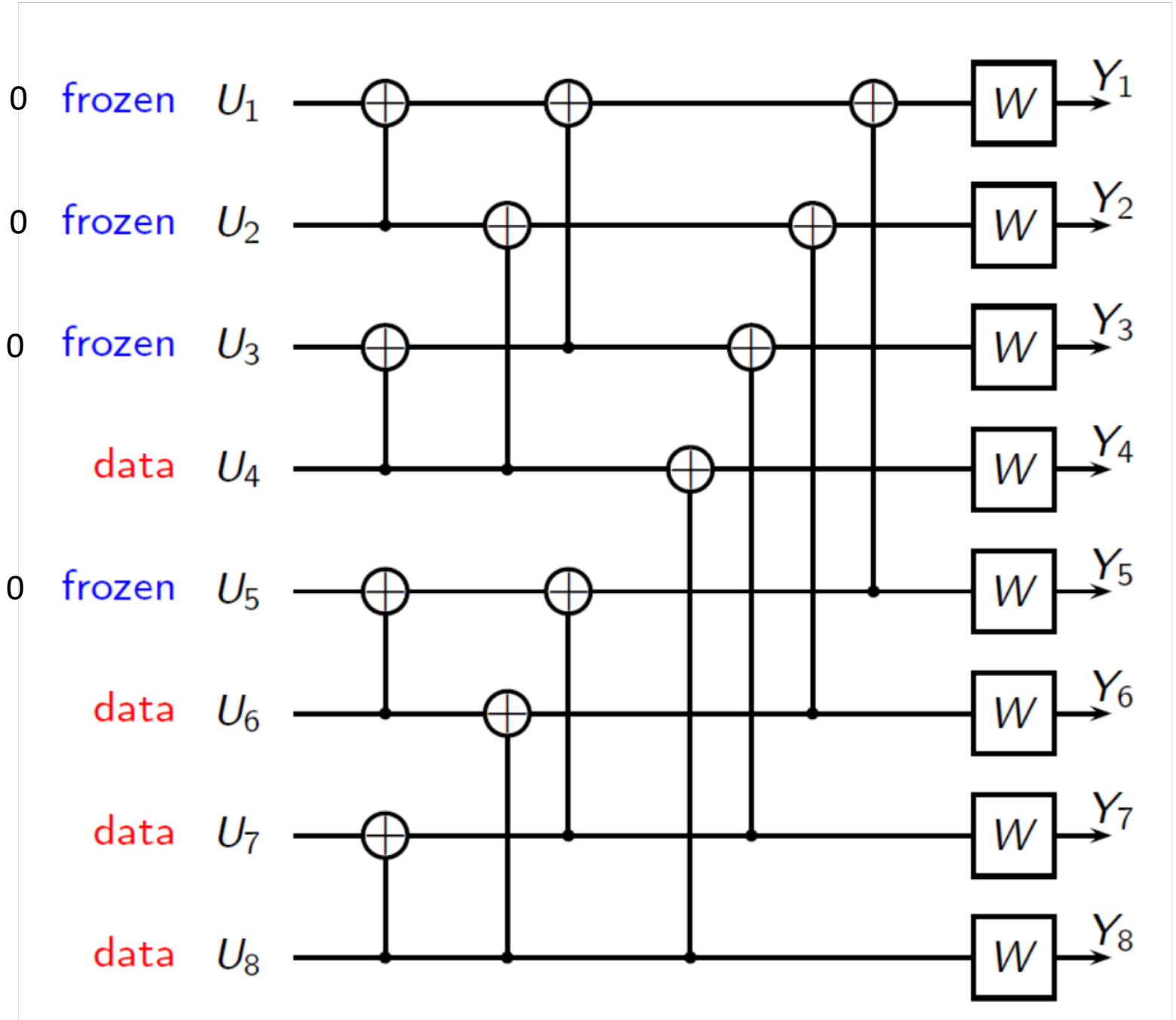
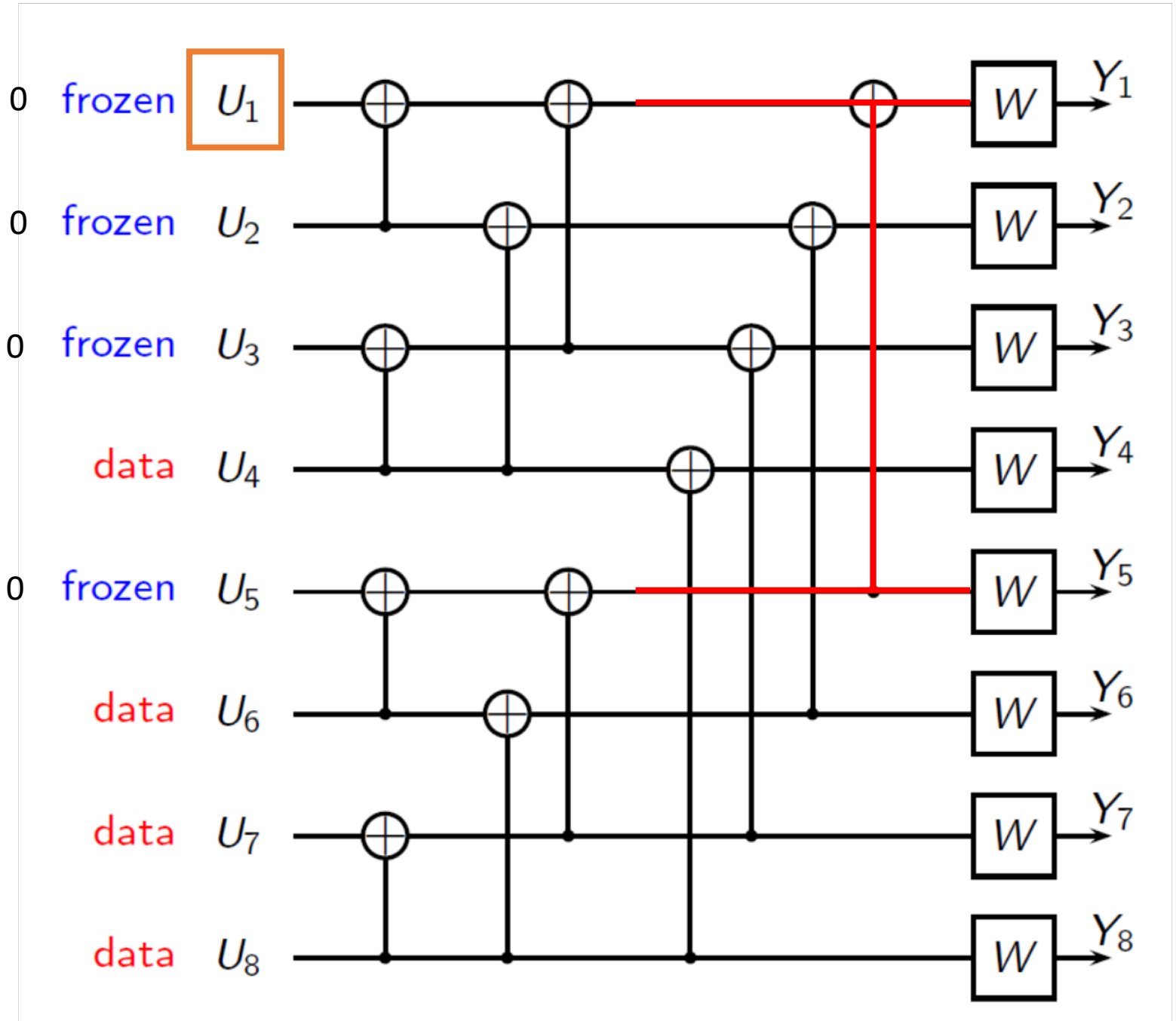
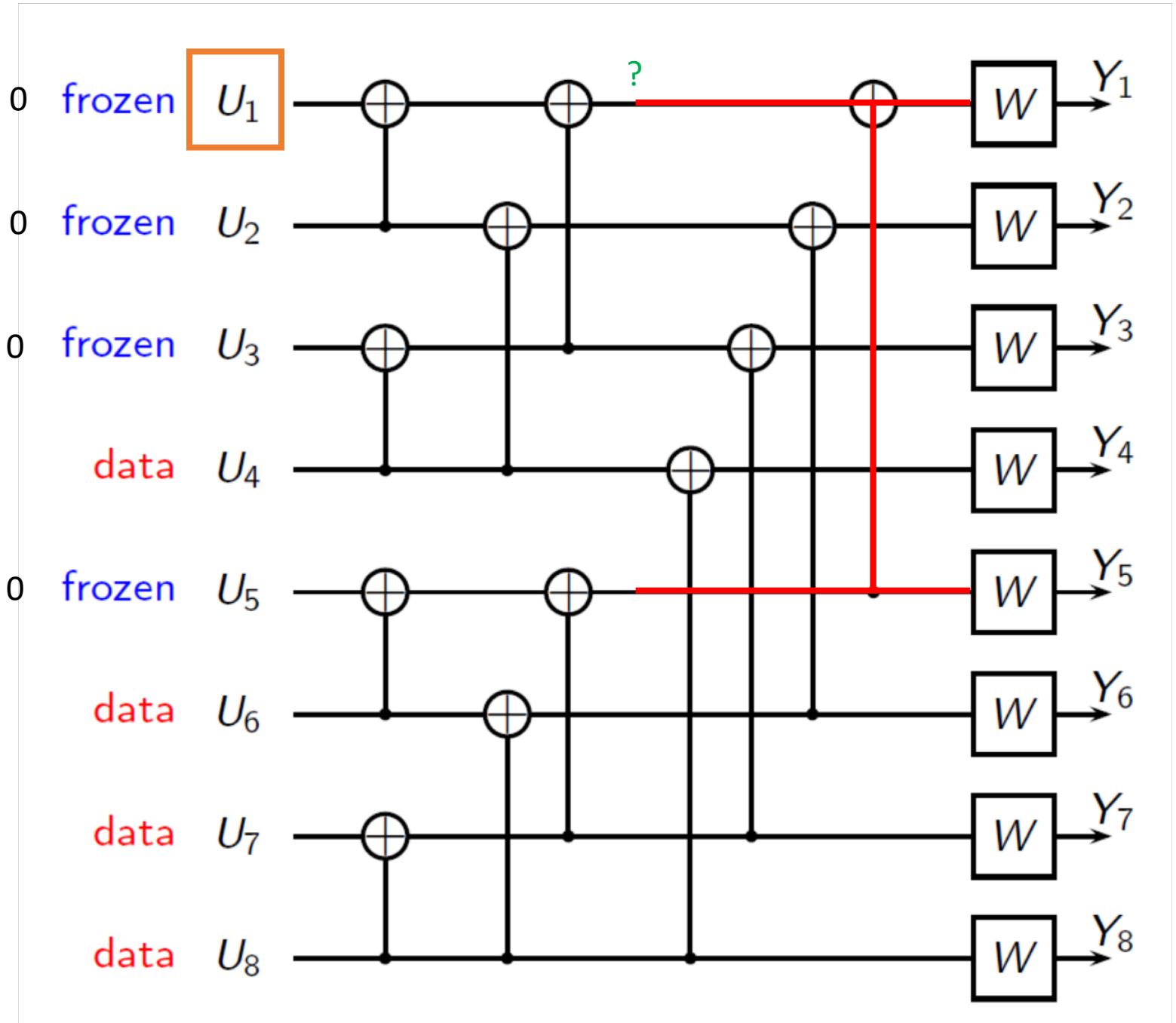
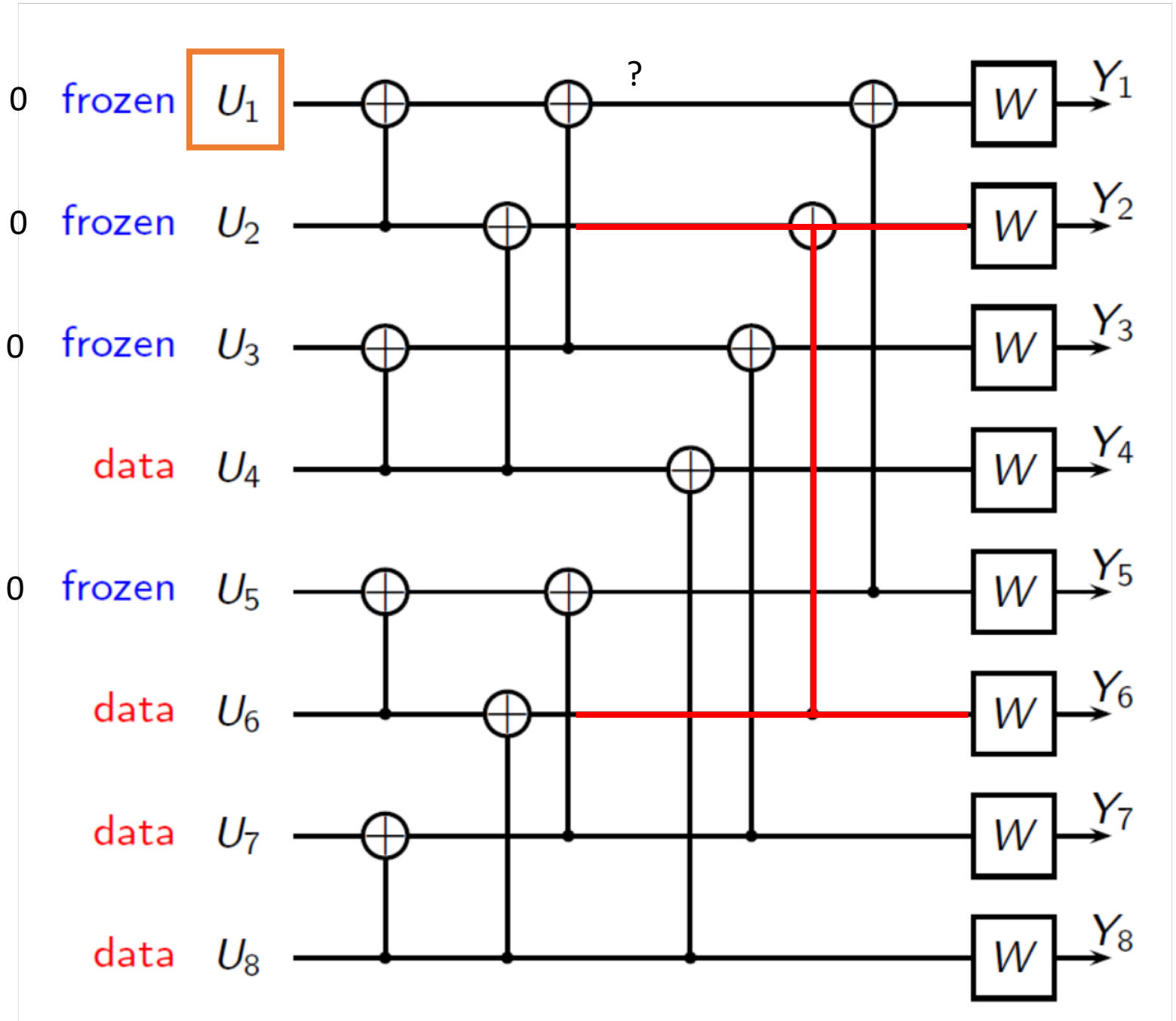
Polar code successive cancellation decoding example

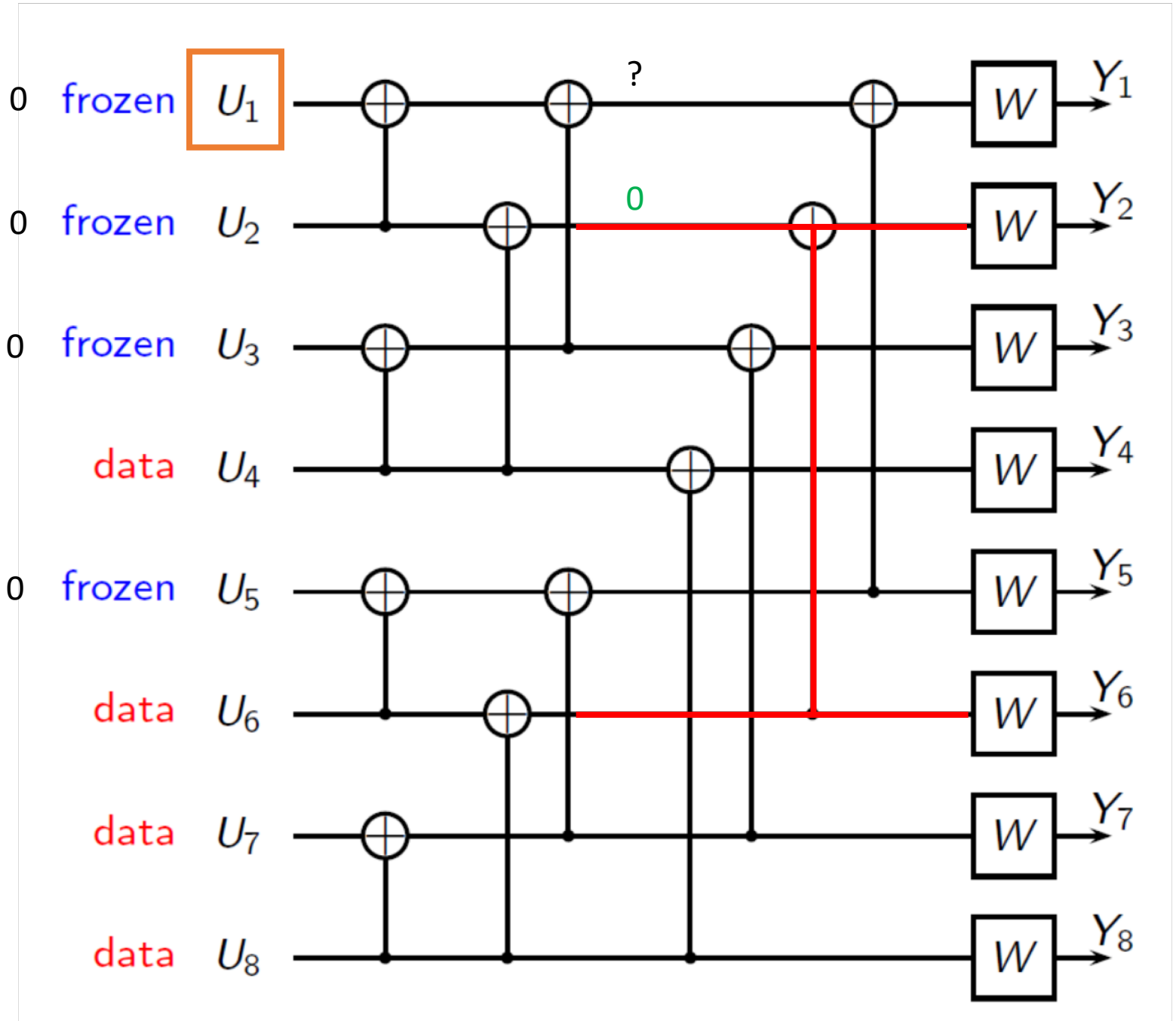
- For BEC
- Block length = 8
- Rate = ½
- Each step is the 2x2 decoding you saw in class

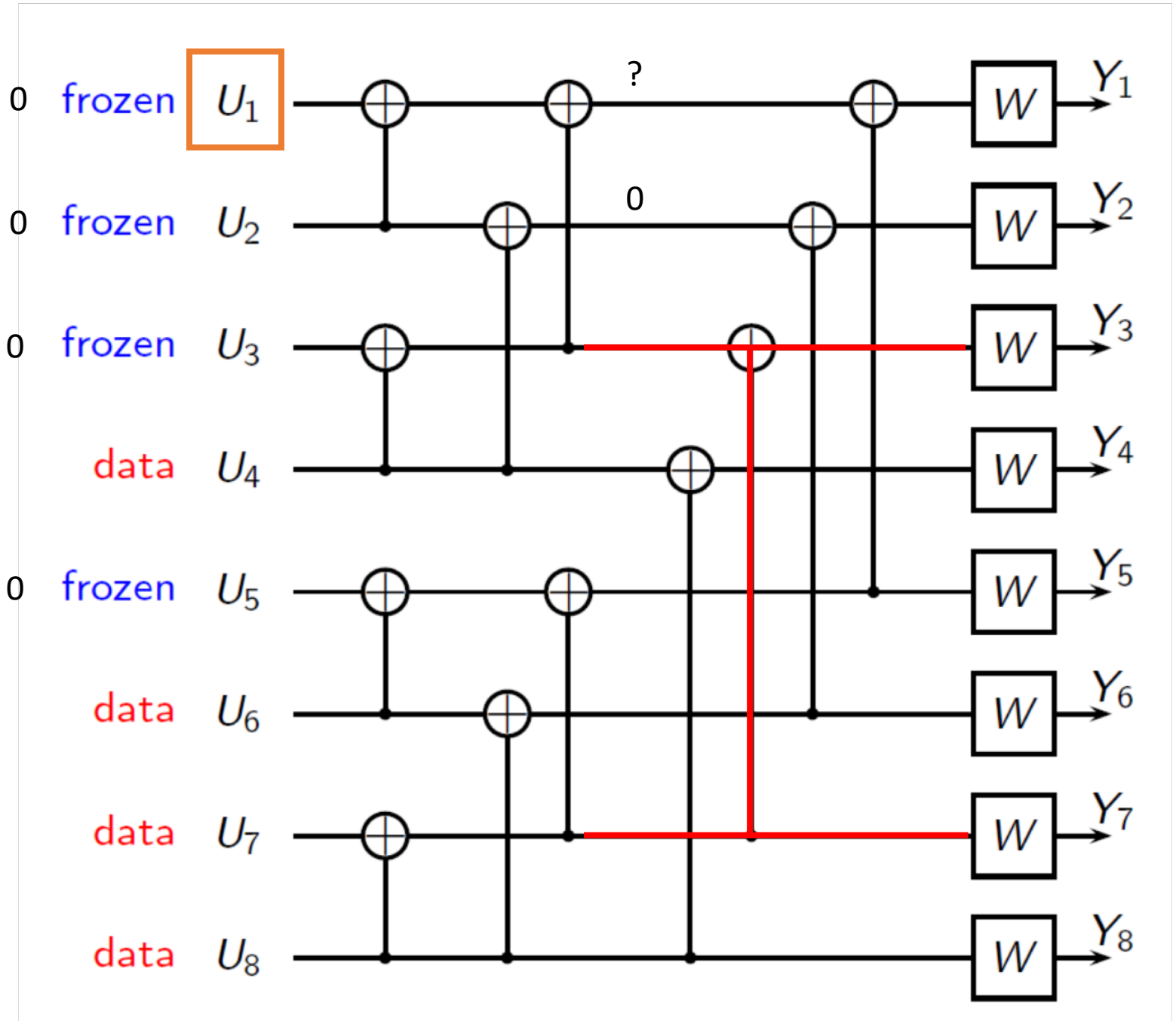

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

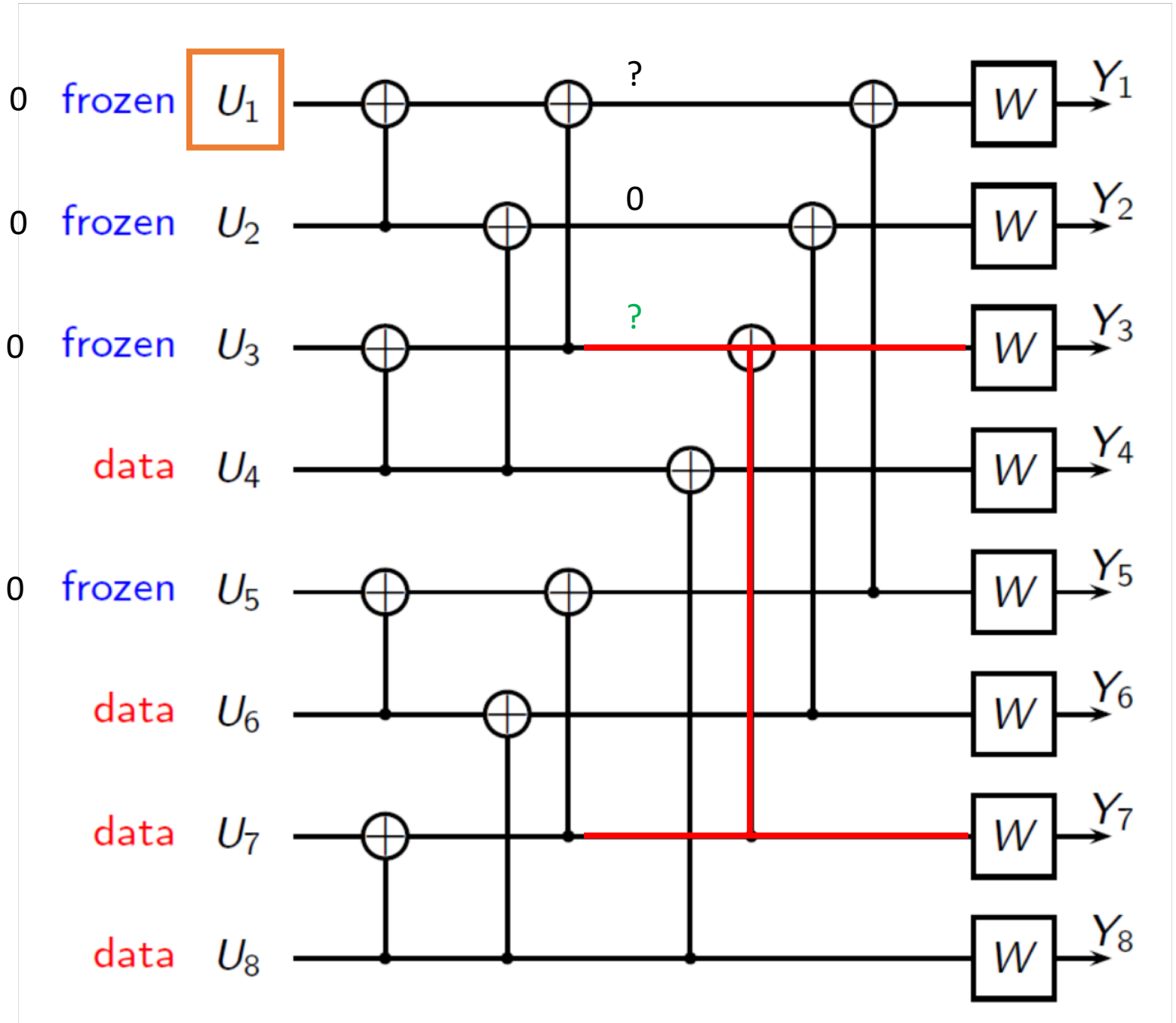

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1 ?
Y_2 0
Y_3 0
Y_4 ?
Y_5 0
Y_6 0
Y_7 ?
Y_8 0

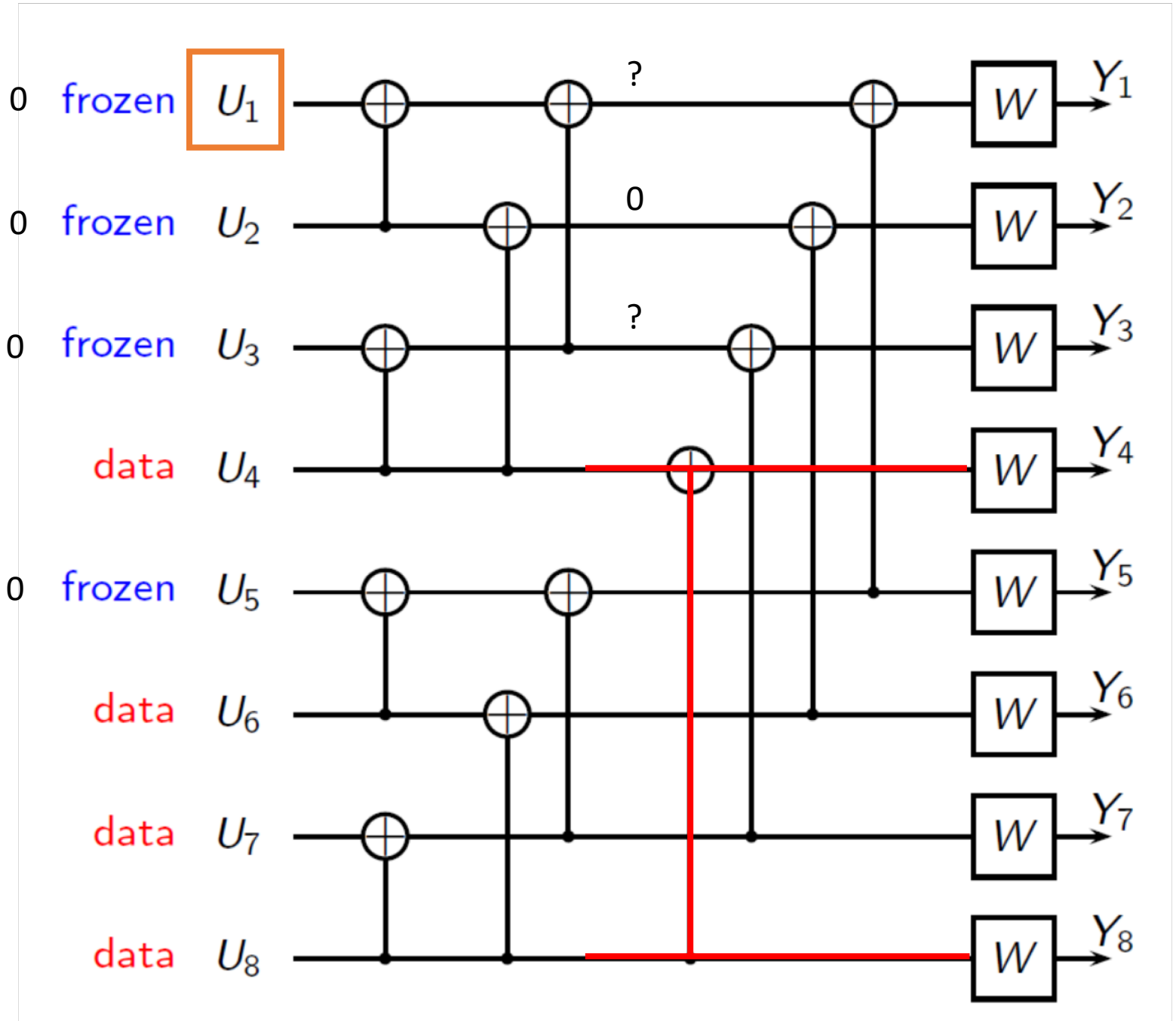

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

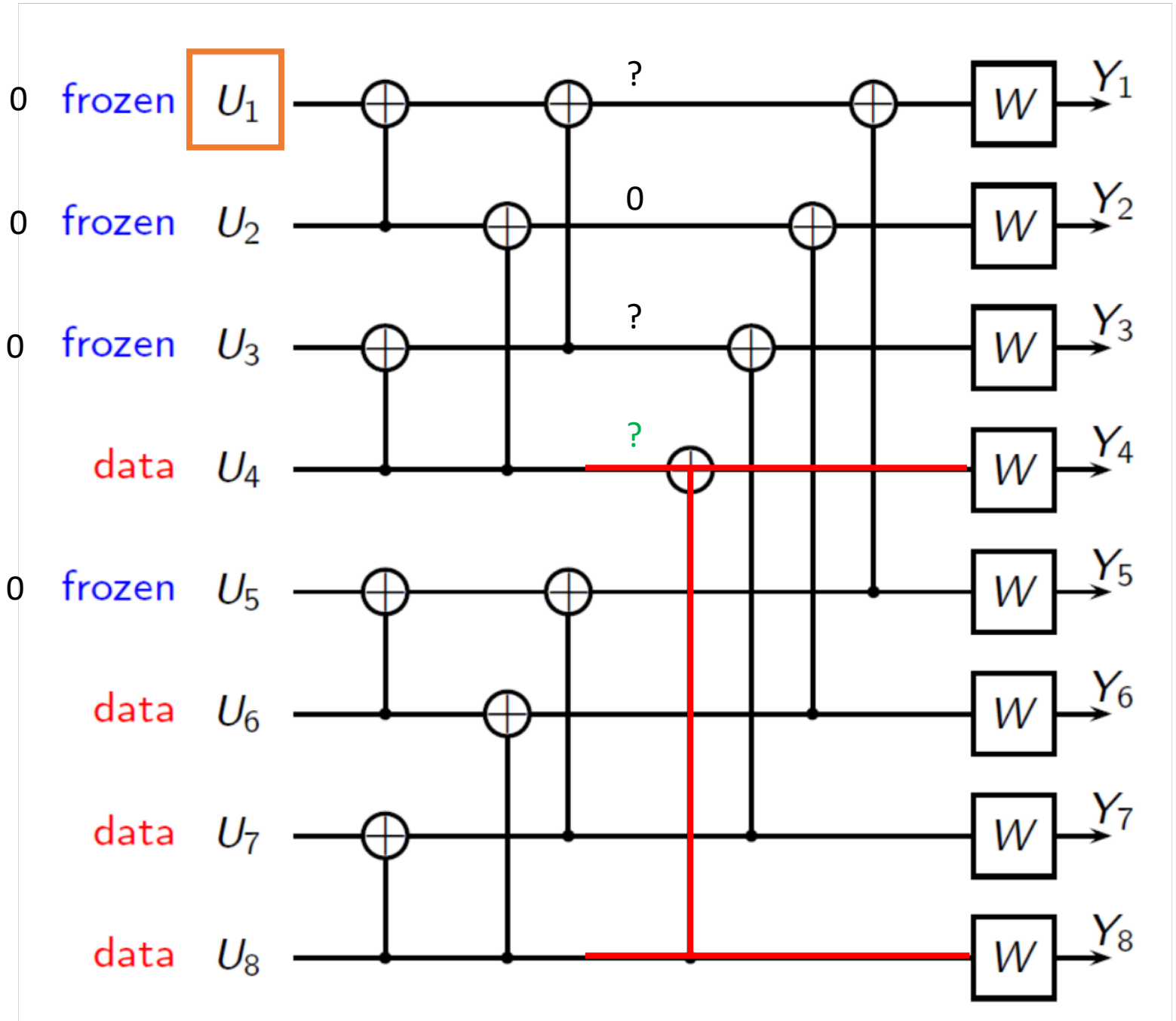

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
W Y_1 ?
W Y_2 0
W Y_3 0
W Y_4 ?
W Y_5 0
W Y_6 0
W Y_7 ?
W Y_8 0

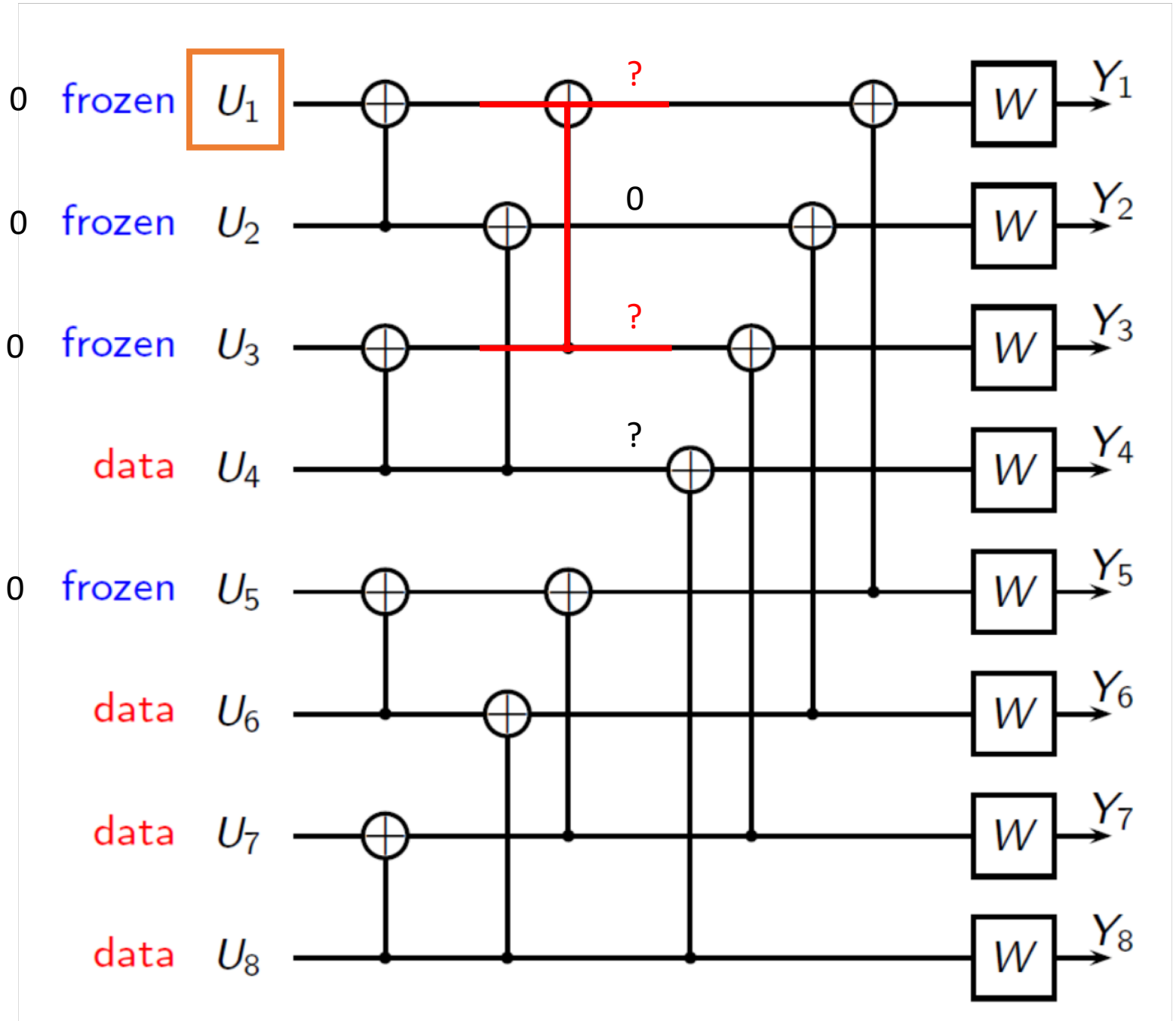

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
0
W Y_1
W Y_2
W Y_3
W Y_4
W Y_5
W Y_6
W Y_7
W Y_8
?
0
0
?
0
0
?
0

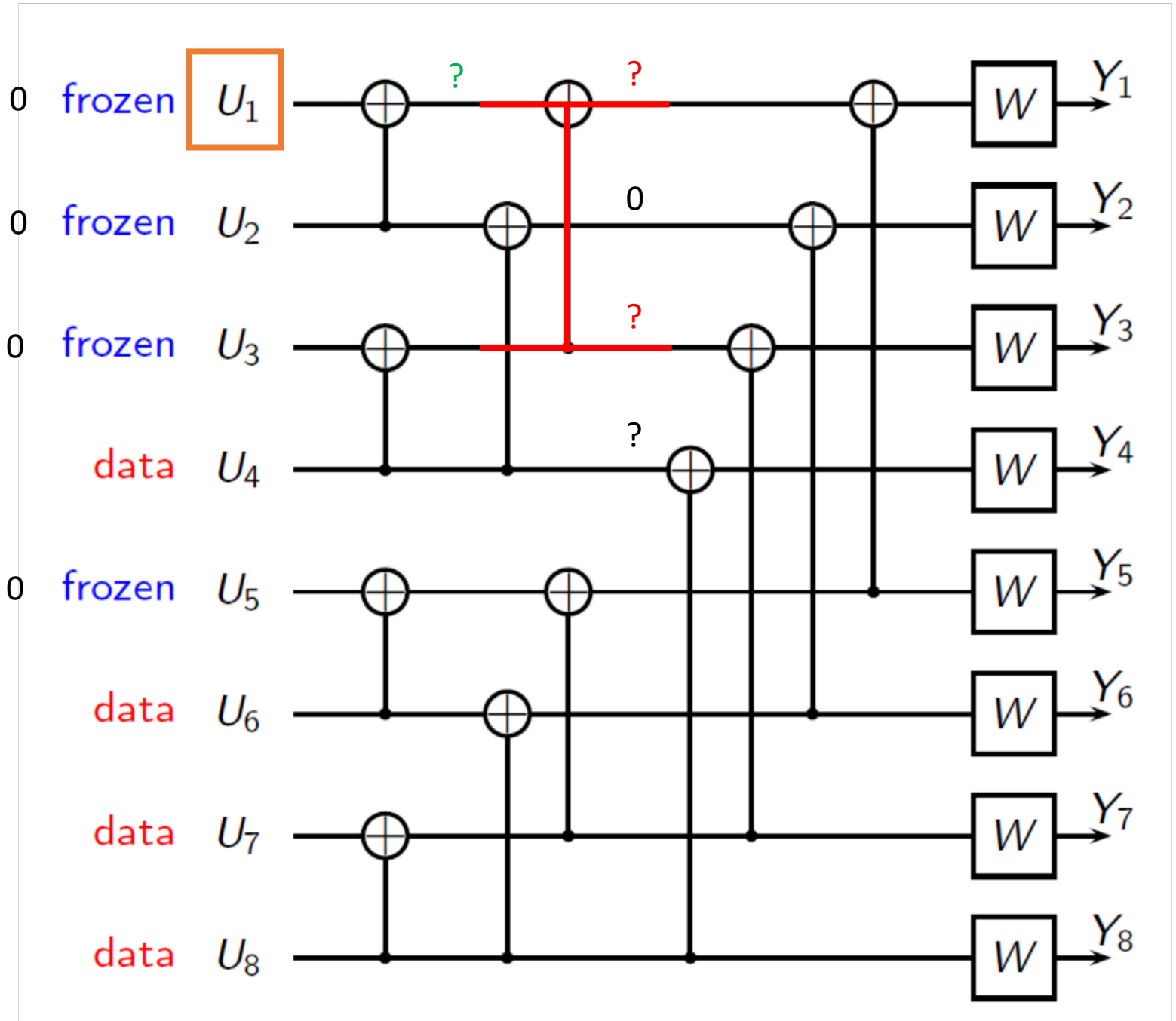

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
0
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?

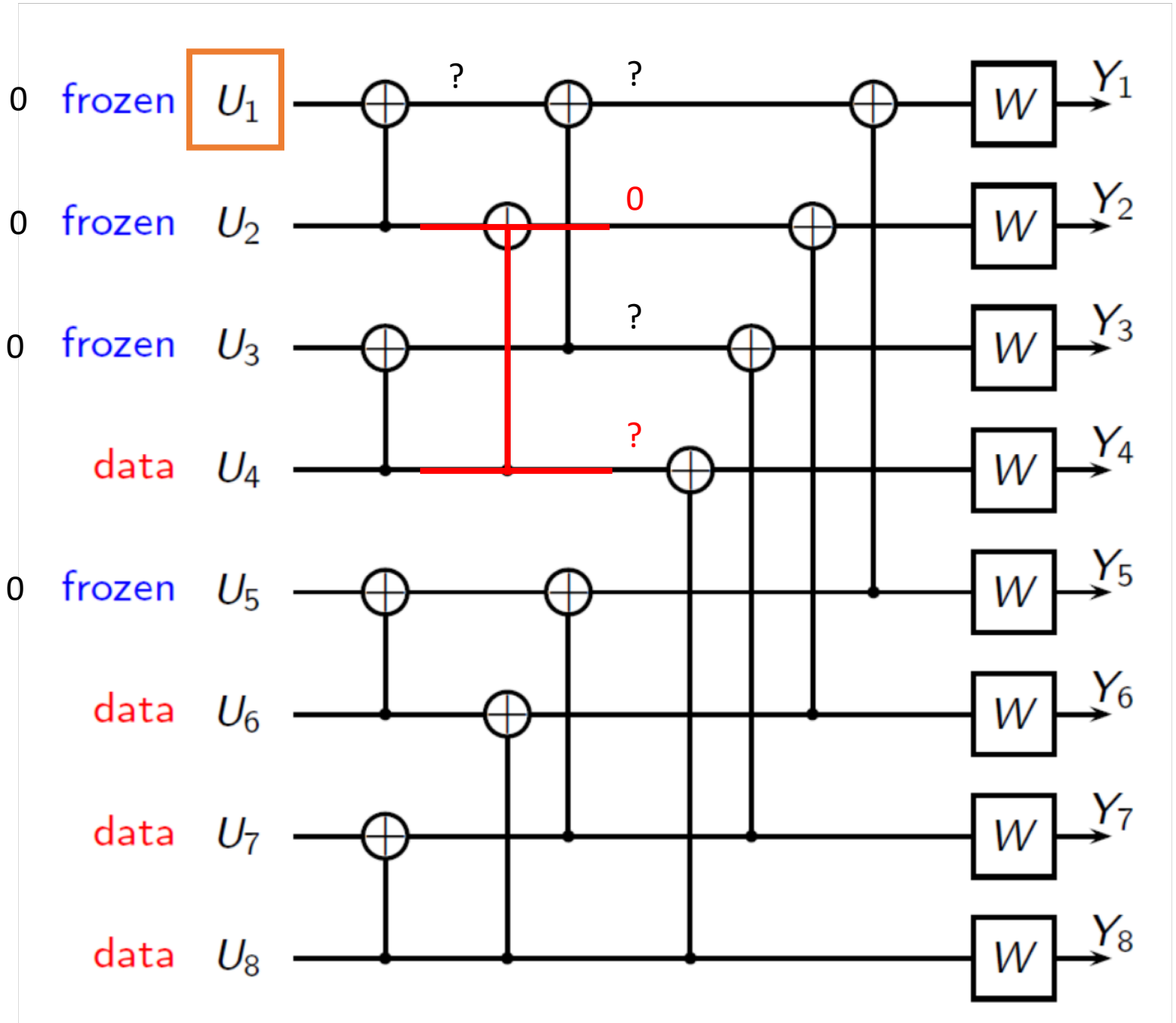

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
0
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

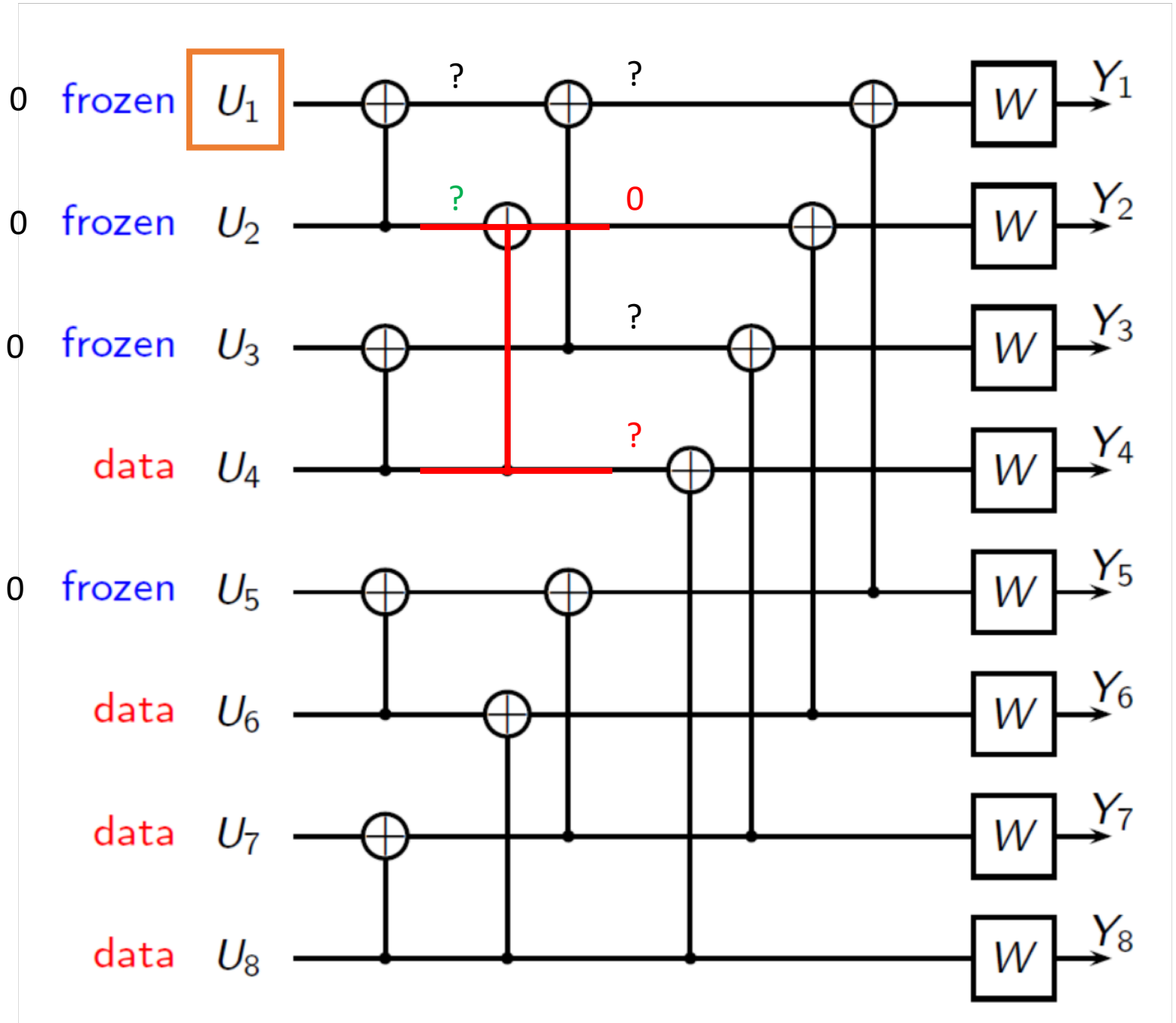

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
0
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

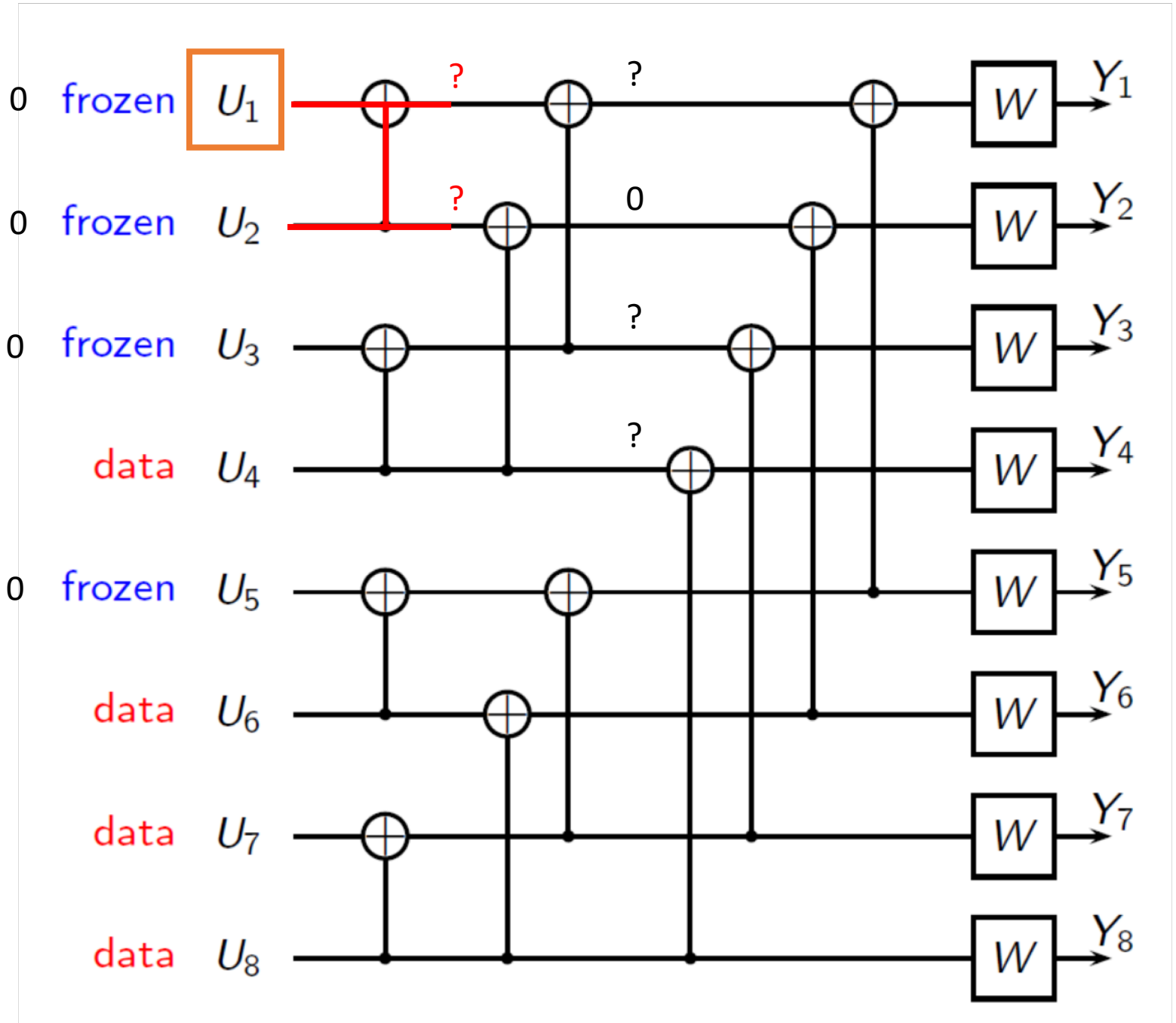

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

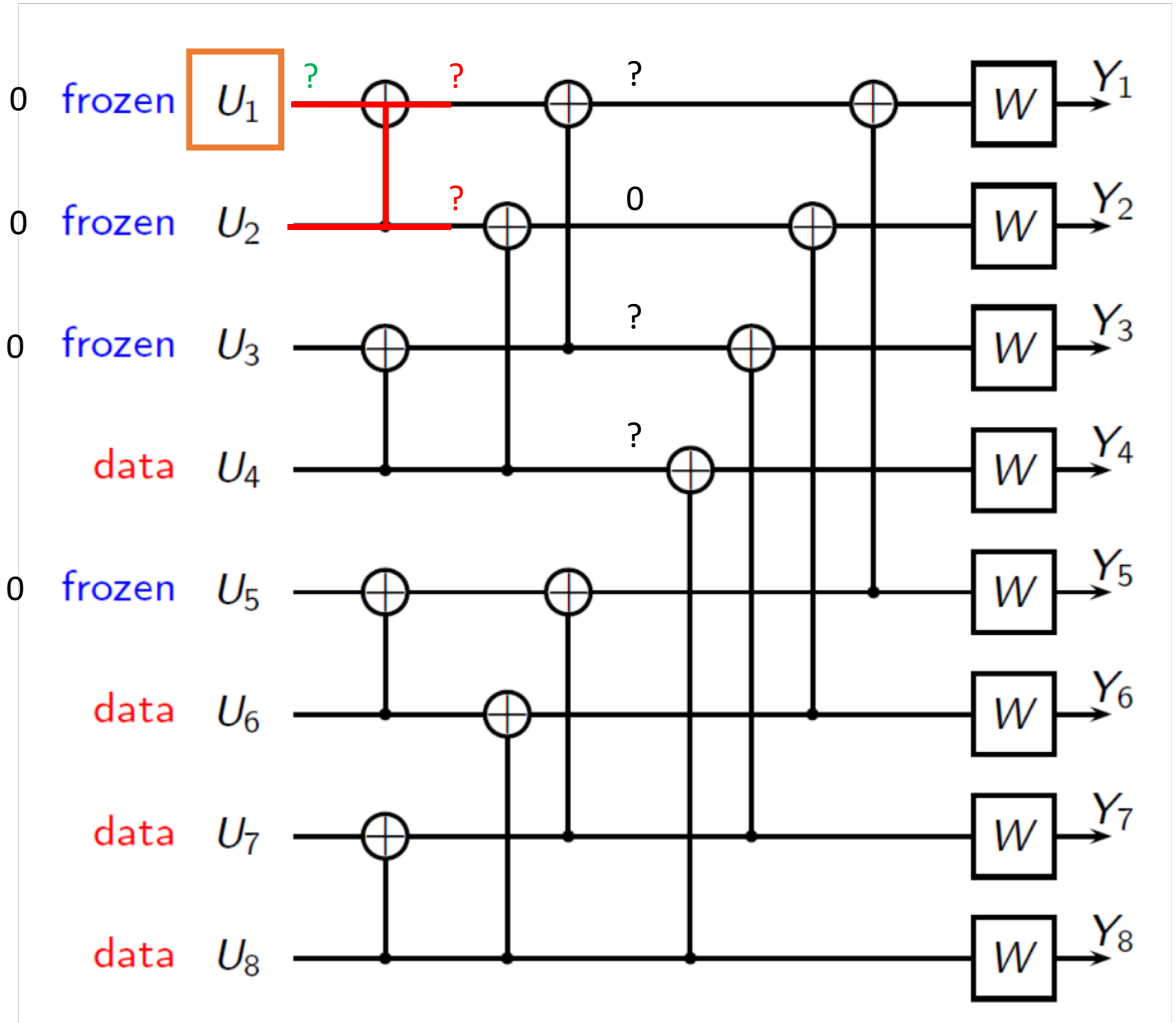

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

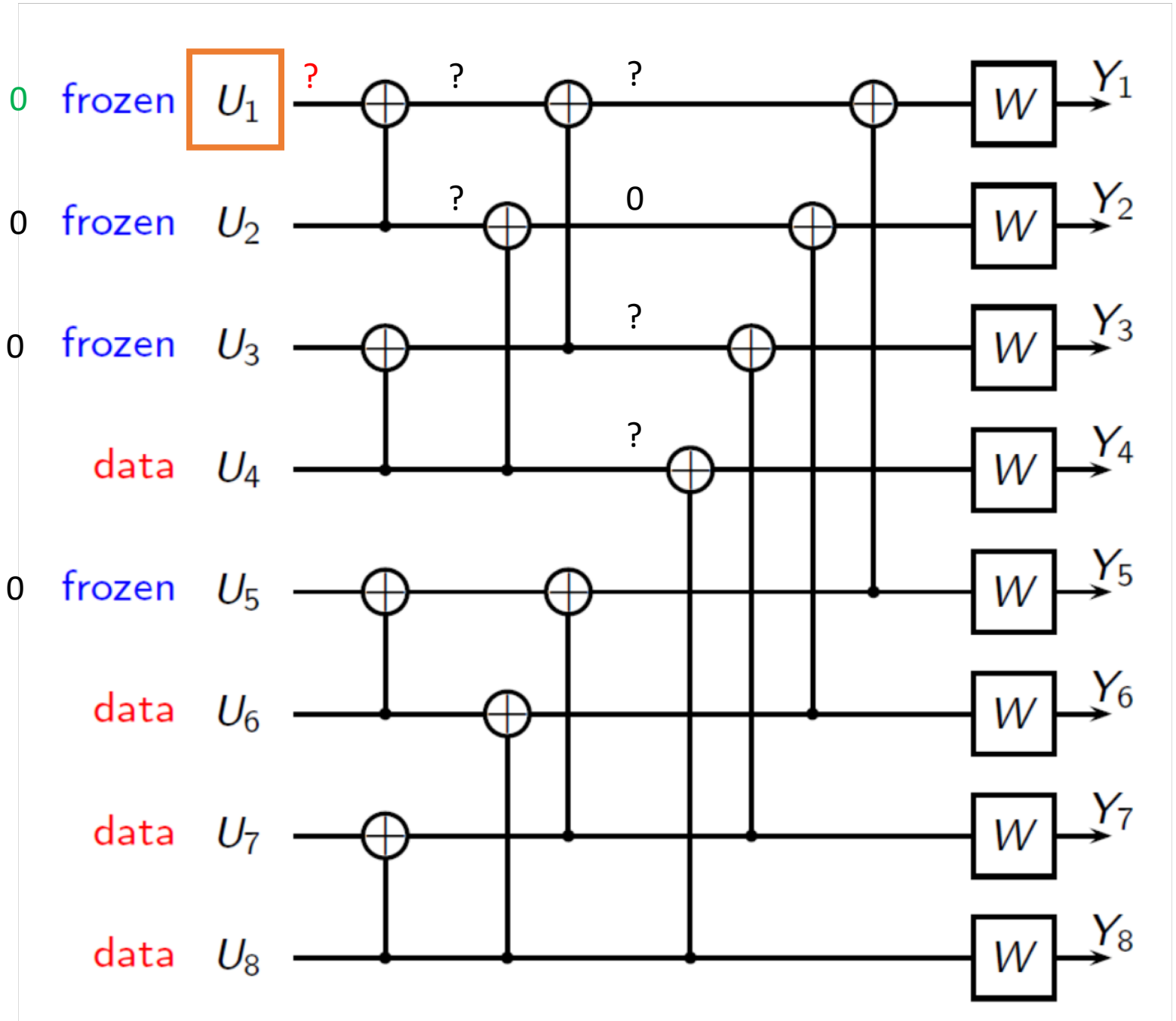

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

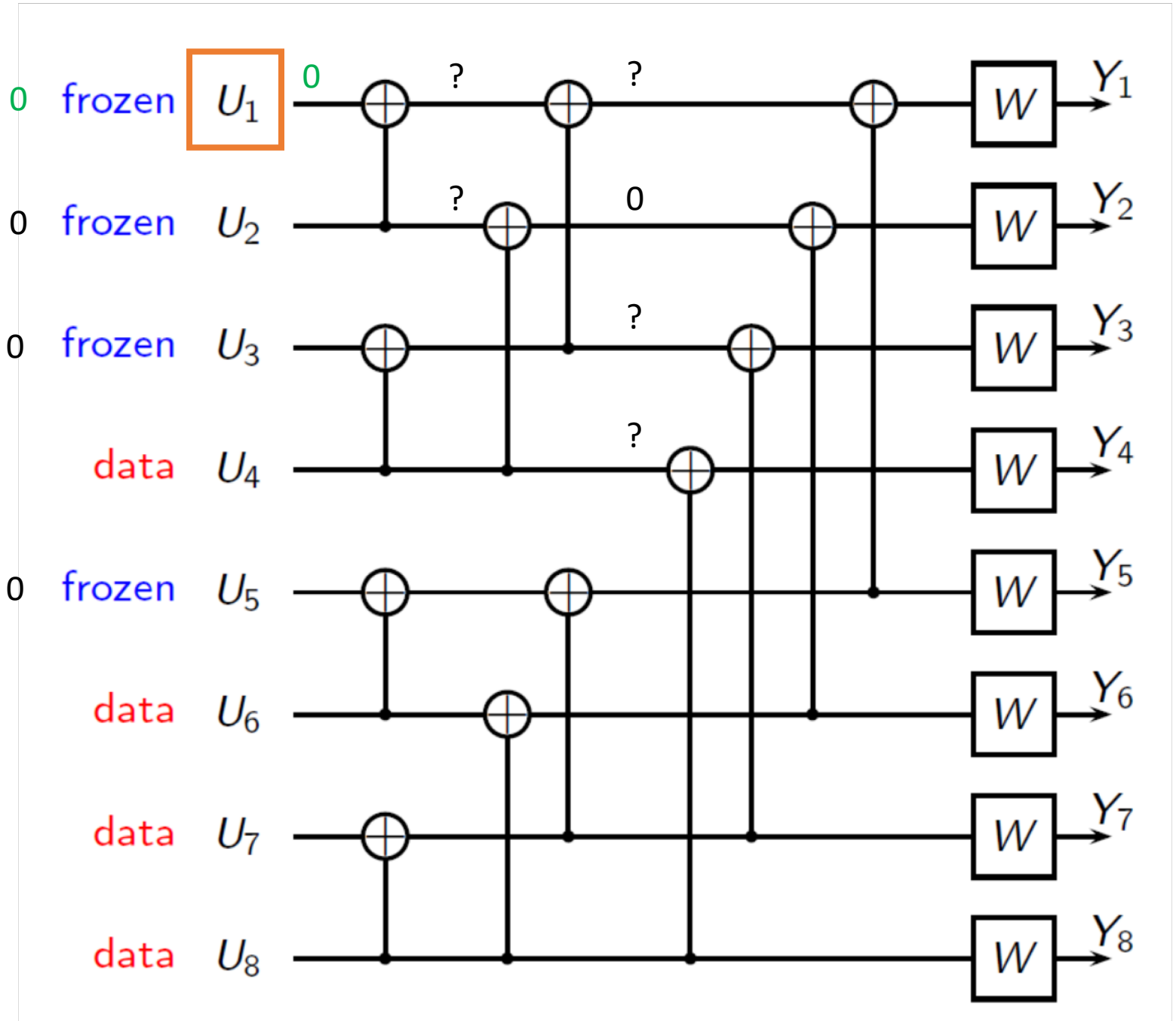

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

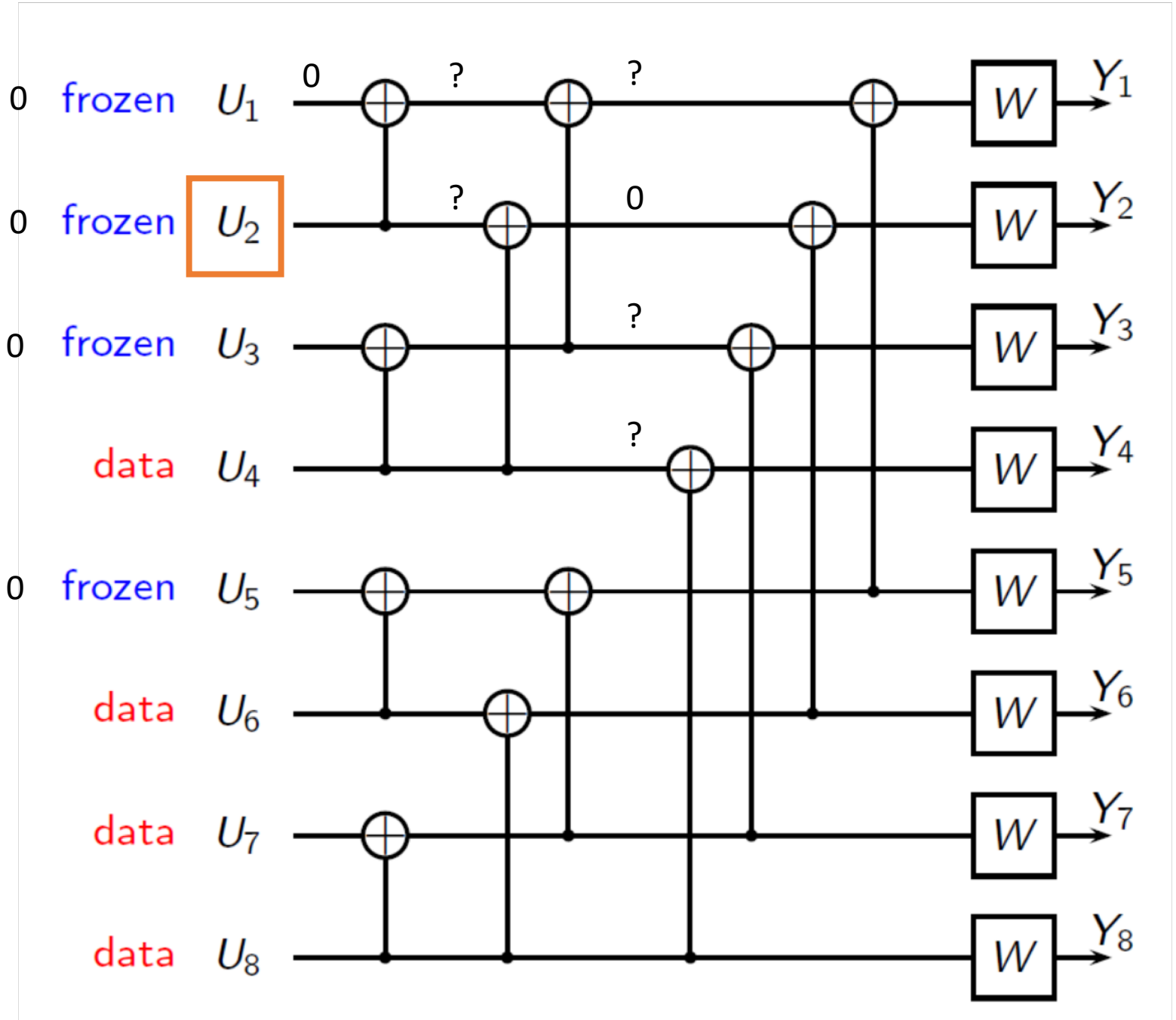

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
?
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

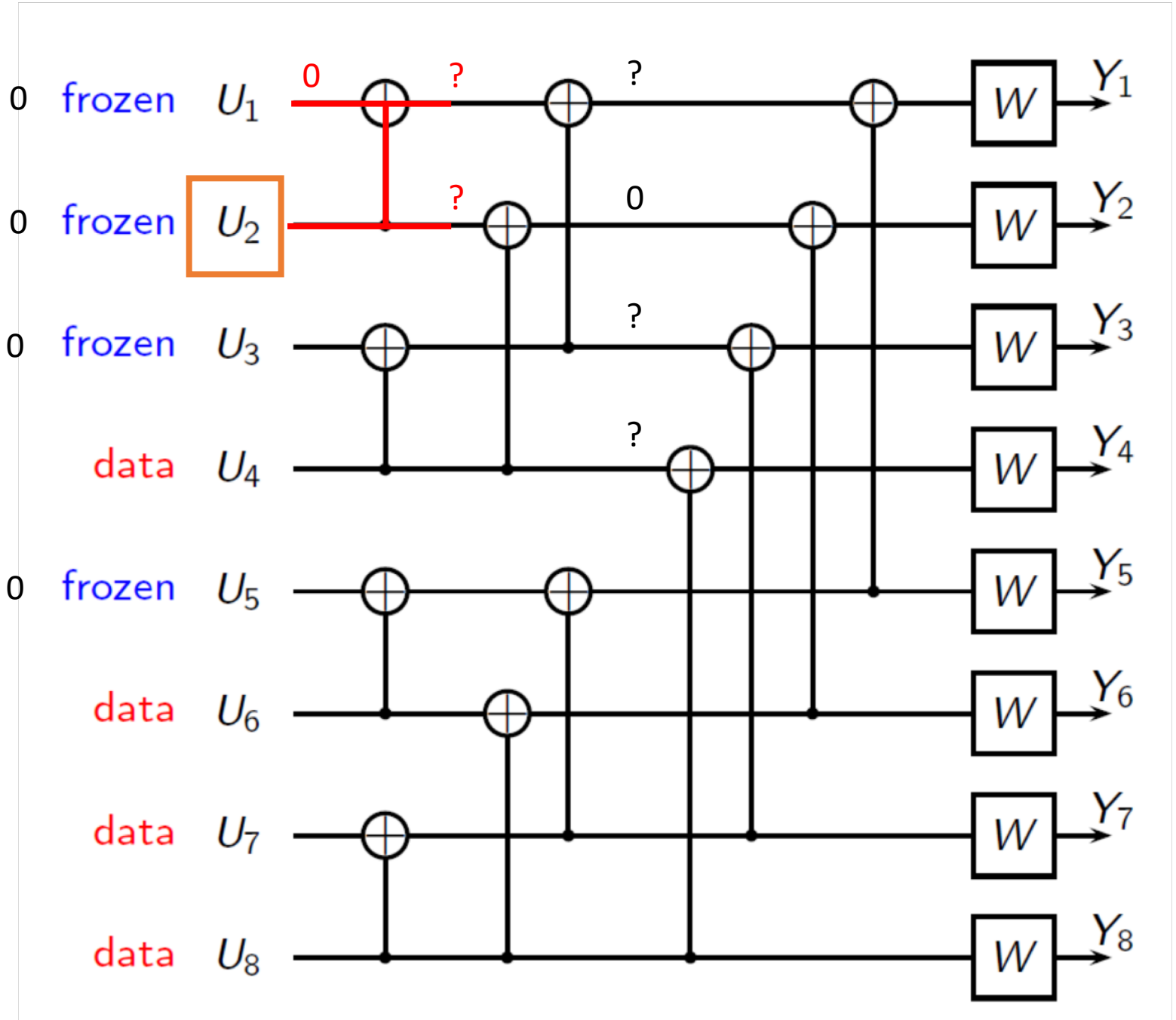

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
?
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

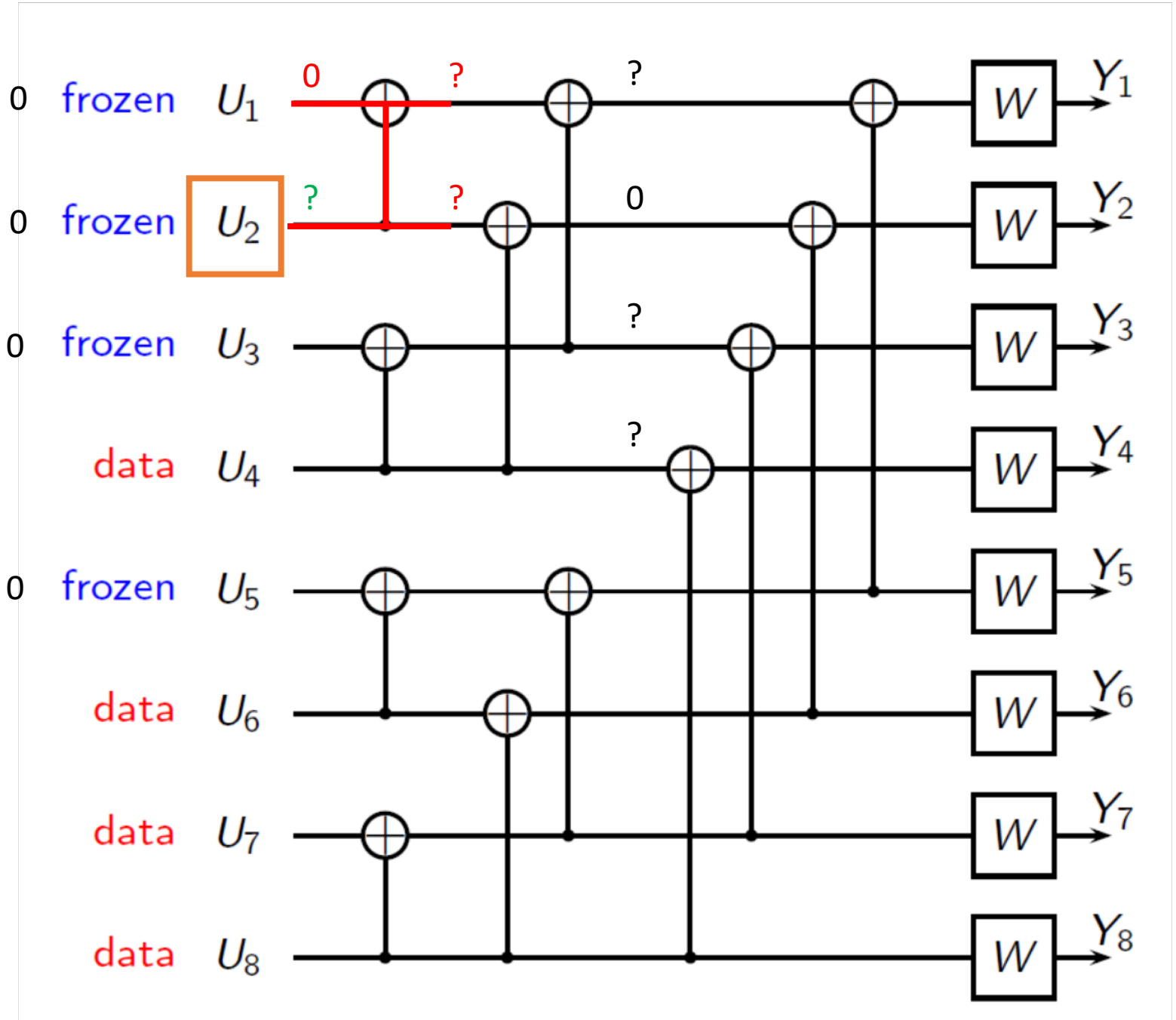

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
?
?
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

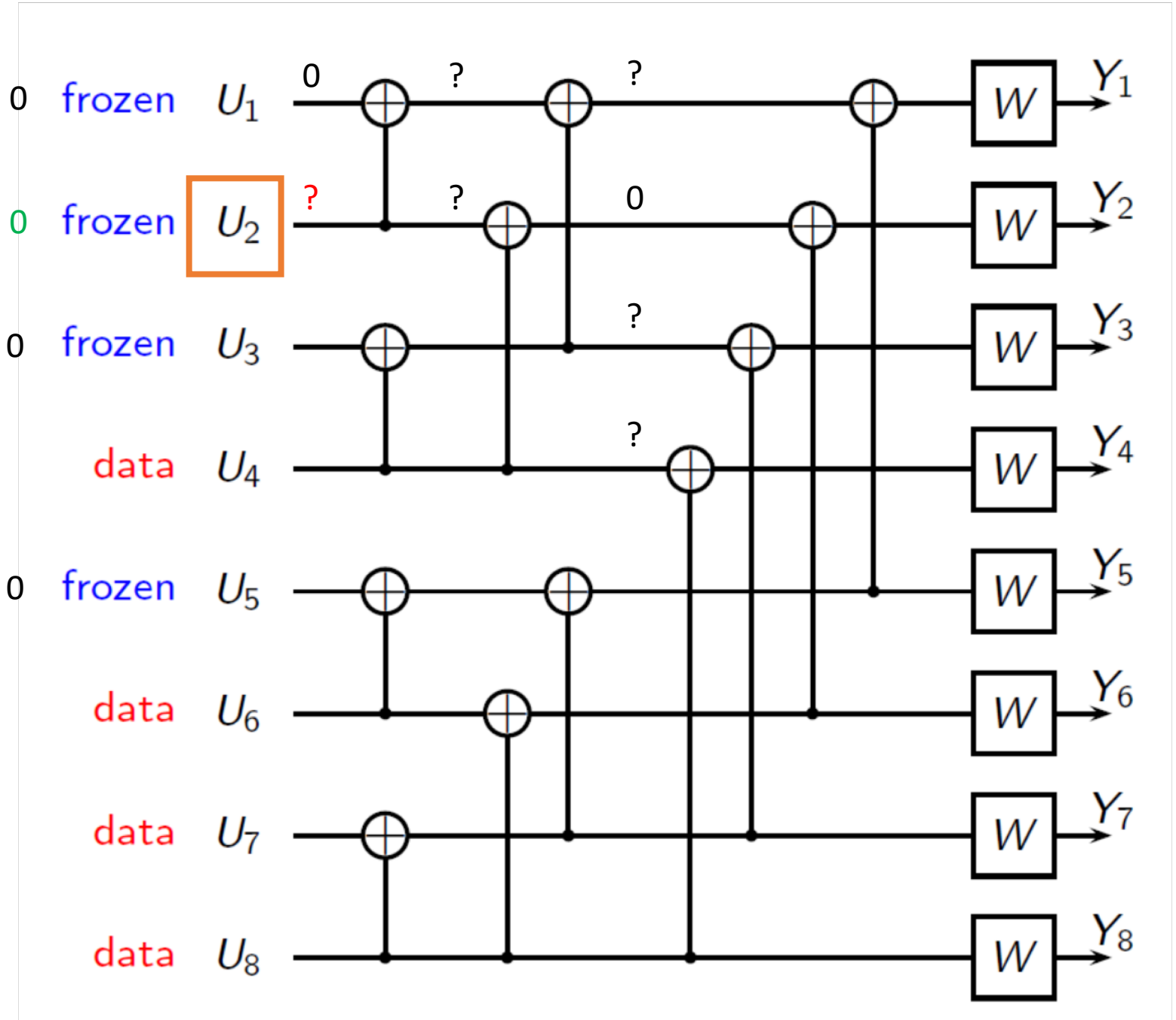

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
?
?
?
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

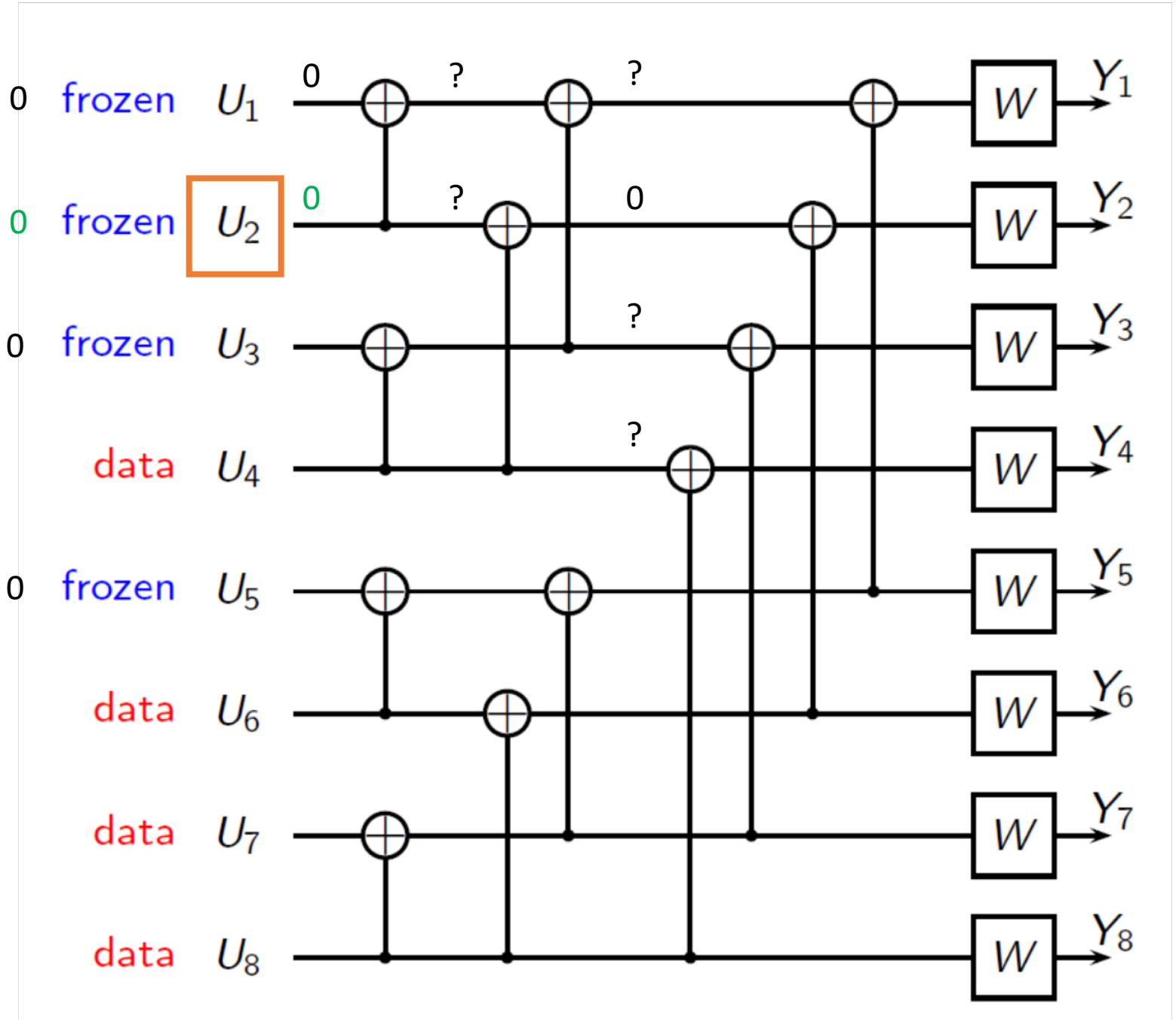

0 frozen U_1 0
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
?
?
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

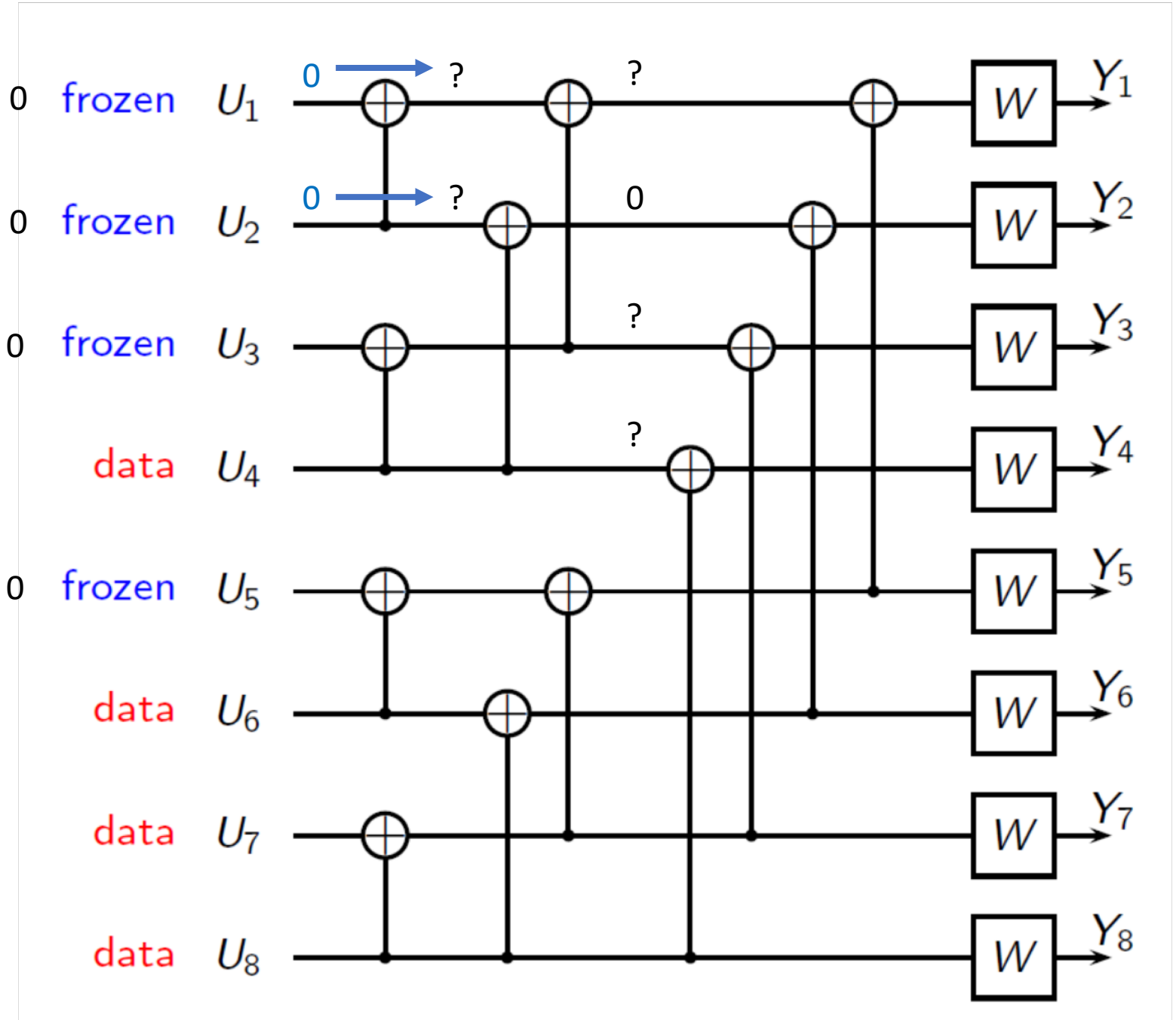

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0
?
?
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

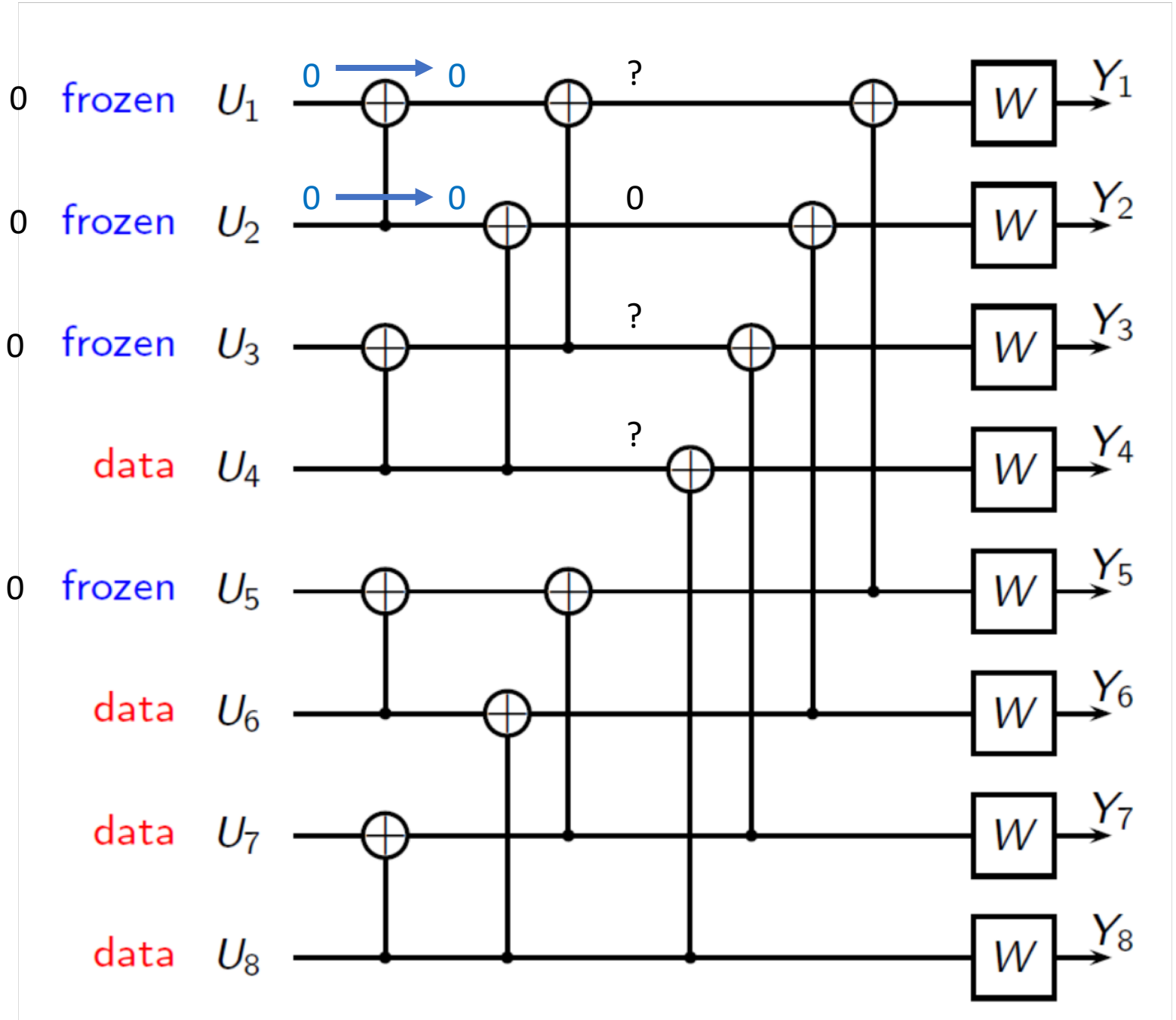

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0
?
?
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

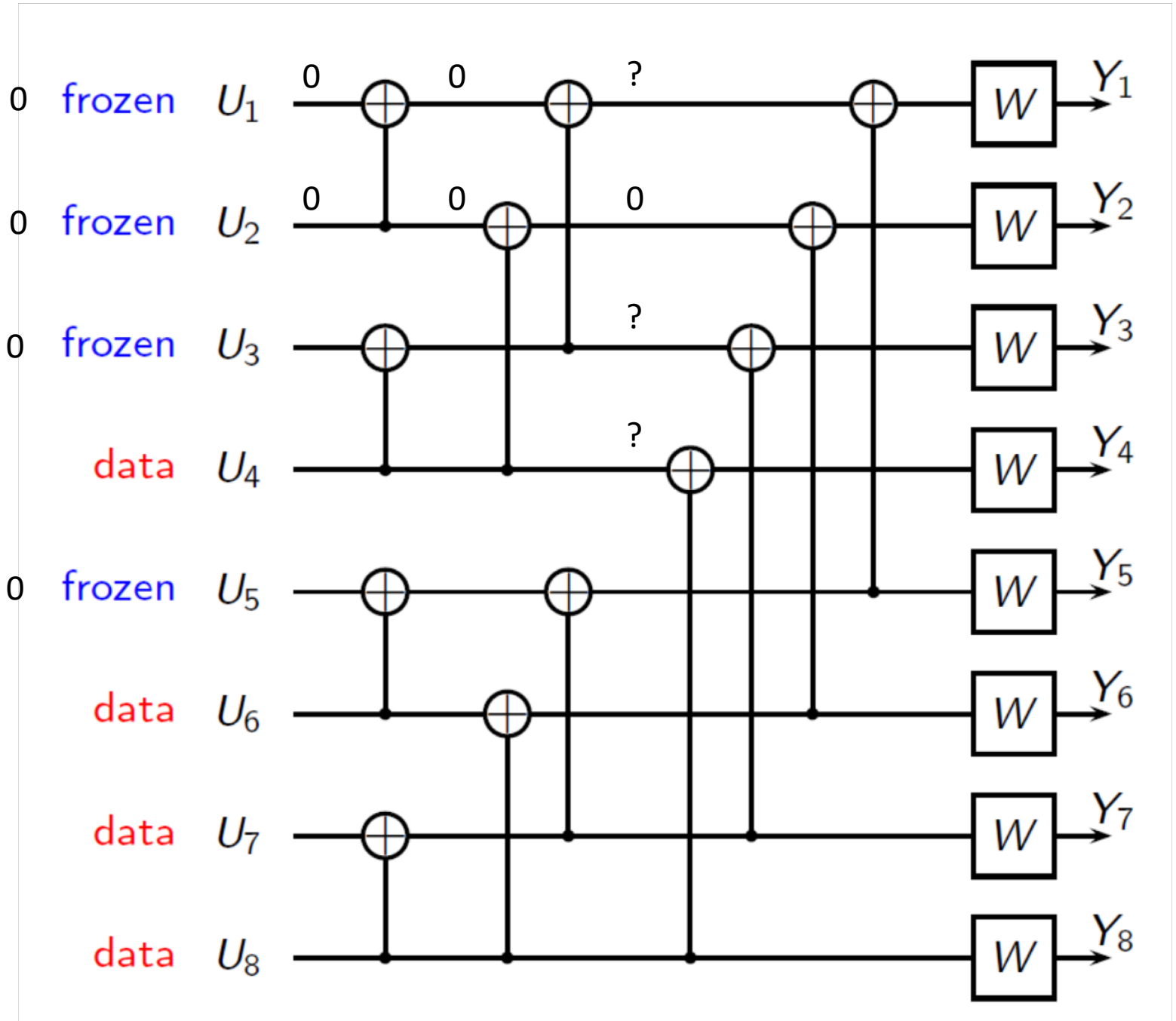

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0
?
?
?
?
?
0
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0


0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0
?
?
?
0
?
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

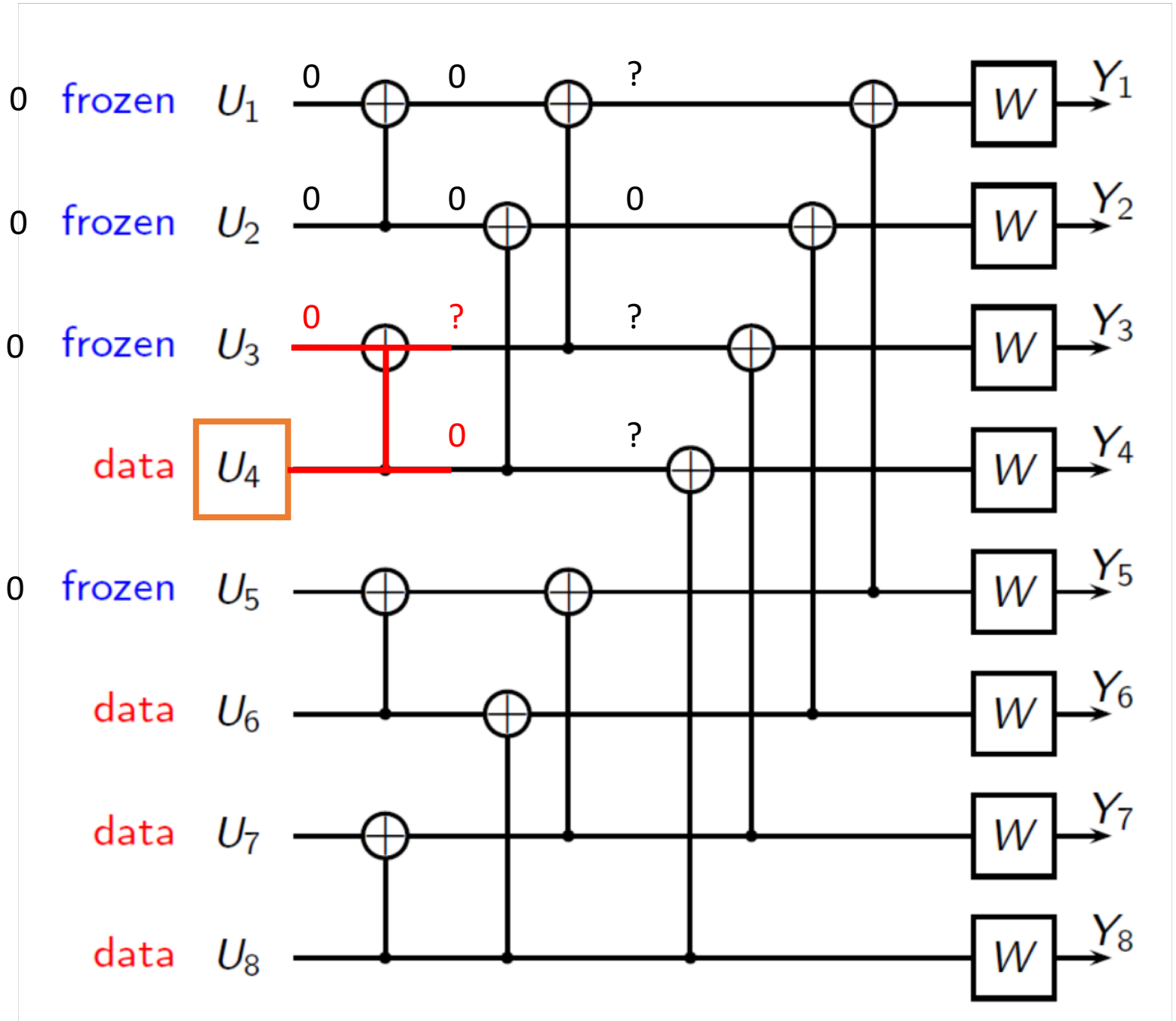

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0
0
?
?
?
0
?
?
0
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?

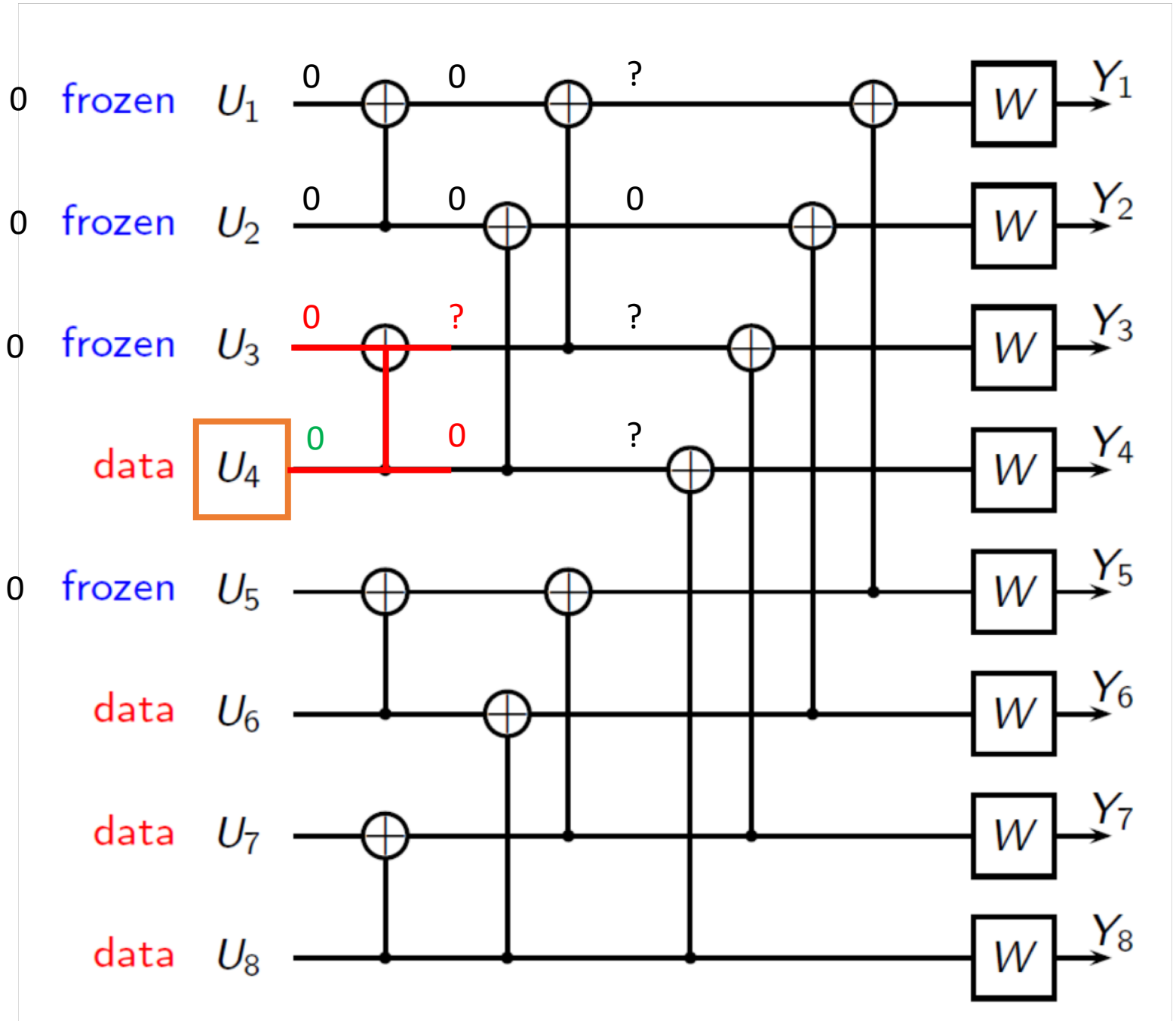

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0
?
?
0
?
0
?
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

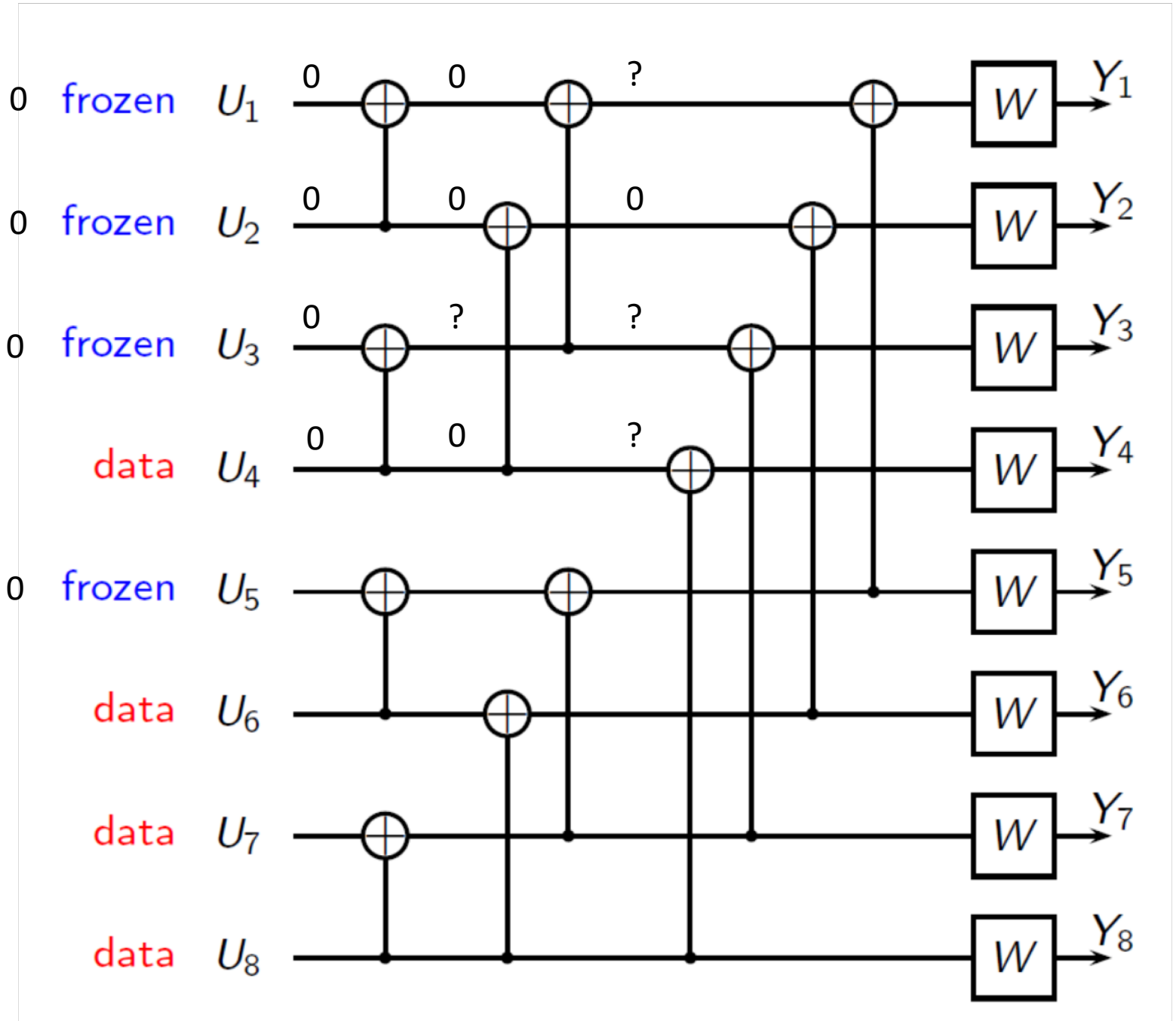

0
frozen
U_1
0
0
?
Y_1
?
0
frozen
U_2
0
0
0
Y_2
0
0
frozen
U_3
?
Y_3
0
data
U_4
?
Y_4
?
0
frozen
U_5
Y_5
0
data
U_6
Y_6
0
data
U_7
Y_7
?
data
U_8
Y_8
0
W

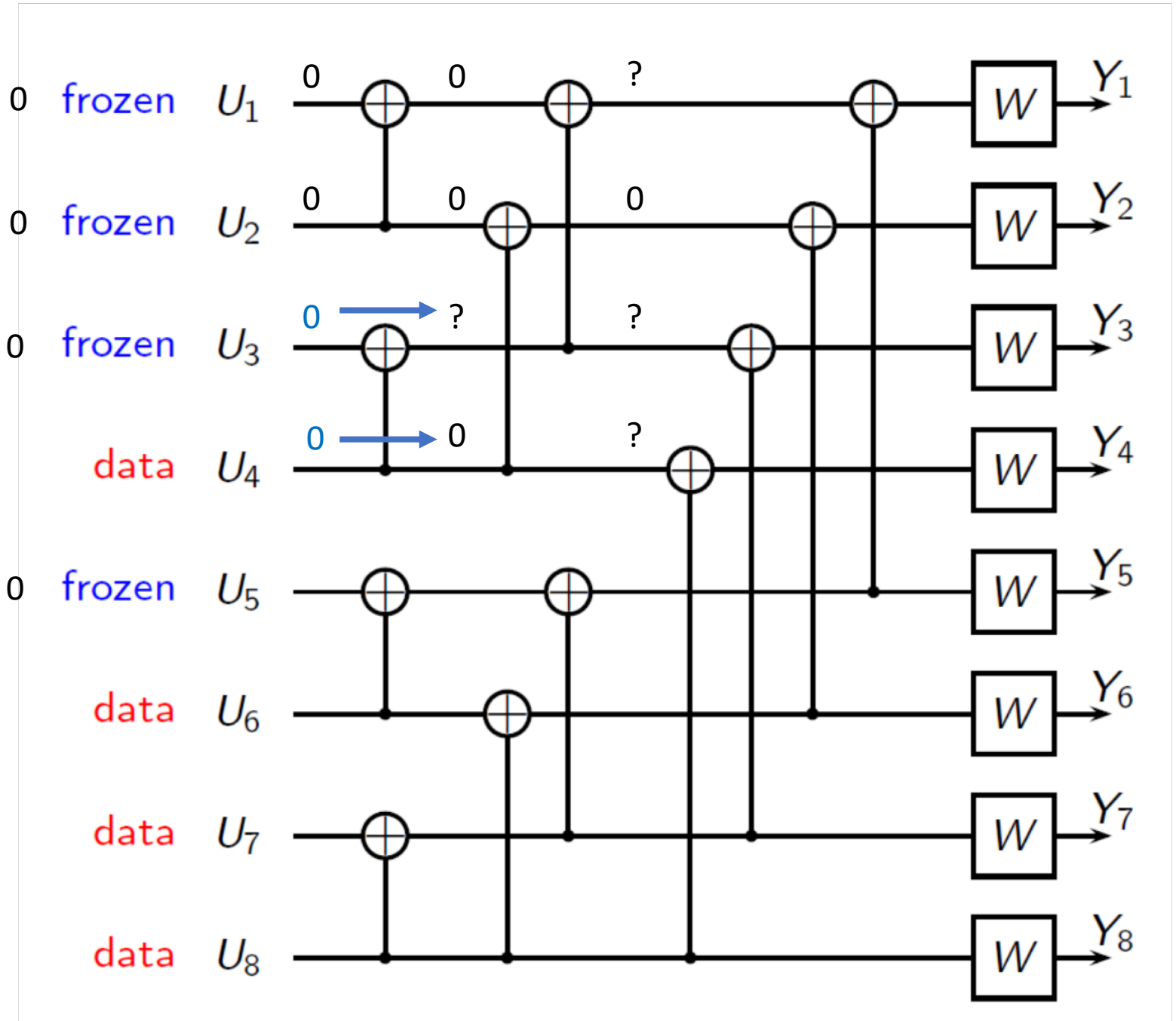

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0 0 ?
0 0 0
? ?
W
Y_1 ?
Y_2 0
Y_3 0
Y_4 ?
Y_5 0
Y_6 0
Y_7 ?
Y_8 0

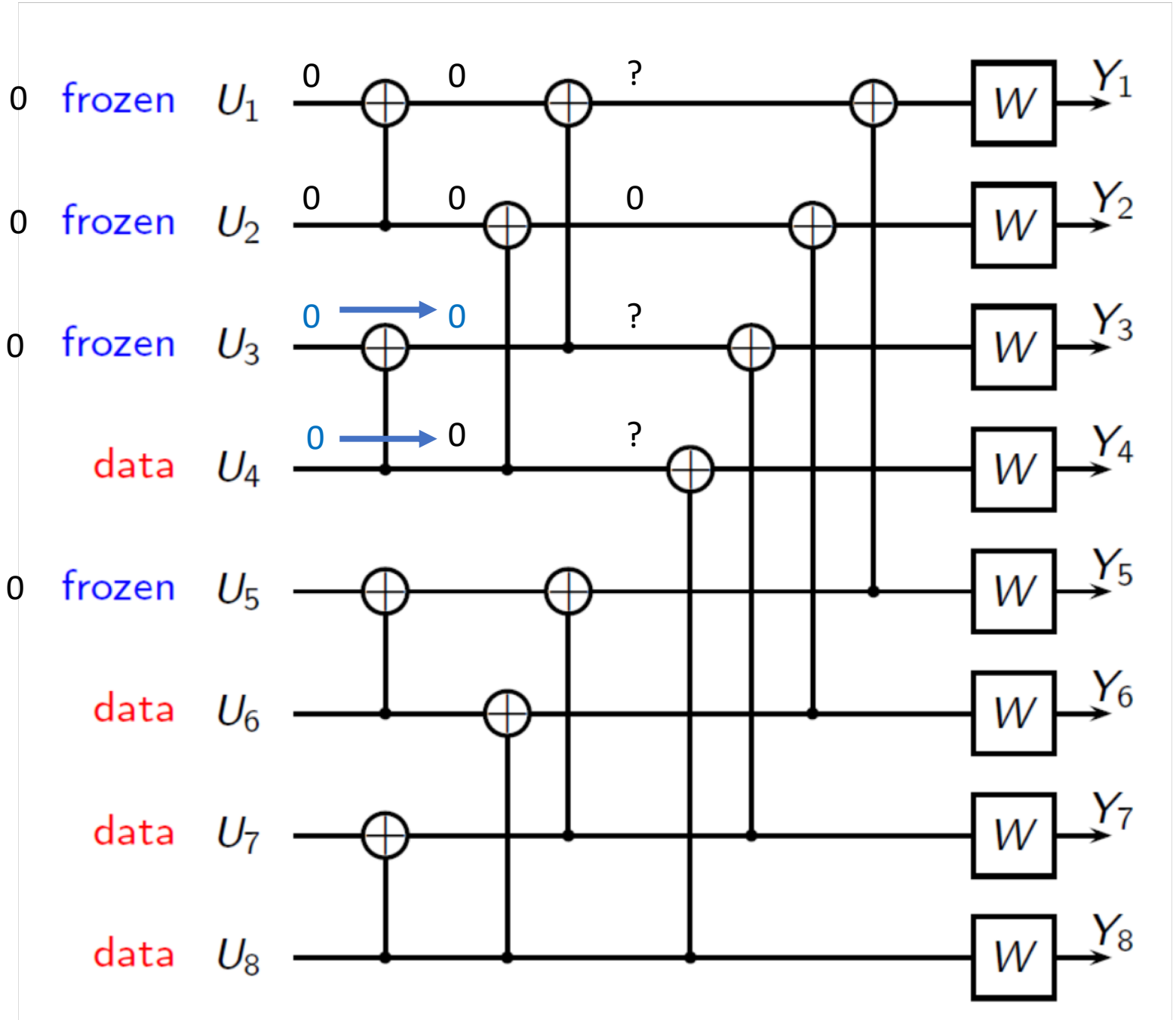
After a few steps …

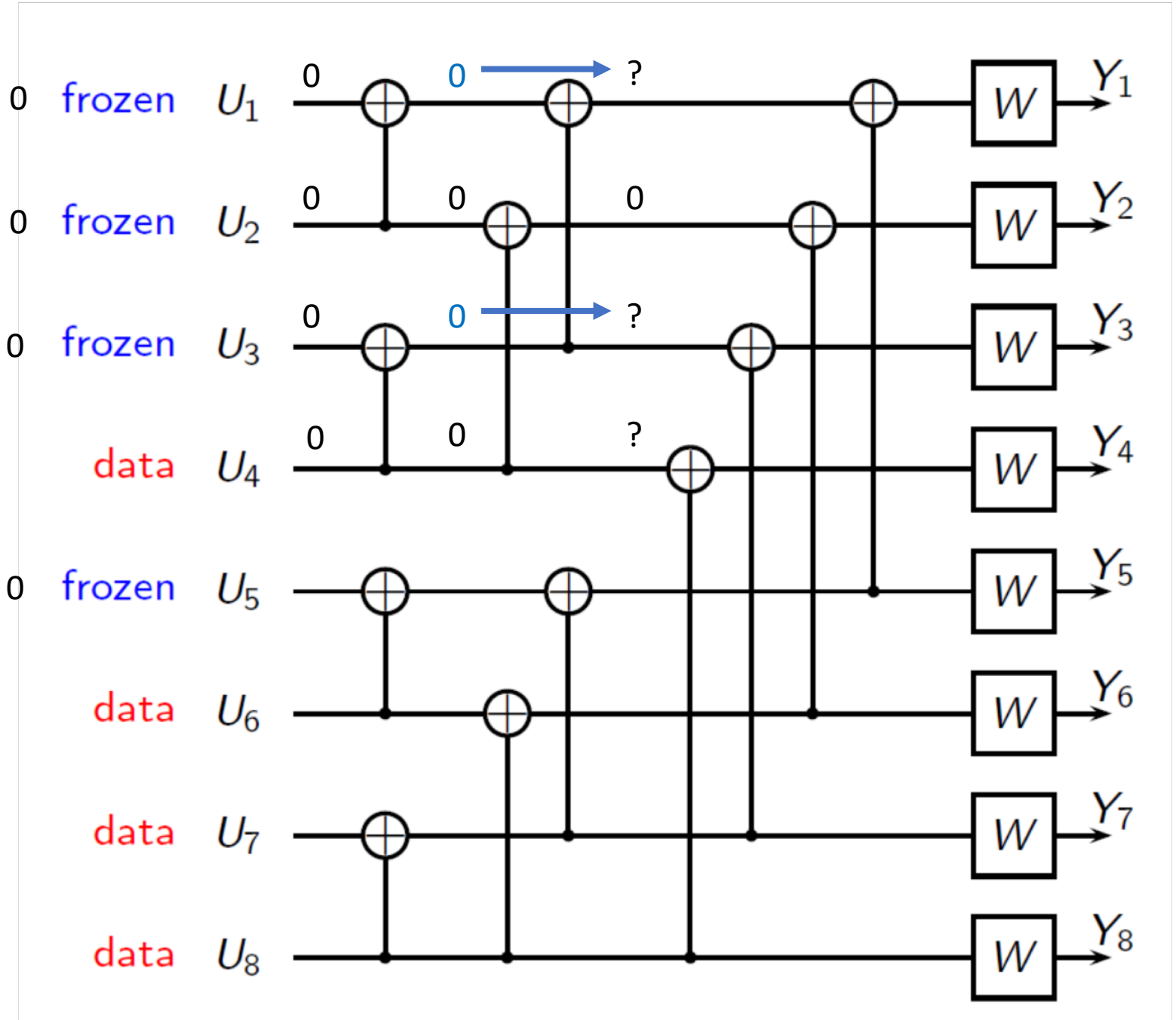

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
? 0 0 0 ? ?

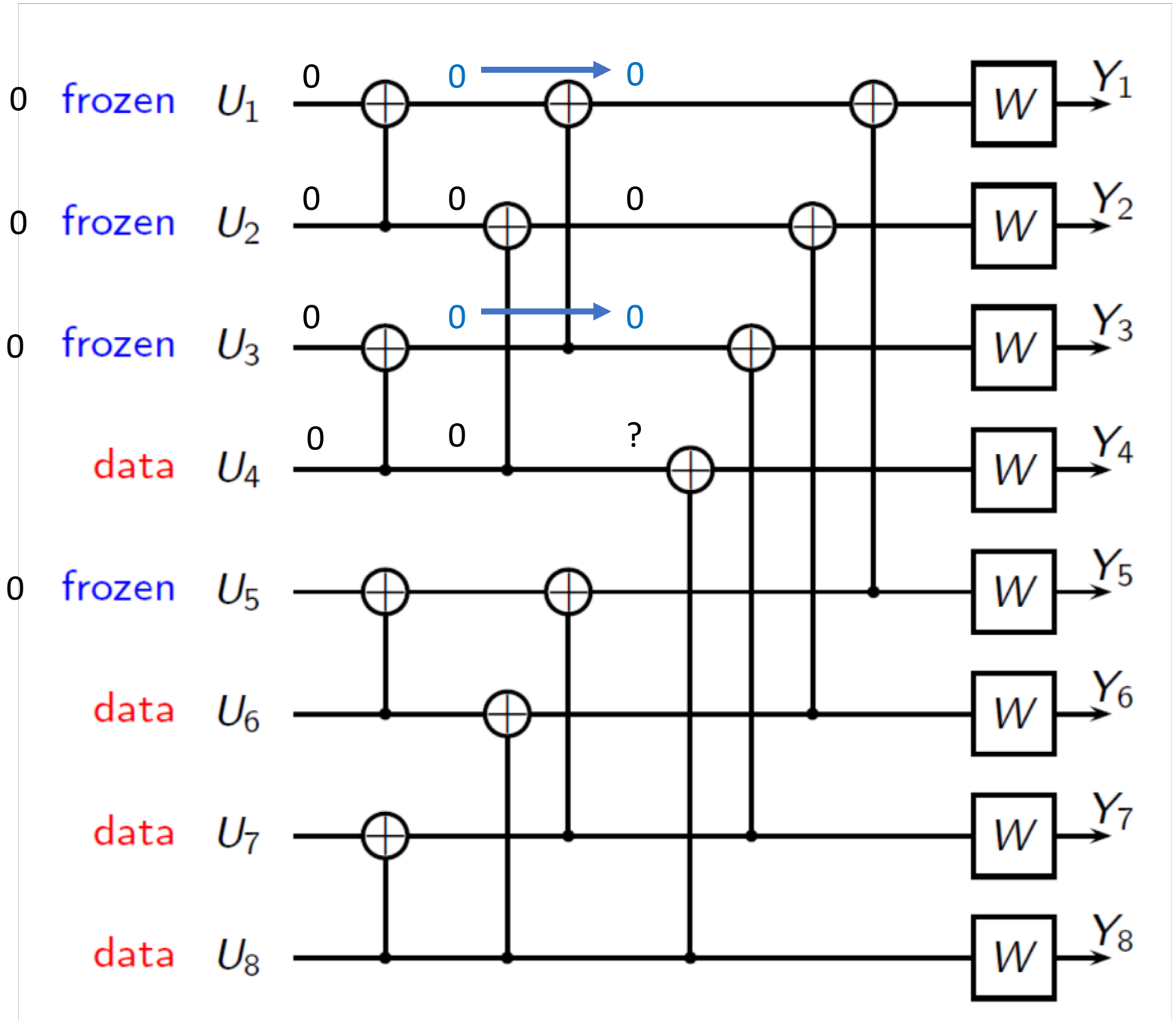

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
? 0 0 ? 0 0 ?

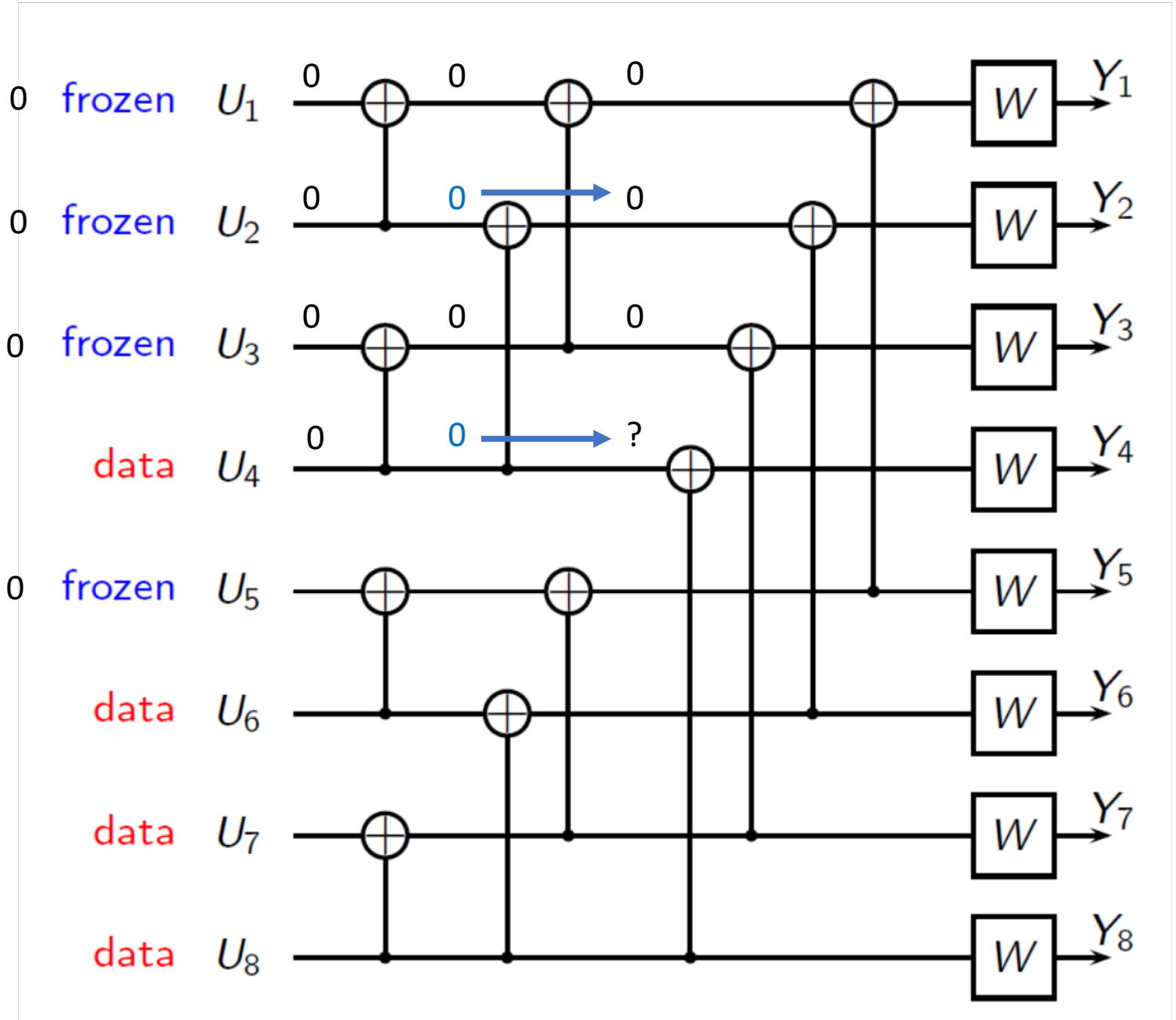

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0 0 ?
0 0 0
0 ? ?
0 0 ?
W
Y_1 ?
Y_2 0
Y_3 0
Y_4 ?
Y_5 0
Y_6 0
Y_7 ?
Y_8 0

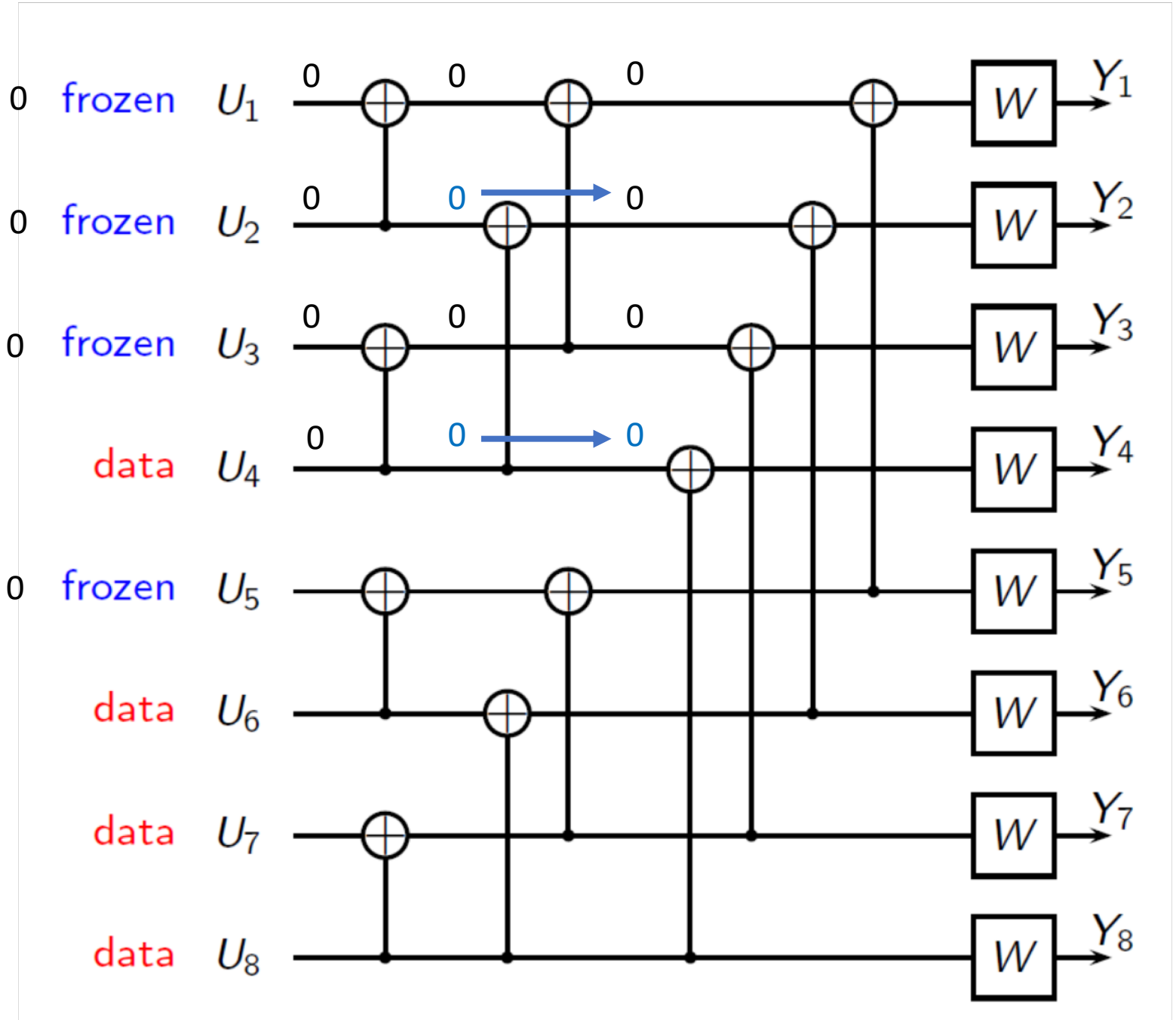

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0

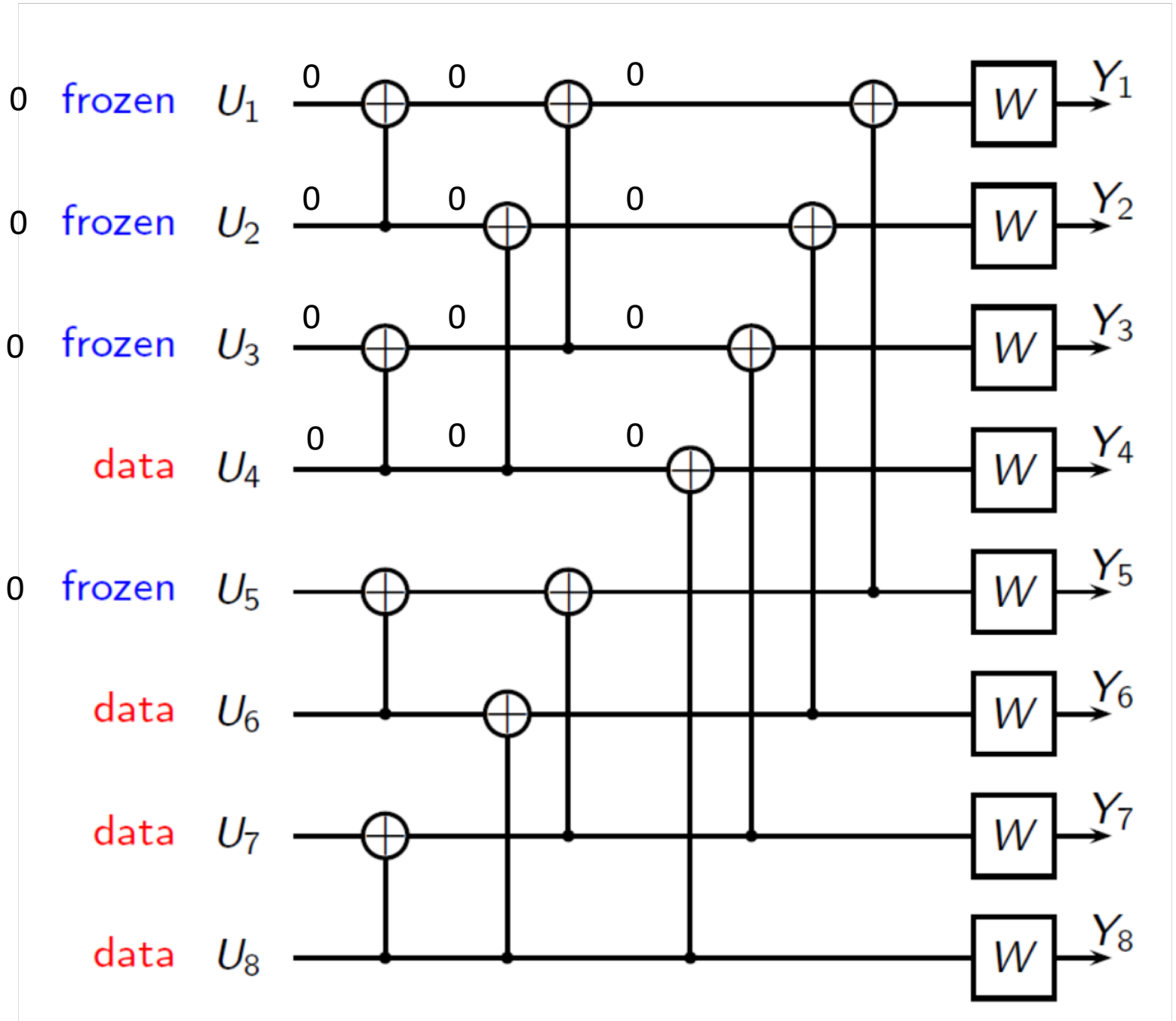

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0 0 ?
0 0 0
0 → 0 ?
0 → 0 ?
W
Y_1 ?
Y_2 0
Y_3 0
Y_4 ?
Y_5 0
Y_6 0
Y_7 ?
Y_8 0


0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0
0
?
0
0
0
0
0
?
0
0
?
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?

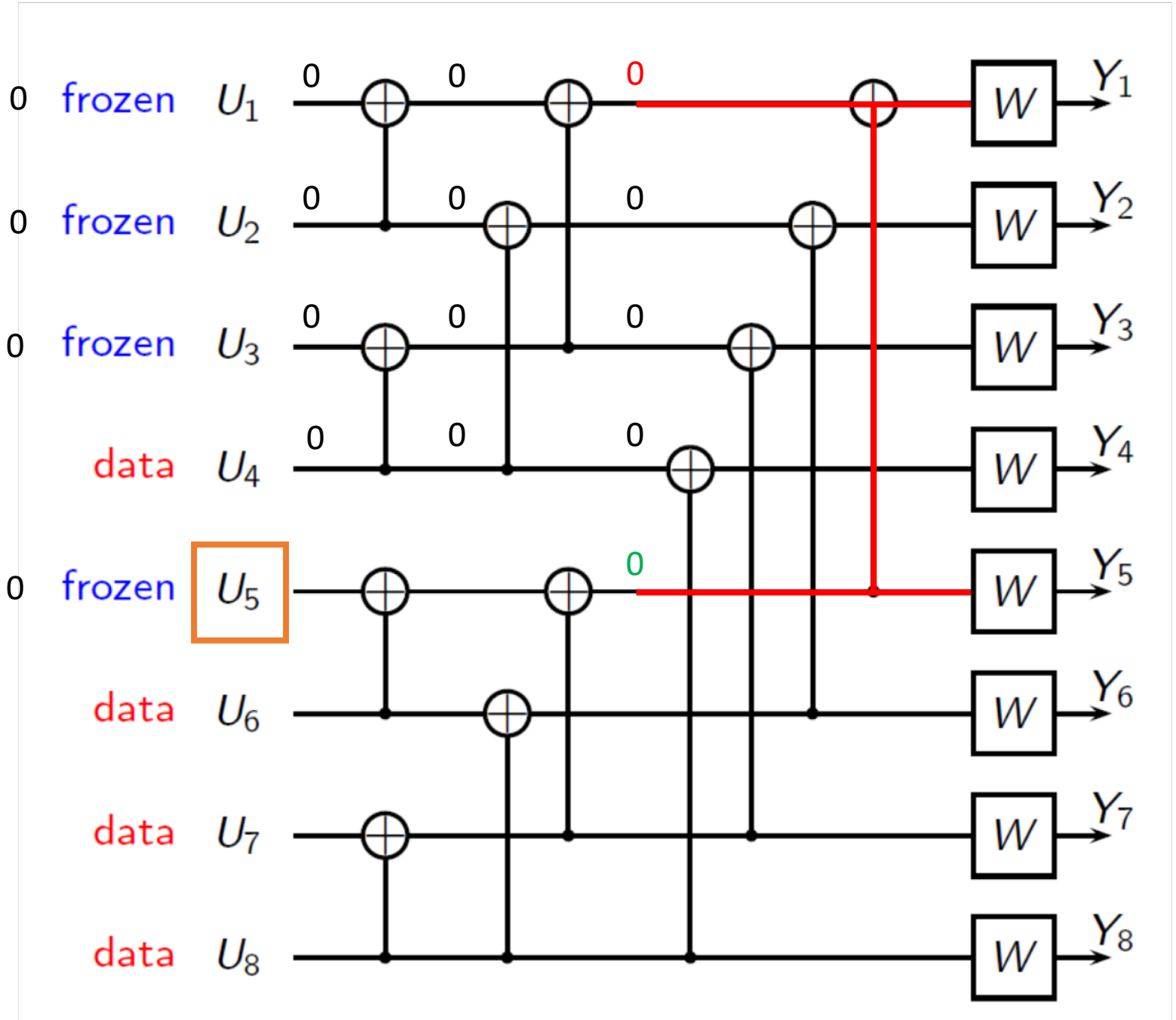

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0 0 0 0 0 0 0 0 ? 0 0 0 0 0 0 0 0
W
Y_1 ?
Y_2 0
Y_3 0
Y_4 ?
Y_5 0
Y_6 0
Y_7 ?
Y_8 0

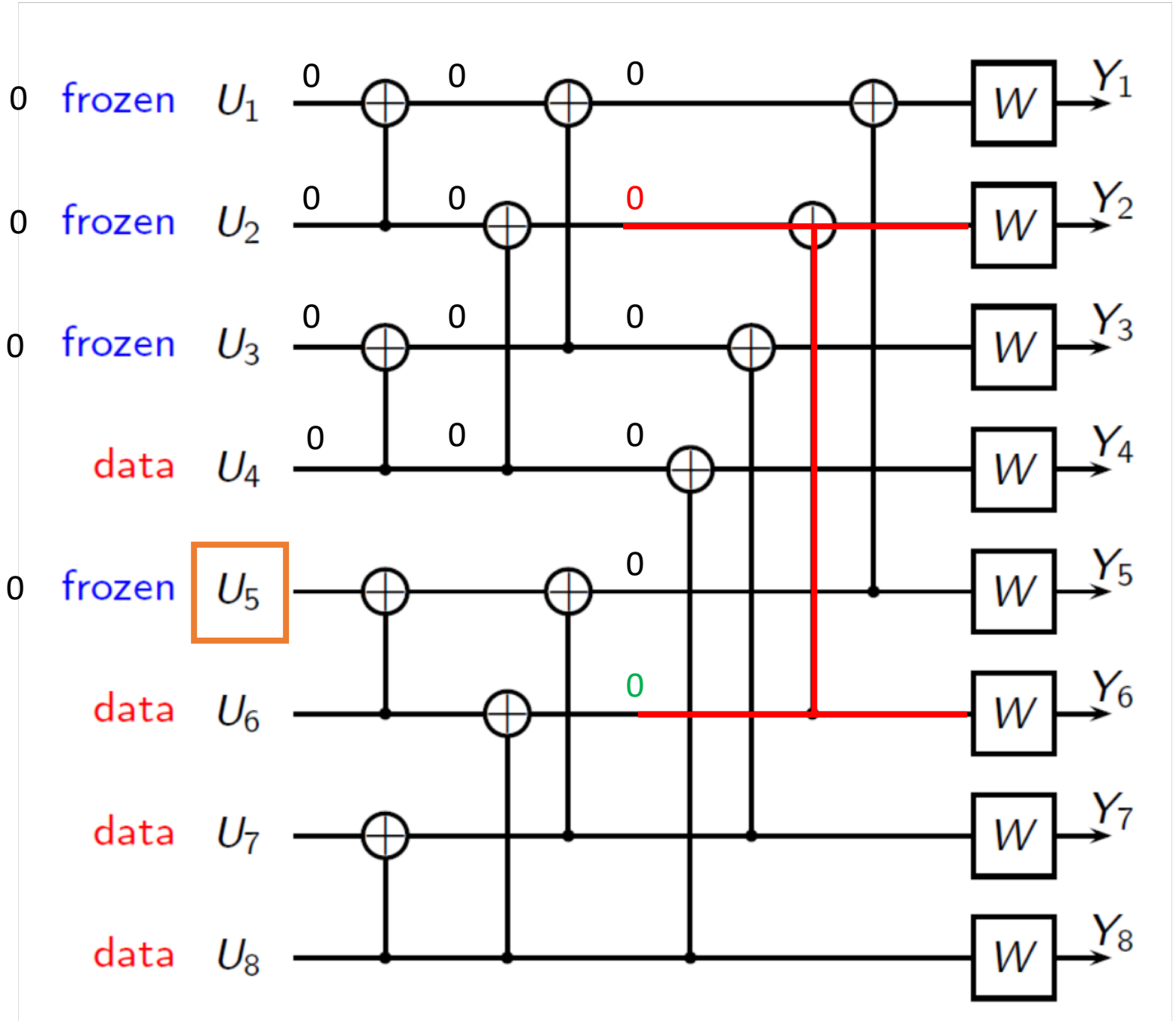

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0 0 0
0 0 0
0 0 0
0 0 ?
W
Y_1 ?
Y_2 0
Y_3 0
Y_4 ?
Y_5 0
Y_6 0
Y_7 ?
Y_8 0

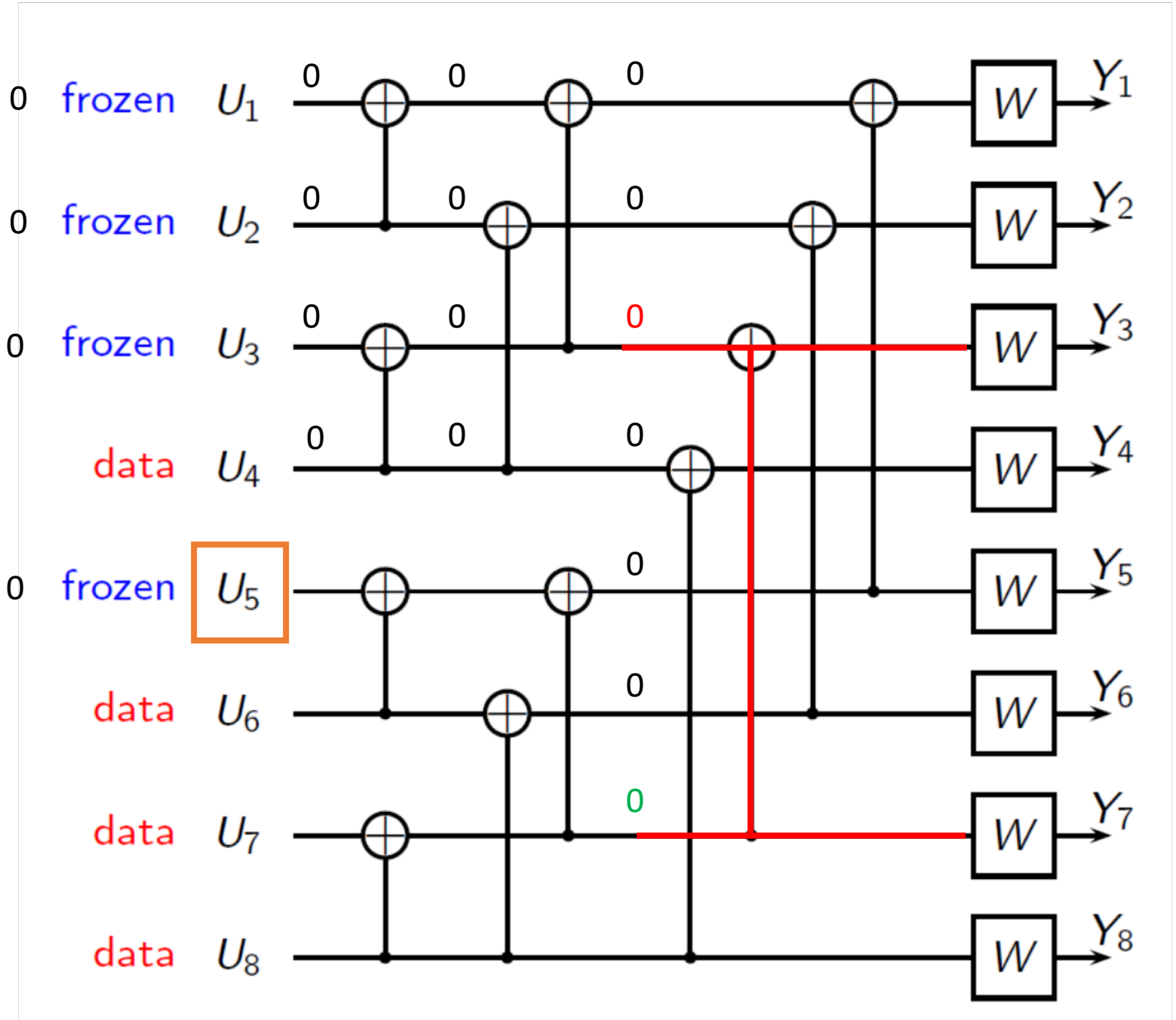

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
?
0
0
?
0
0

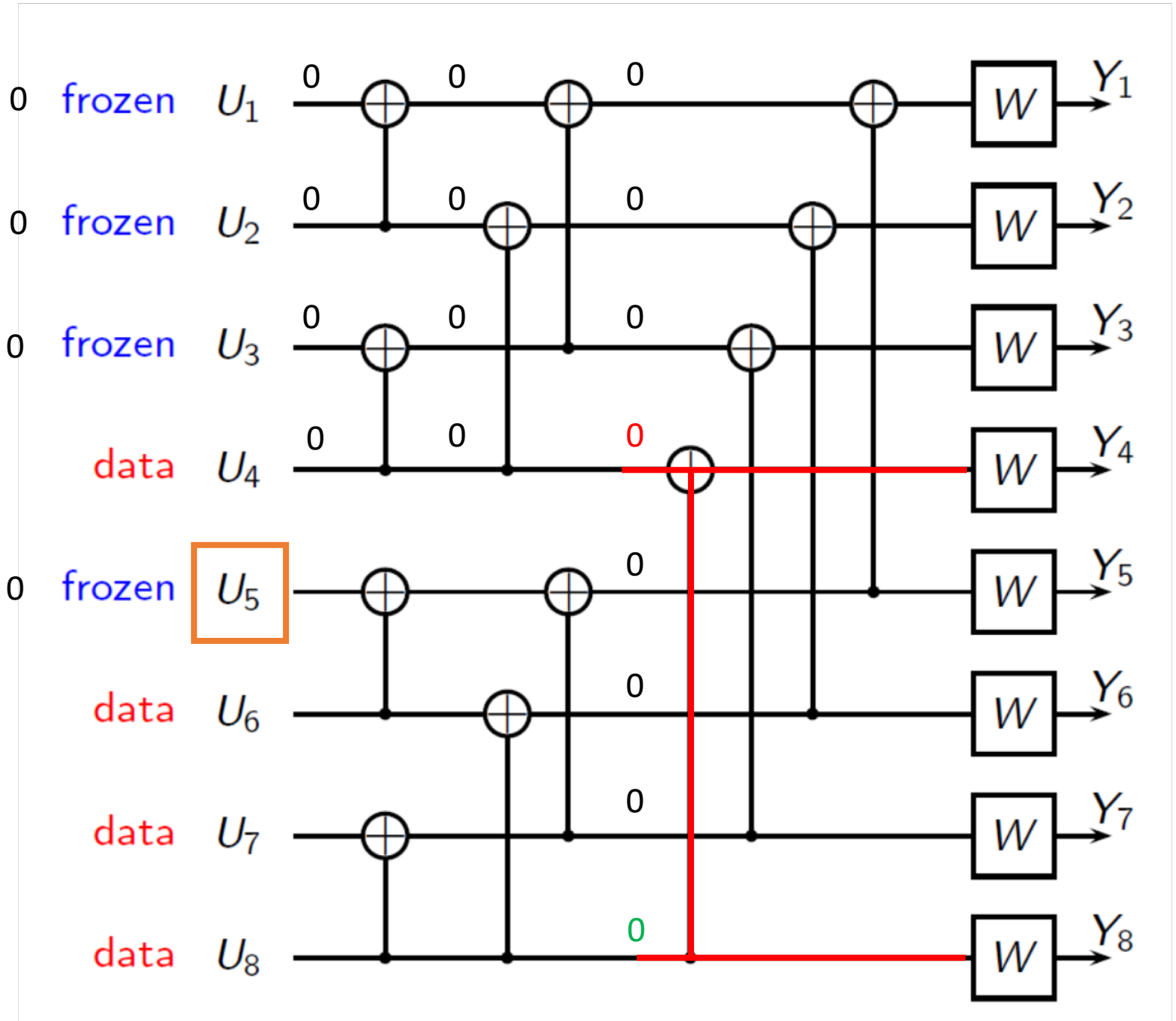

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
0 0 0
0 0 0
0 0 0
0 0 0
W Y_1 ?
W Y_2 0
W Y_3 0
W Y_4 ?
W Y_5 0
W Y_6 0
W Y_7 ?
W Y_8 0

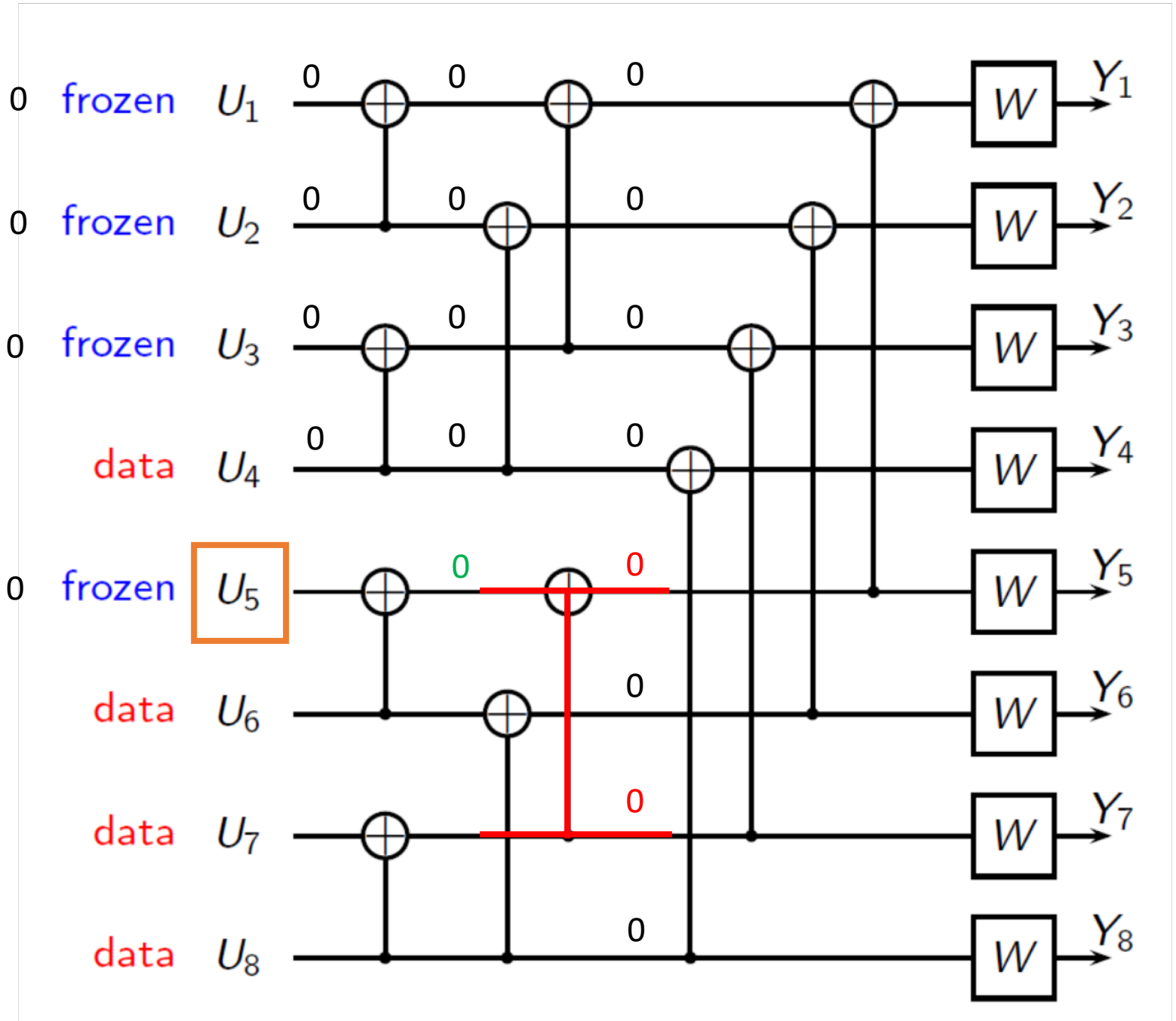
Now you get it (hopefully), so we'll be quicker

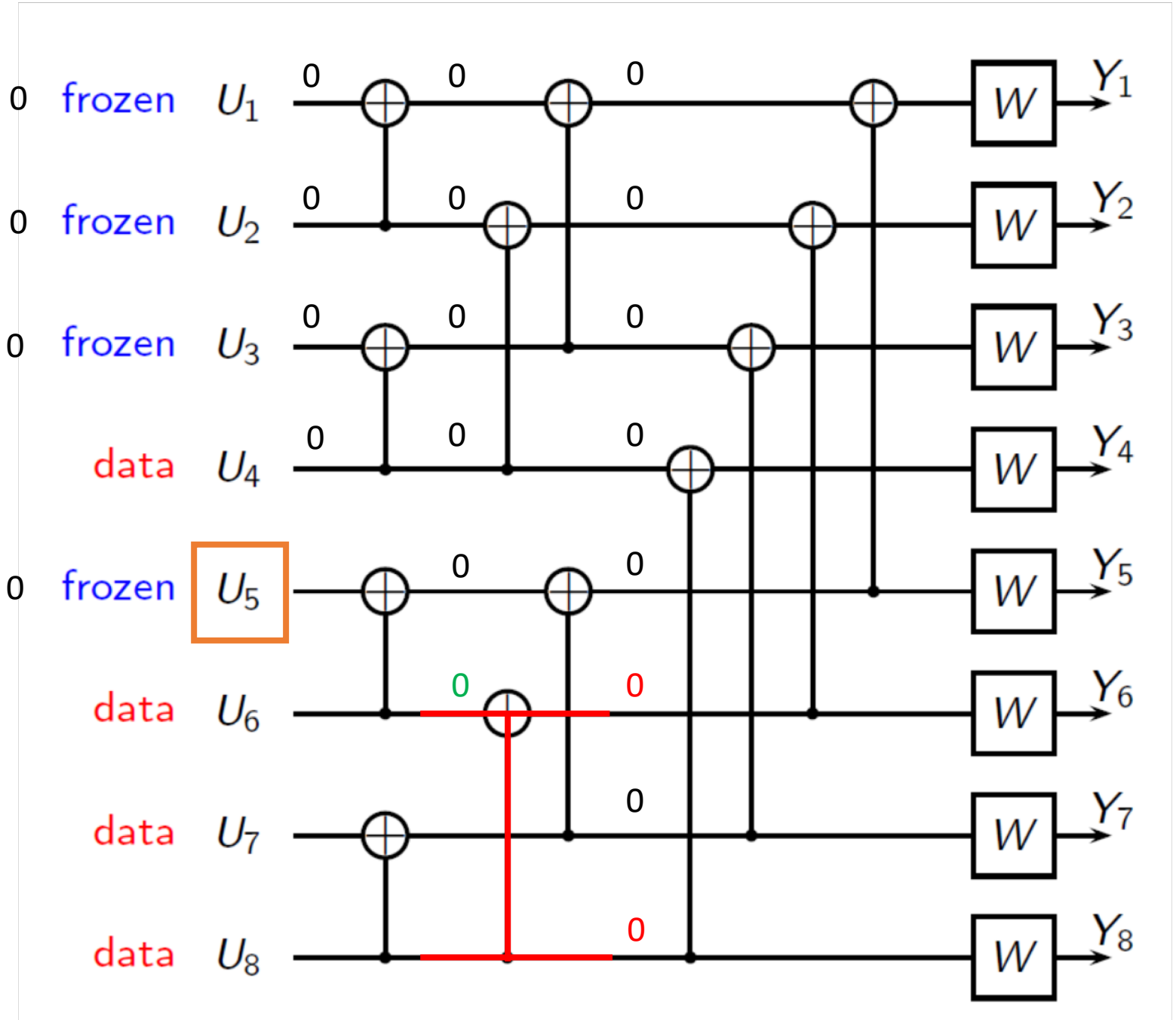

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

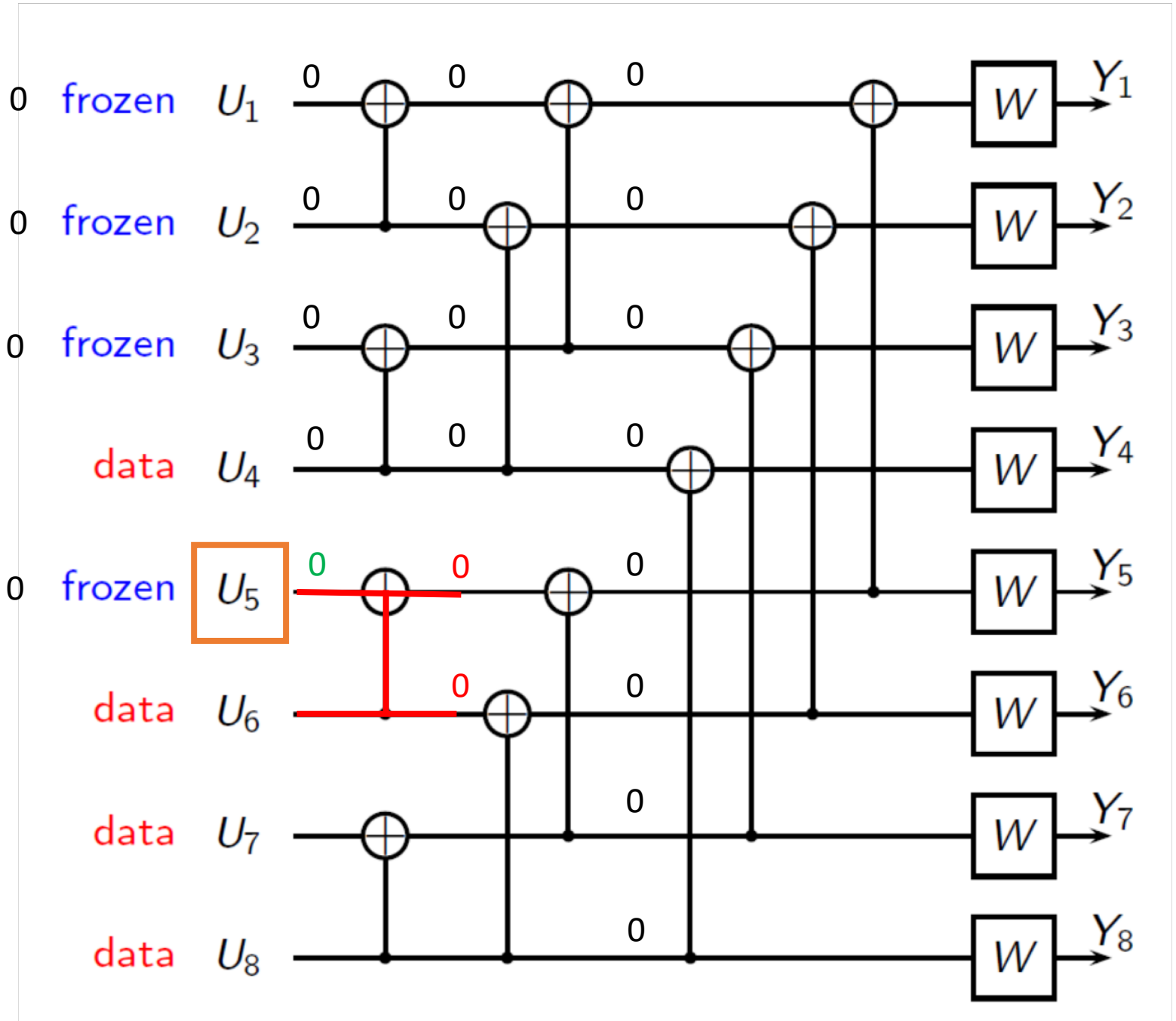

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

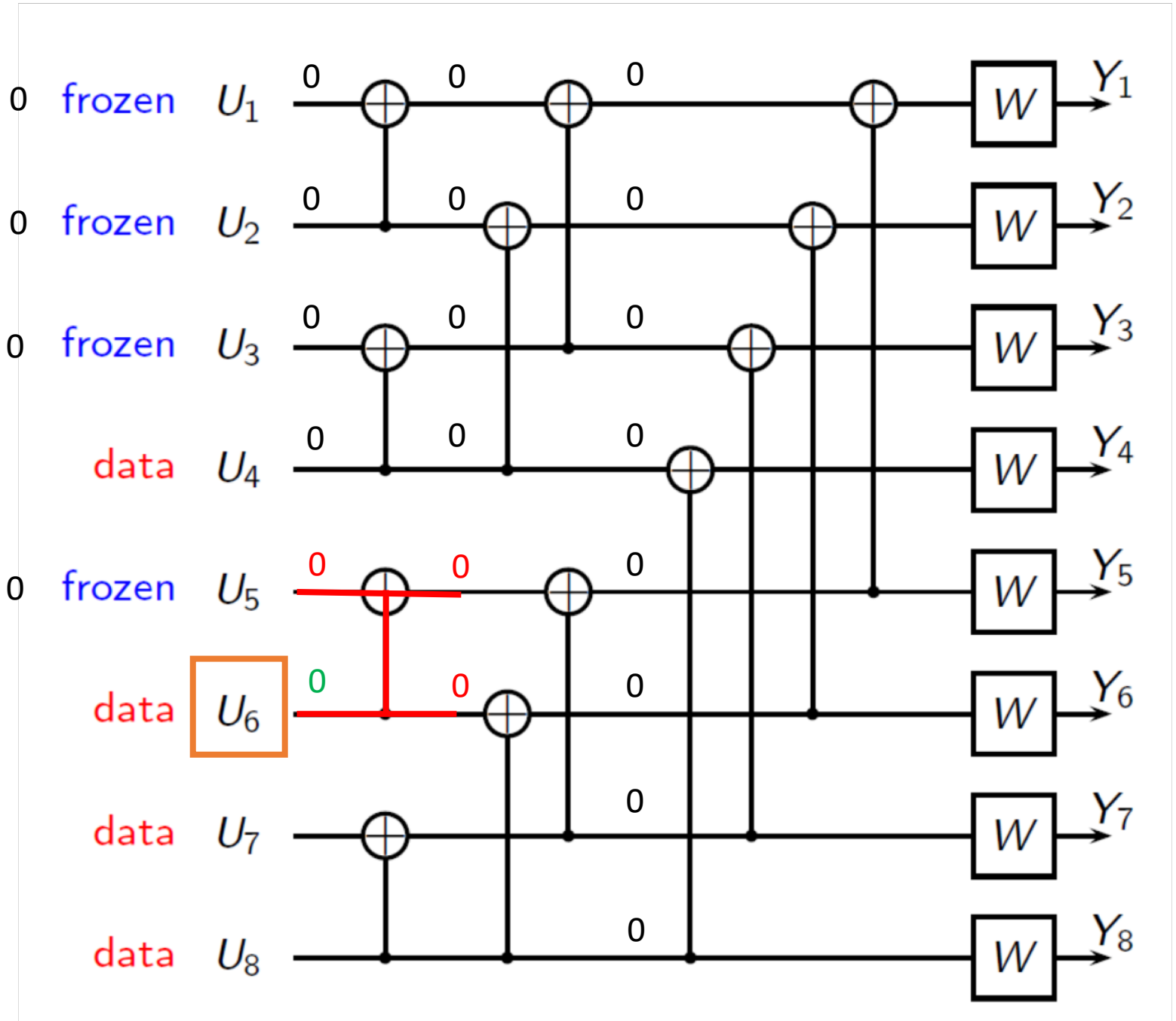

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0


0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
0
?
0
0
?
0

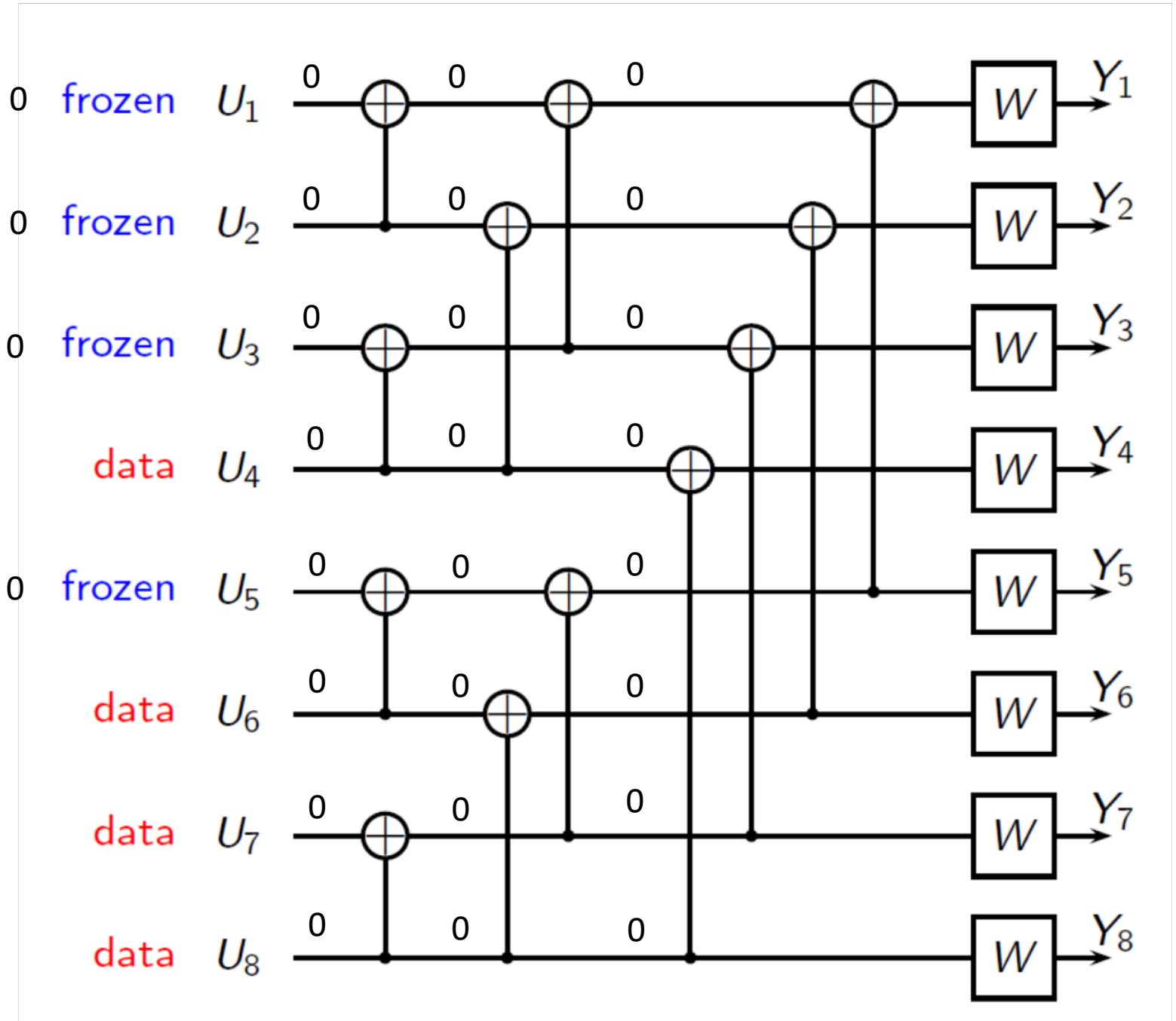

0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
?
0
0
?
0


0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
?
0
0
?
0


0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
?
0
0
?
0
0


0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1
Y_2
Y_3
Y_4
Y_5
Y_6
Y_7
Y_8
?
0
?
0
0
?
0
0

And so on until


0 frozen U_1
0 frozen U_2
0 frozen U_3
data U_4
0 frozen U_5
data U_6
data U_7
data U_8
W
Y_1 ?
Y_2 0
Y_3 0
Y_4 ?
Y_5 0
Y_6 0
Y_7 ?
Y_8 0

And so we succeed!!!

- If we fail to decode U_i for some i, we declare failure as we can't go ahead.
- Note that this can be suboptimal because some of the upcoming bits might be frozen and hence useful for decoding U_i. But we assumed all the bits yet to come are random when we decoded U_i.
- That being said, SC decoding can be shown to achieve capacity.
- Situation for other channels like BSC is slightly more complicated but the basic concepts remain the same.
- For short block lengths, we can do better, e.g., using list decoding and CRCs (see https://ieeexplore.ieee.org/document/7055304). These are one of the best codes at short block lengths and are part of 5G standards.
- Note that a lot of computation is reused and hence the complexity of decoding is $O(n \log n)$ where n is the block length (so n is the vertical height in the previous figures and $\log n$ is the horizontal width).