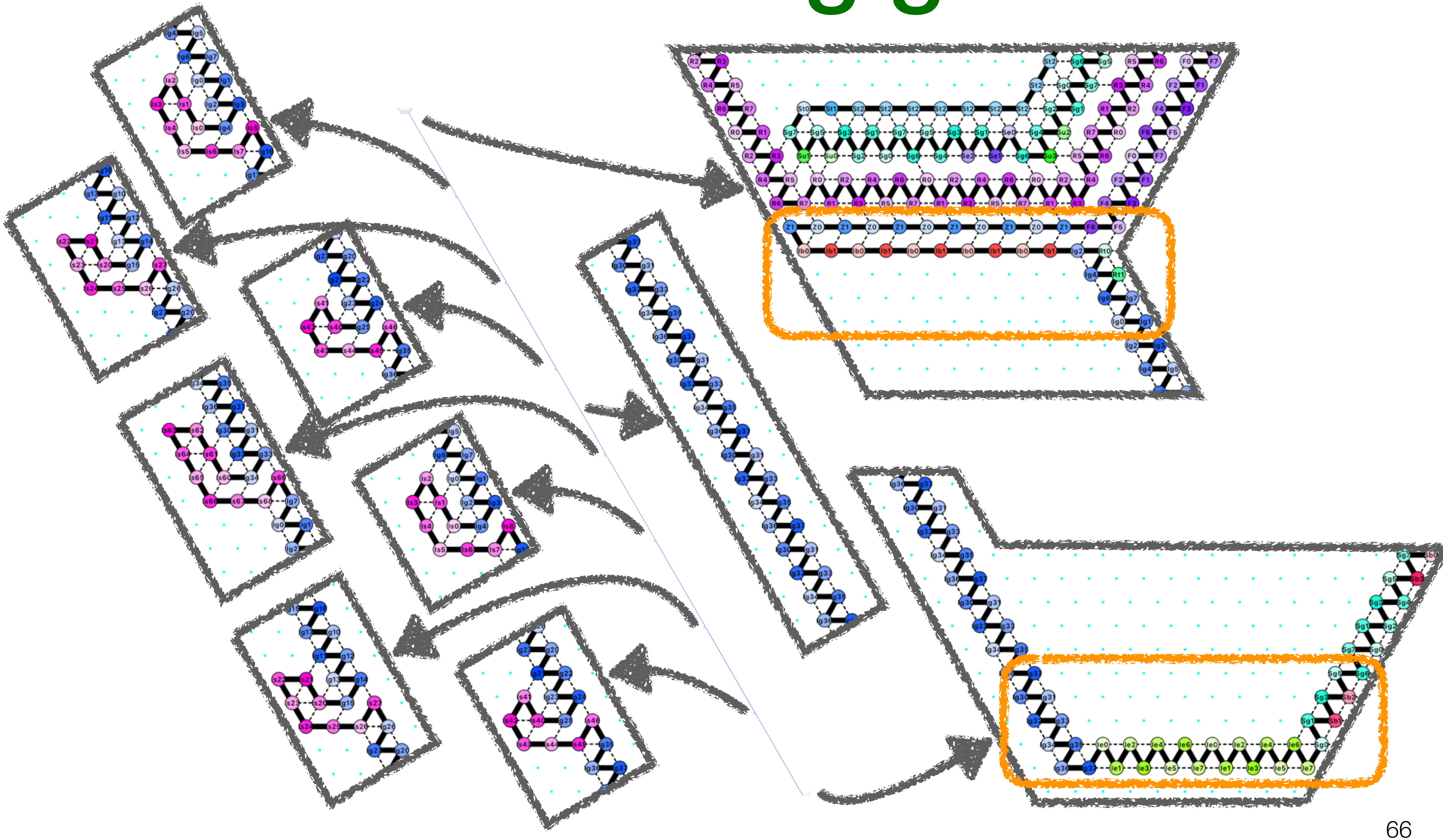


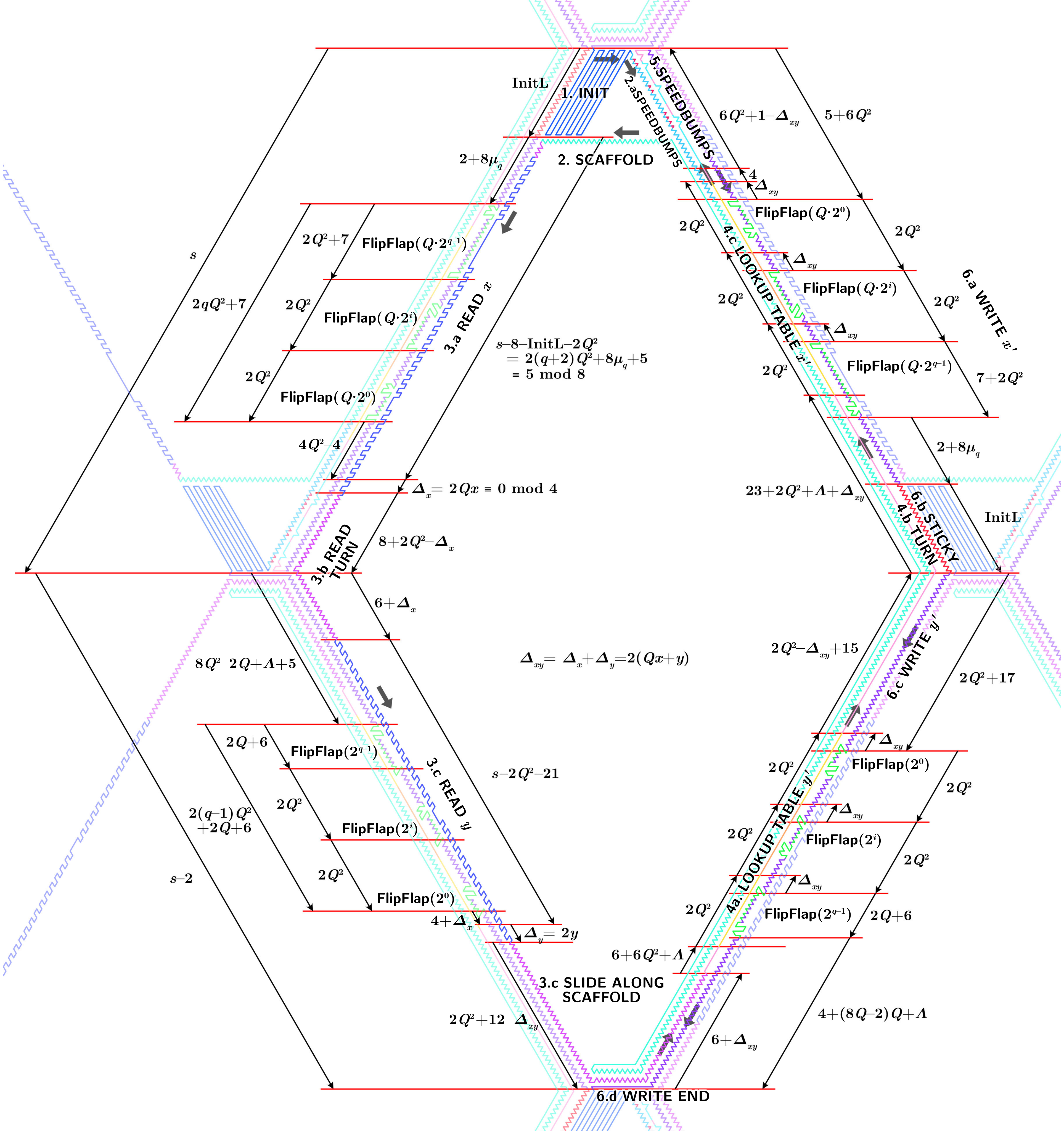
if there is a neighboring cell :
Compact form



if there is no neighboring cell :
Extended form

Getting hands dirty:
Init: unfolding glider




InitL
1. INIT
2.aSPEEDBUMPS
5.SPEEDBUMPS
$6Q^2+1-\Delta_{xy}$
$5+6Q^2$
$2+8\mu_q$
2. SCAFFOLD
4
Δ_{xy}
FlipFlap($Q\cdot2^0$)
$2Q^2$
$2Q^2$
$2Q^2+7$
FlipFlap($Q\cdot2^{q-1}$)
s
4.c LOOKUP TABLE x'
Δ_{xy}
FlipFlap($Q\cdot2^i$)
$2Q^2$
$2Q^2$
6.a WRITE x'
$2qQ^2+7$
$2Q^2$
FlipFlap($Q\cdot2^i$)
3.a READ x
$s-8-\text{InitL}-2Q^2$
$=2(q+2)Q^2+8\mu_q+5$
$\equiv 5 \bmod 8$
Δ_{xy}
FlipFlap($Q\cdot2^{q-1}$)
$2Q^2$
$7+2Q^2$
$2Q^2$
FlipFlap($Q\cdot2^0$)
$2+8\mu_q$
$4Q^2-4$
$\Delta_x = 2Qx \equiv 0 \bmod 4$
$23+2Q^2+\Lambda+\Delta_{xy}$
6.b STICKY
4.b TURN
InitL
3.b READ TURN
$8+2Q^2-\Delta_x$
$6+\Delta_x$
$2Q^2-\Delta_{xy}+15$
6.c WRITE y'
$\Delta_{xy} = \Delta_x+\Delta_y = 2(Qx+y)$
$8Q^2-2Q+\Lambda+5$
$2Q^2+17$
$2Q+6$
FlipFlap(2^{q-1})
Δ_{xy}
FlipFlap(2^0)
$2Q^2$
$s-2Q^2-21$
3.c READ y
$2Q^2$
$2Q^2$
$2(q-1)Q^2$
$+2Q+6$
FlipFlap(2^i)
$2Q^2$
4a. LOOKUP TABLE y'
Δ_{xy}
FlipFlap(2^i)
$2Q^2$
$s-2$
$2Q^2$
FlipFlap(2^0)
$2Q^2$
Δ_{xy}
FlipFlap(2^{q-1})
$2Q+6$
$4+\Delta_x$
$\Delta_y = 2y$
$6+6Q^2+\Lambda$
3.c SLIDE ALONG
SCAFFOLD
$2Q^2+12-\Delta_{xy}$
$6+\Delta_{xy}$
$4+(8Q-2)Q+\Lambda$
6.d WRITE END

Conclusion

Our results

- Oritatami system can simulate intrinsically any 1D cellular automata
- "Mechanical" tools for designing simpler oritatami system

Next…

- An oritatami programming language?
- How to implement RAM? Loops? Concatenation? Subroutine call?
- Design a program simple enough to be *implemented in wet-lab?*