

Figure 1: The tile transition function and a DNA DX nanostructure for the tile.

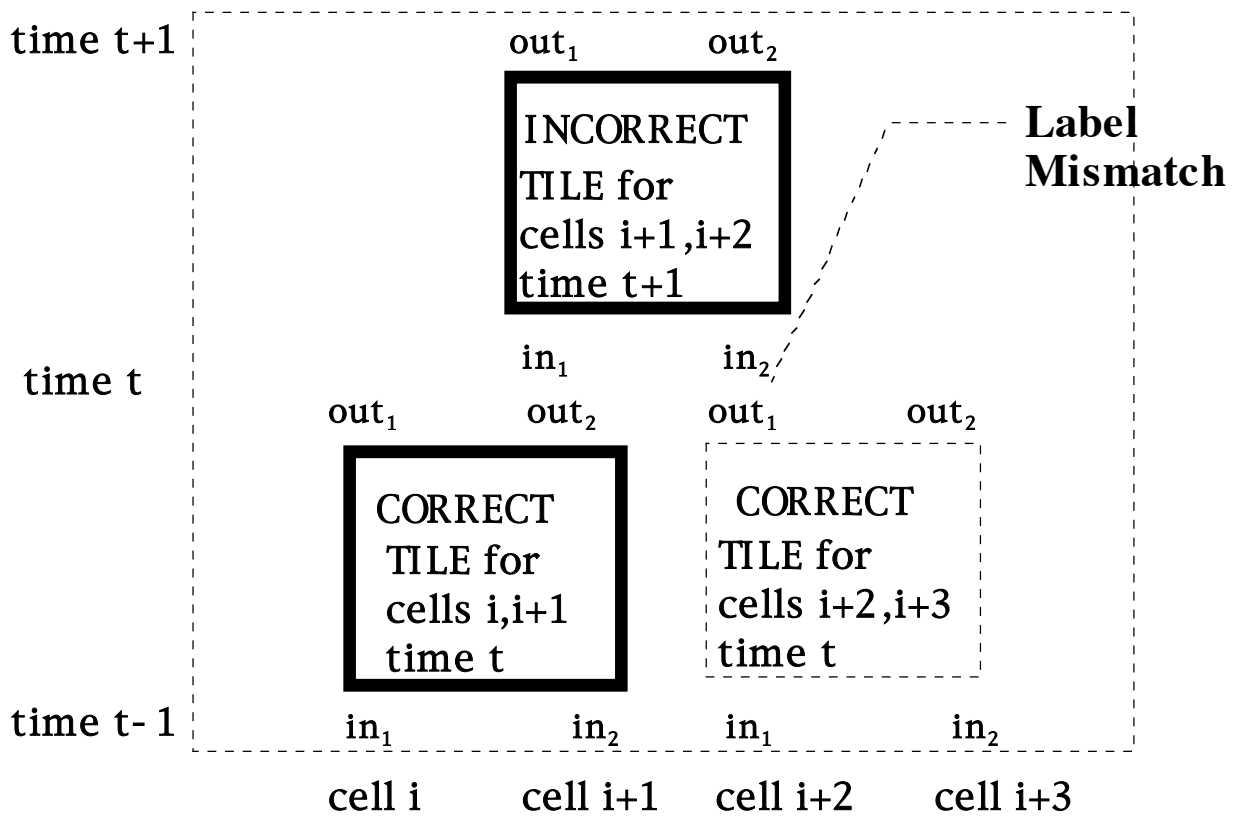


Figure 2: Unmediated self-assembly correctly places a tile at (i, t) but incorrectly places a tile at $(i+1, t+1)$. Later the tile at $(i+2, t)$ can not be placed, so assembly is blocked.

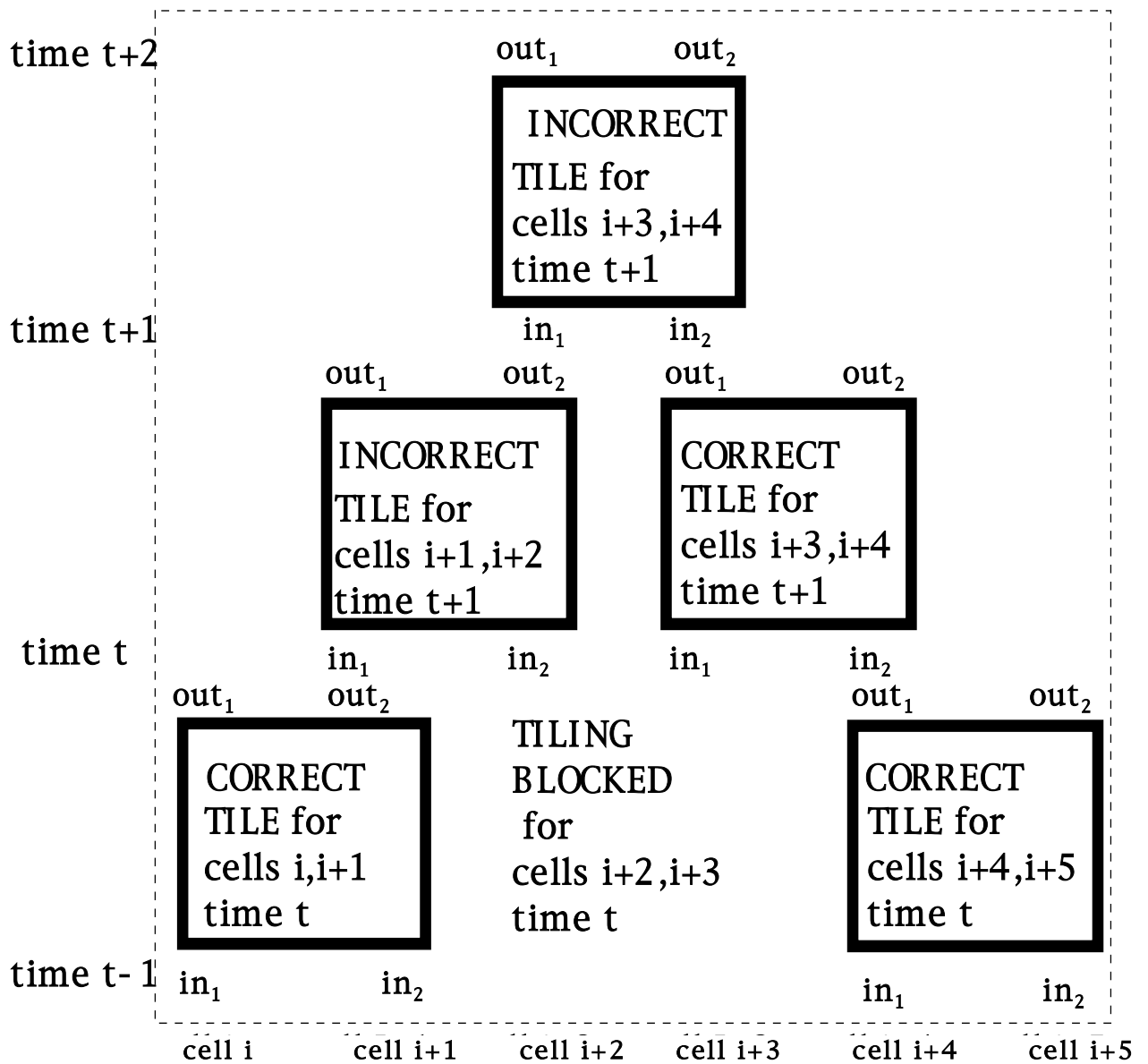


Figure 3: Unmediated self-assembly correctly places a tile at (i, t) , $(i+4, t)$ but incorrectly places a tile at $(i+1, t+1)$. Later tiles placed at $(i+3, t+1)$, $(i+2, t+2)$ reinforce the incorrect tiling, so the tile at $(i+2, t)$ can not be placed, and the assembly is permanently blocked.

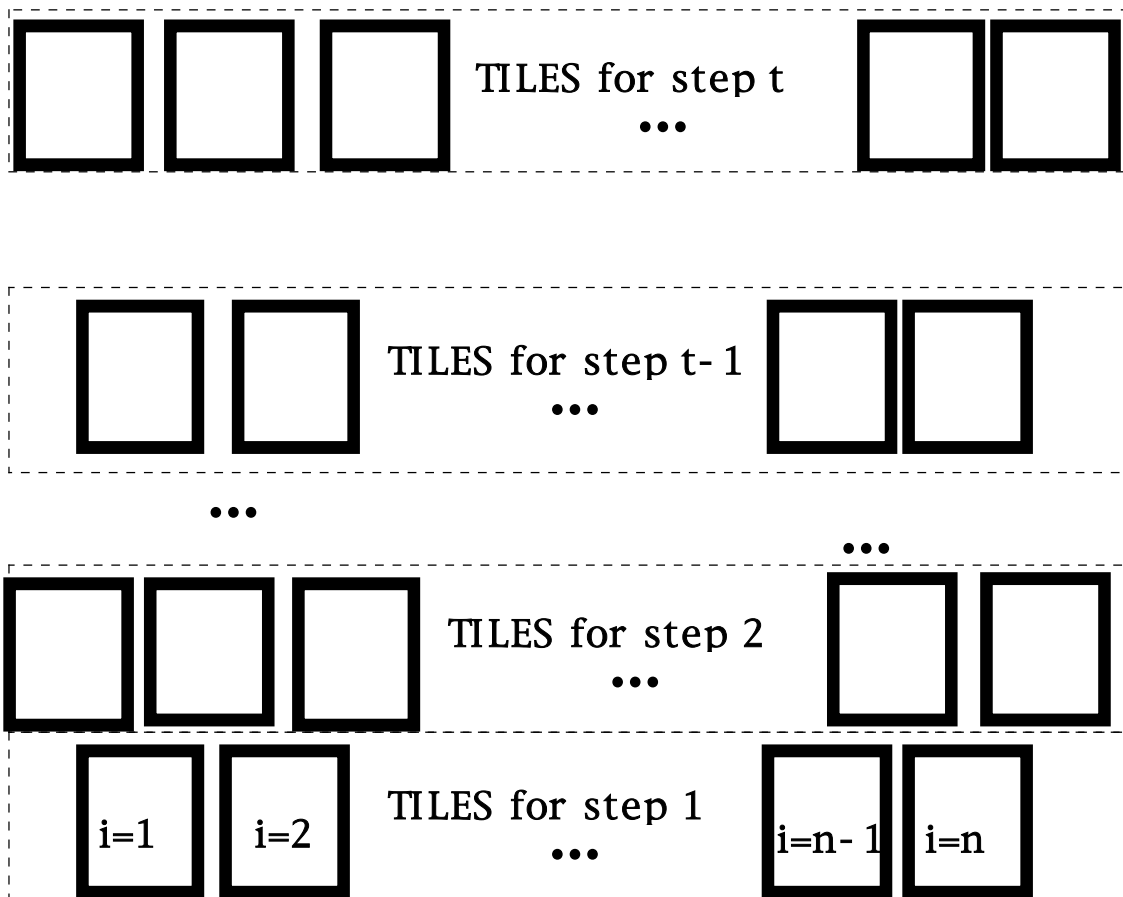


Figure 4: Step-wise assembly for step t.

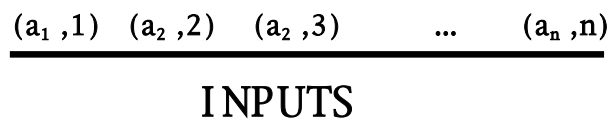


Figure 5: ssDNA encoding input n-vector fixed in a straight line.

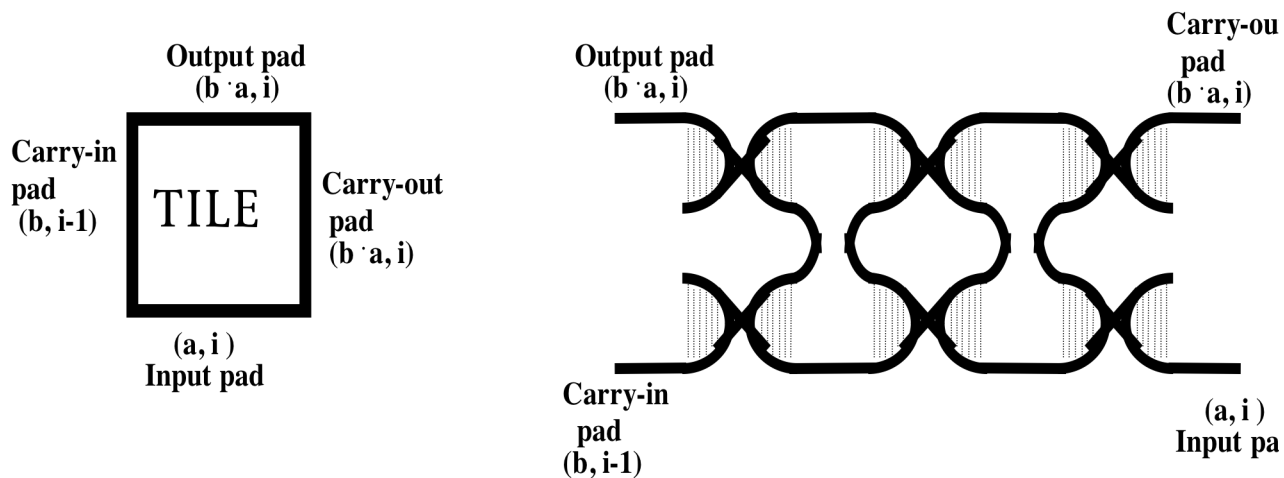


Figure 6: Square tile for sequential prefix computation and a DNA DX nanostructure for the tile.

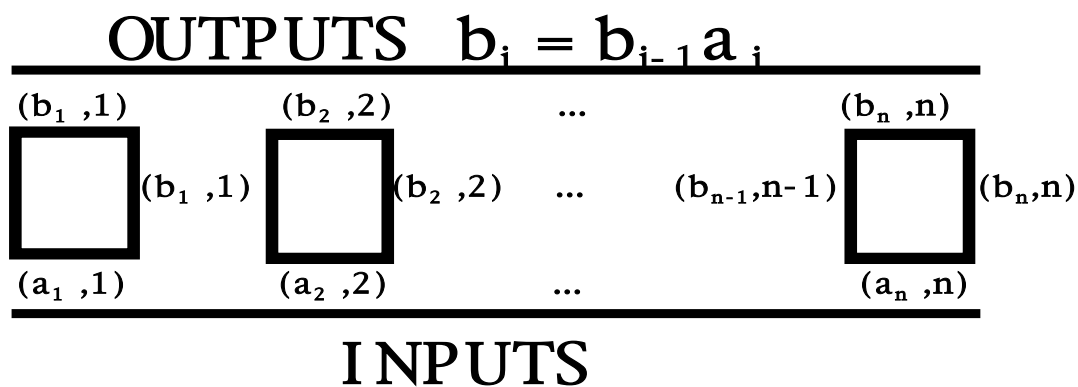
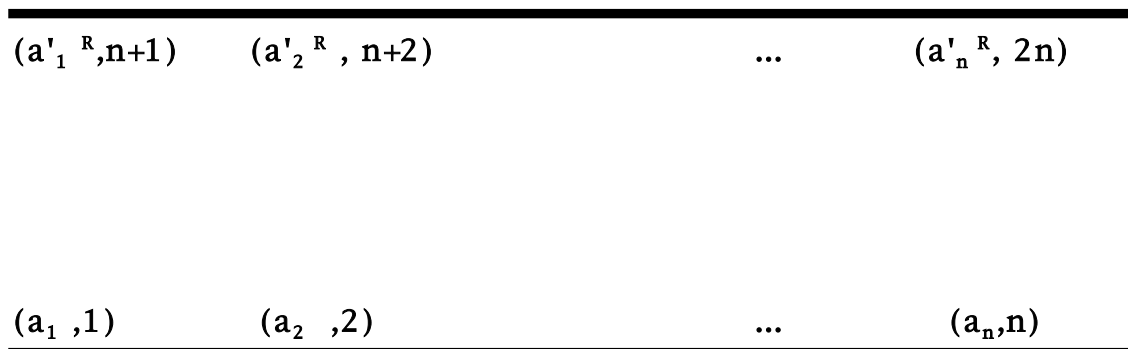


Figure 7: 2D assembly for sequential prefix computation.

INPUT₂



INPUT₁

Figure 8: Input ssDNA encoding two binary numbers.

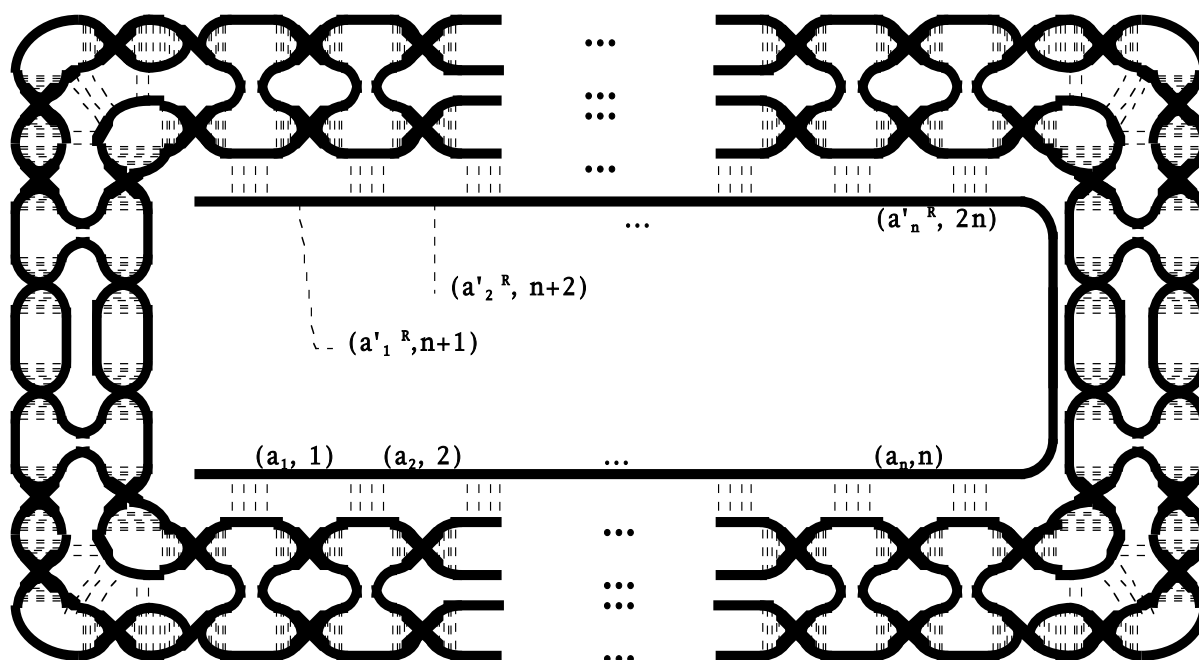


Figure 9: A DNA DX nanostructure for the rectangular frame.

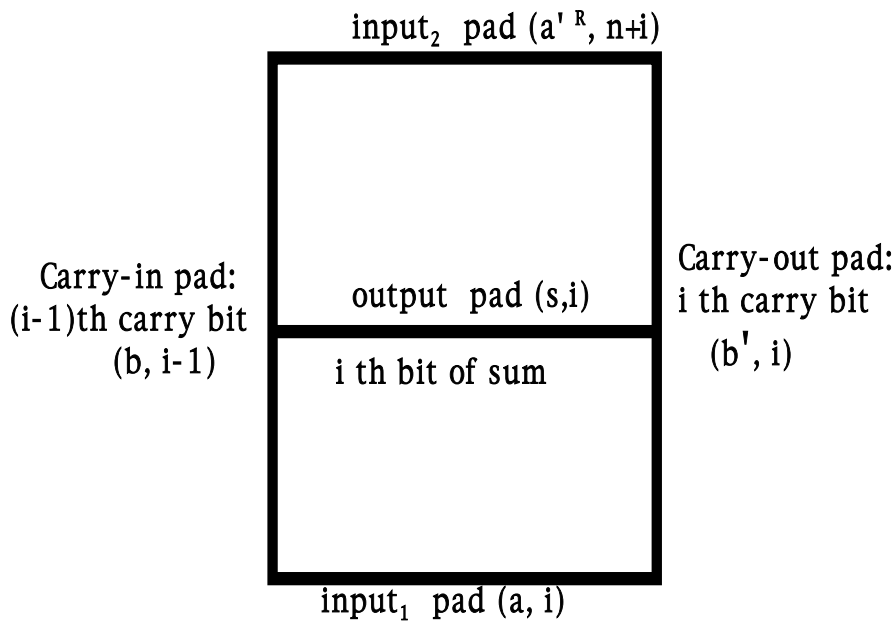


Figure 10: Square tile with middle segment, used for bit-serial addition.

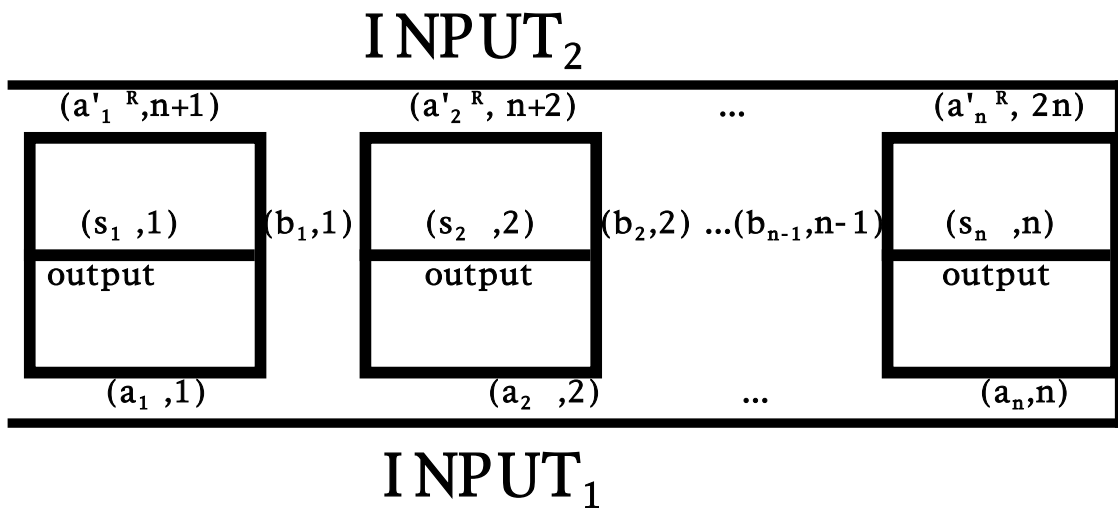


Figure 11: 2D assembly for bit-serial addition.

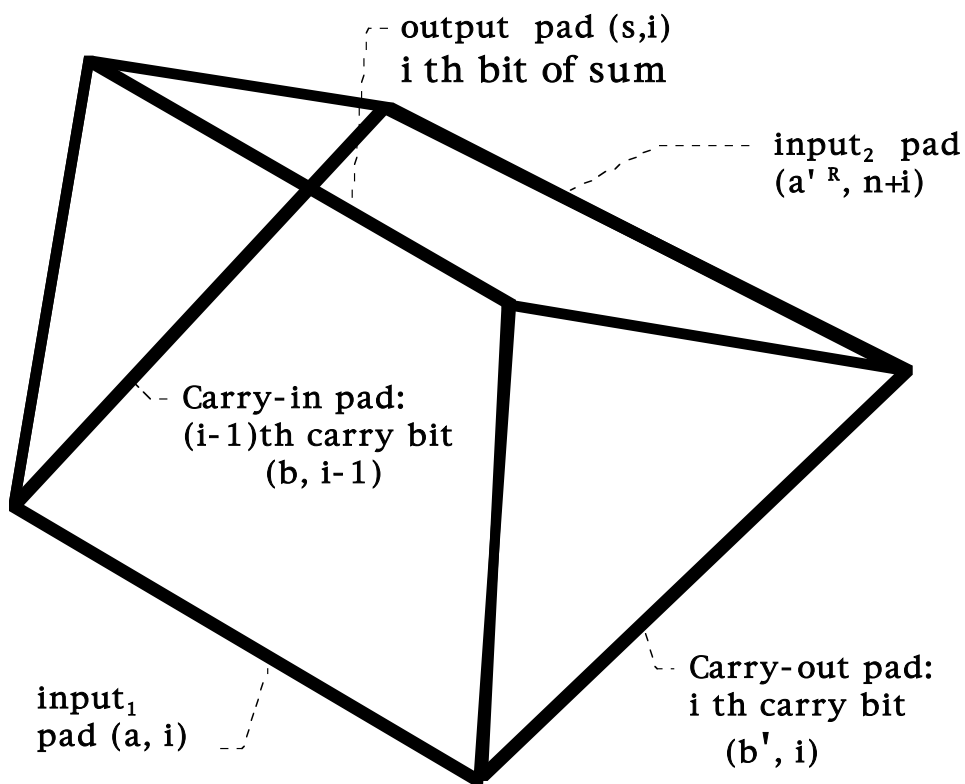


Figure 12: Alternative 3D polyhedral tile for bit-serial addition.

$(a_1,1,0) (a_2,2,0) (a_3,3,0) (a_4,4,0) (a_5,5,0) (a_6,6,0) (a_7,7,0) (a_8,8,0)$

Figure 13: Input ssDNA encoding an n-vector.

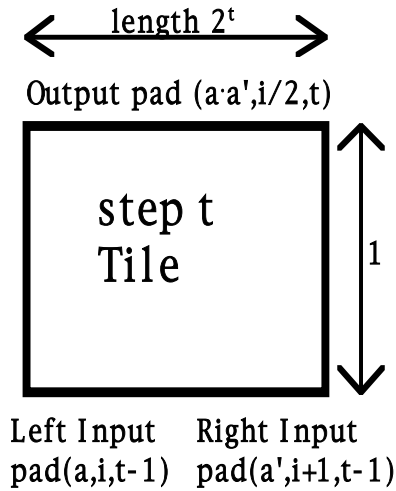


Figure 14: A Rectangular tile for parallel monoid sum computation.

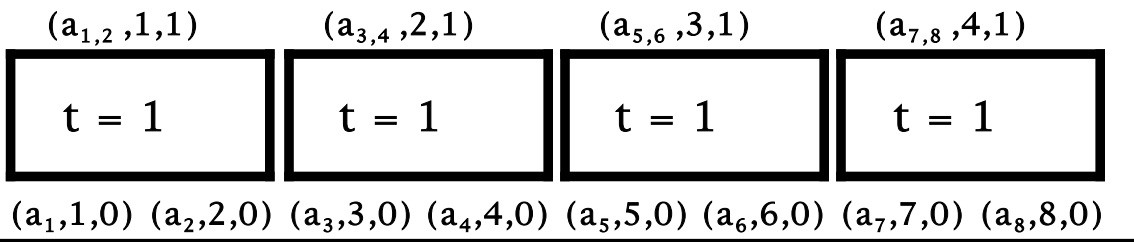


Figure 15: Step $t = 1$ of step-wise assembly for parallel monoid sum computes $a_{i,i+1} = a_i \cdot a_{i+1}$ for odd i .

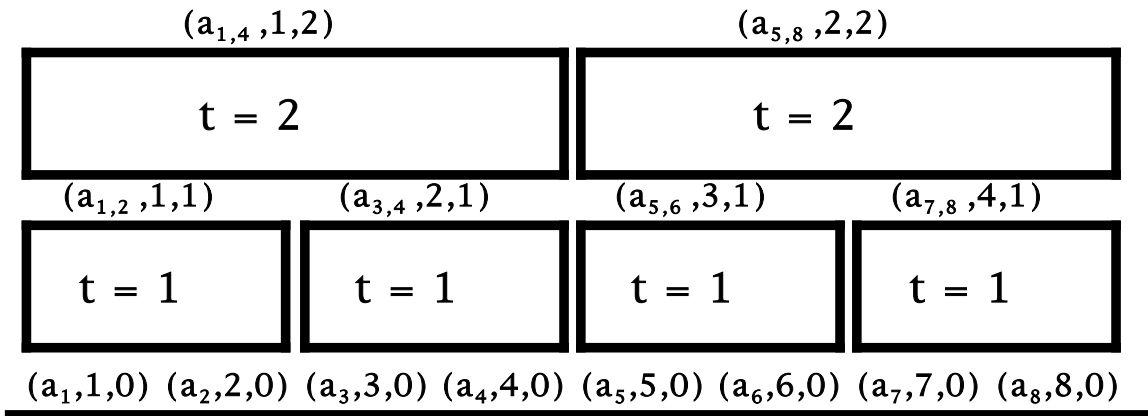


Figure 16: Step $t = 2$ of step-wise assembly for parallel monoid sum computes $b_4 = a_{1,4} = a_1 \cdot a_2 \cdot a_3 \cdot a_4$ and $a_{5,8} = a_5 \cdot a_6 \cdot a_7 \cdot a_8$.

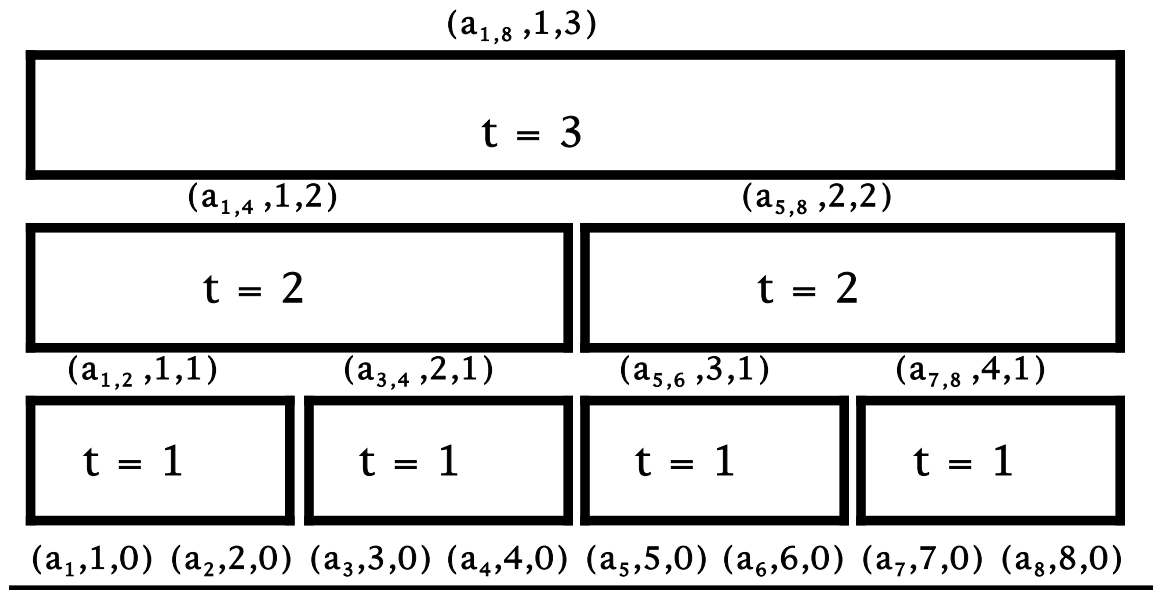


Figure 17: Final step $t = 3$ of assembly for parallel monoid sum computes $b_8 = a_{1,8} = a_1 \cdot a_2 \cdot a_3 \cdot a_4 \cdot a_5 \cdot a_6 \cdot a_7 \cdot a_8$.

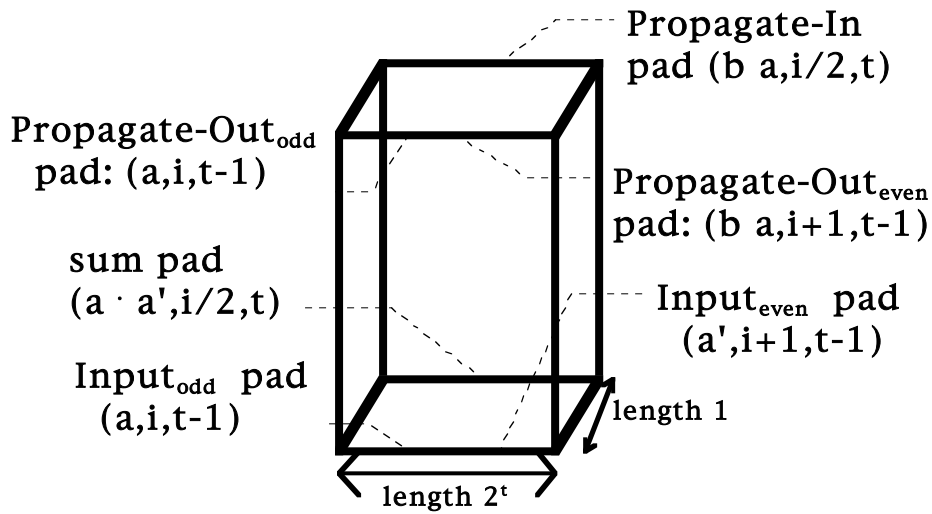


Figure 18: A 3D polyhedron tile for parallel prefix computation.

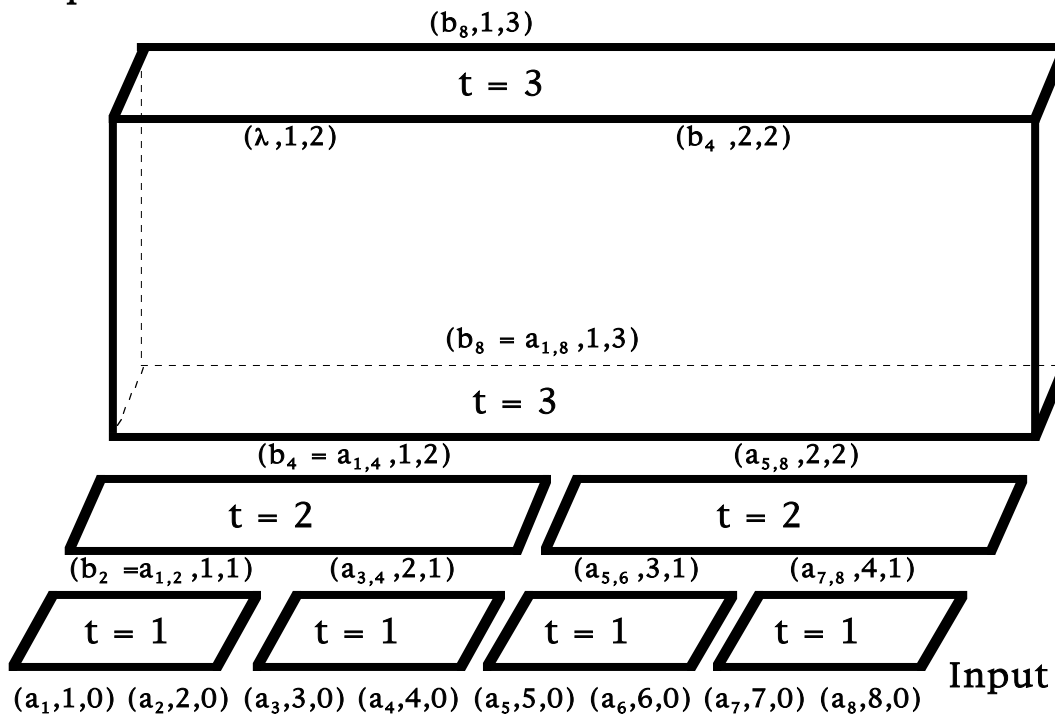


Figure 19: Stage $t=3$ of 3D parallel prefix assembly: computation of b_4 .

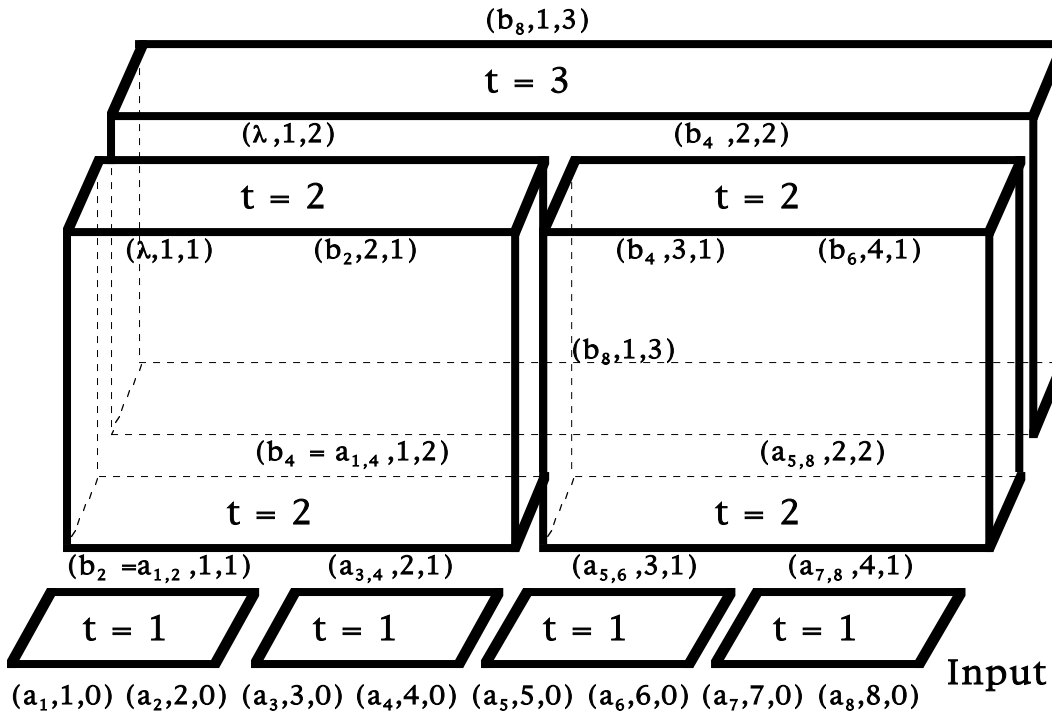


Figure 20: Stage $t=2$ of 3D parallel prefix assembly: computation of $b_2, b_4,$ and b_6 .

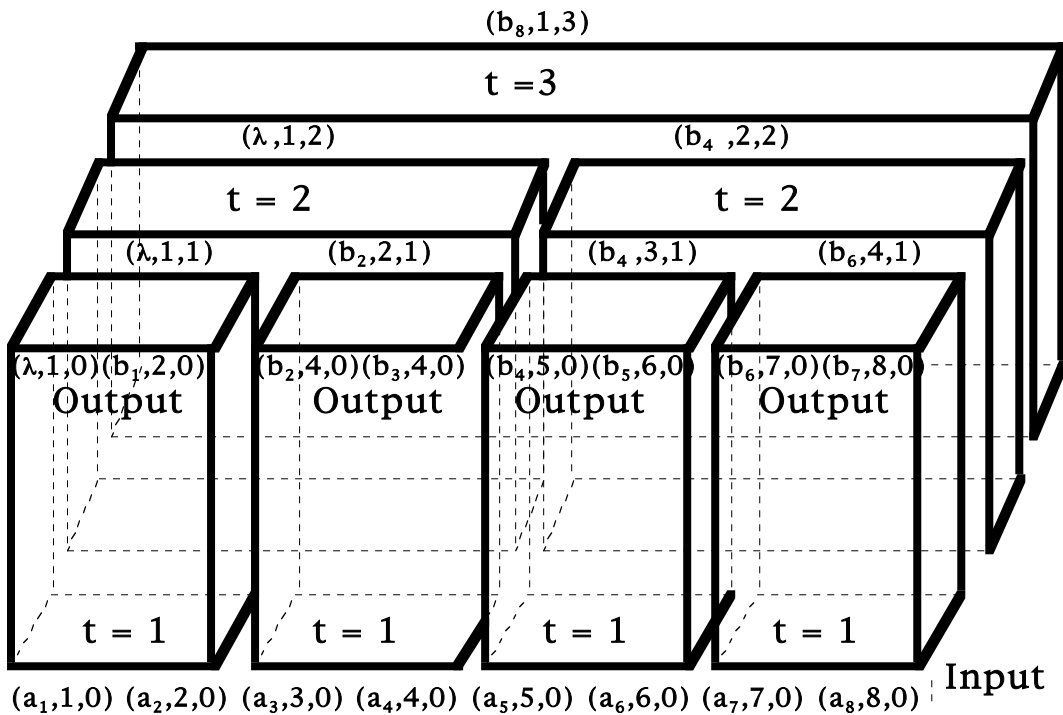


Figure 21: Final stage $t=1$ of 3D parallel prefix assembly: computation of b_1, \dots, b_7 .

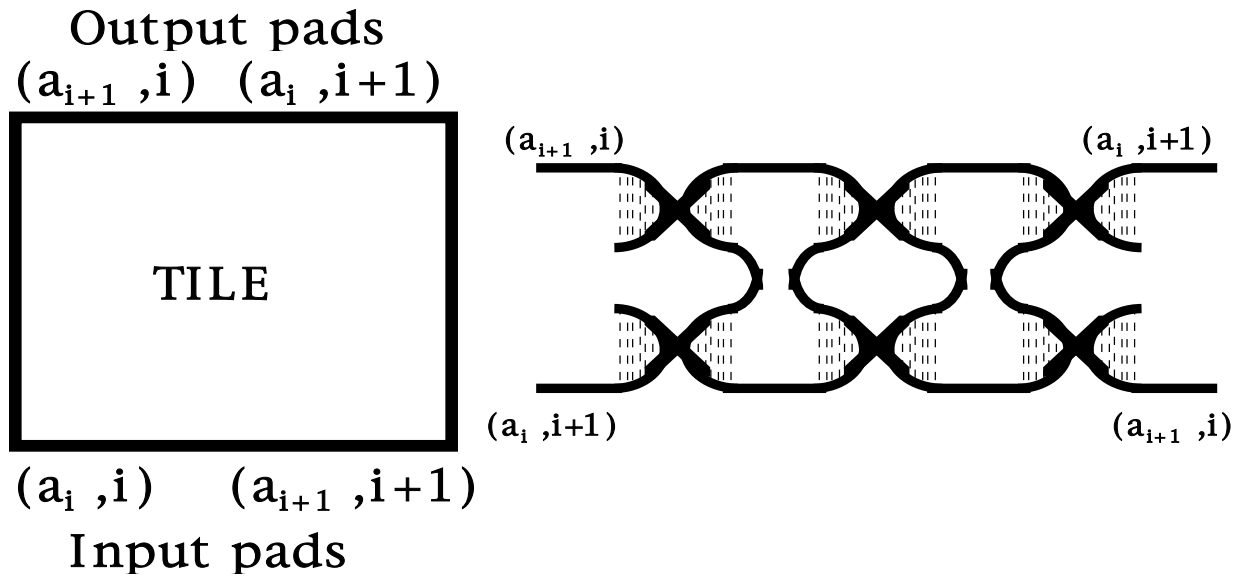


Figure 22: A square tile for pair-wise exchange and a DNA DX nanostructure for the tile.

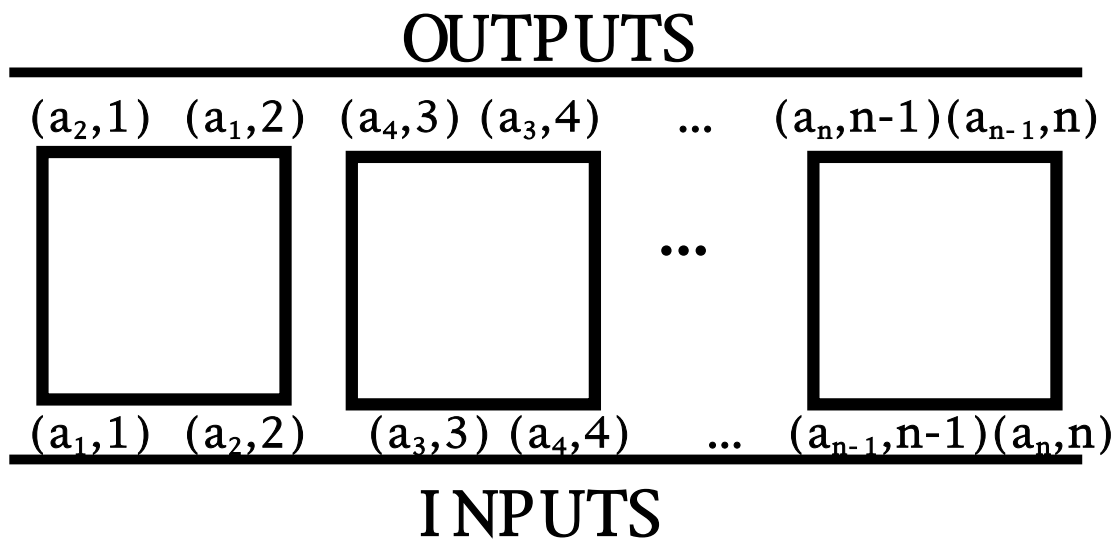
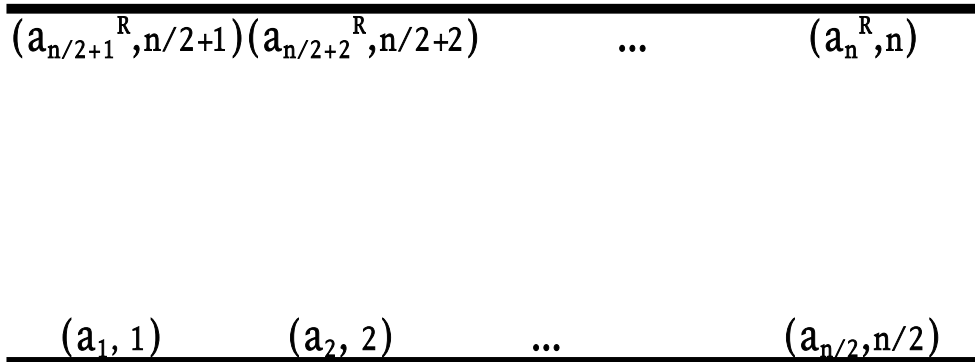


Figure 23: The 2D assembly for pair-wise exchange.

HIGH INPUTS



LOW INPUTS

Figure 24: An ssDNA encoding input n-vector.

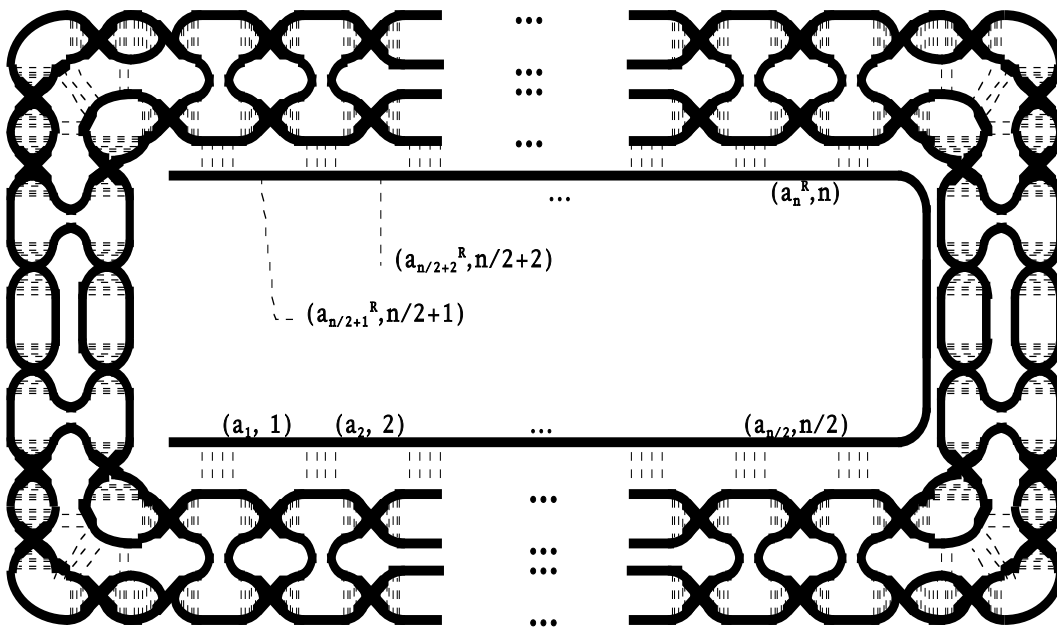


Figure 25: A DNA DX nanostructure for the rectangular frame.

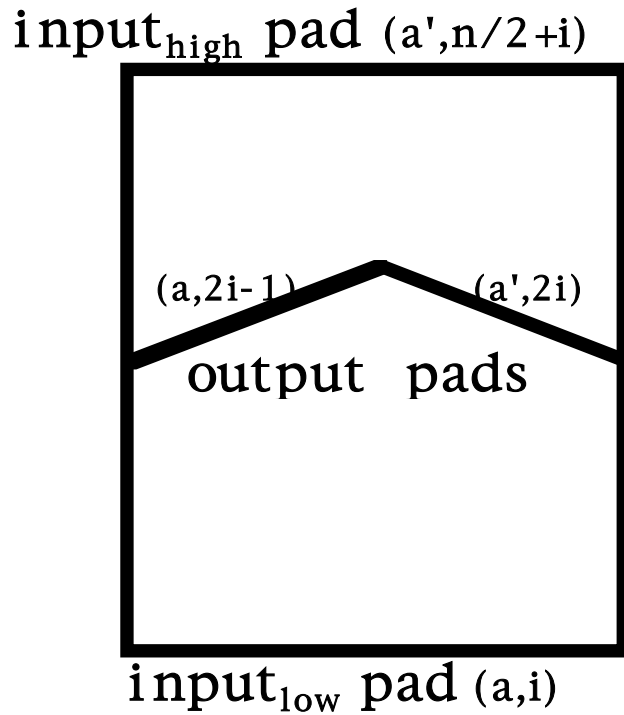


Figure 26: A square tile for perfect shuffle, with middle segment used for output.

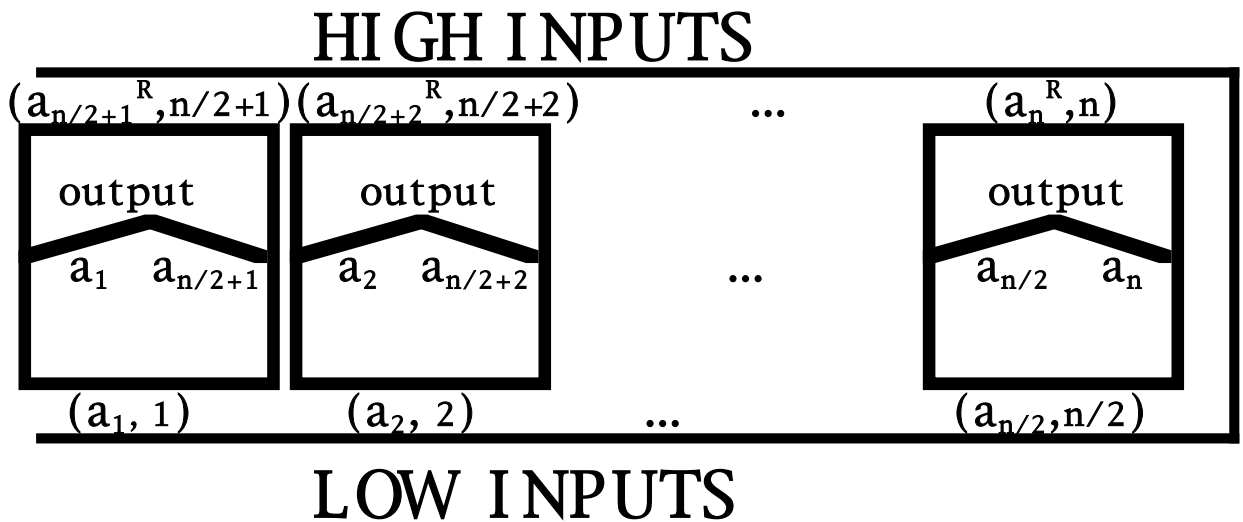


Figure 27: 2D assembly for perfect shuffle. The middle segments give output.

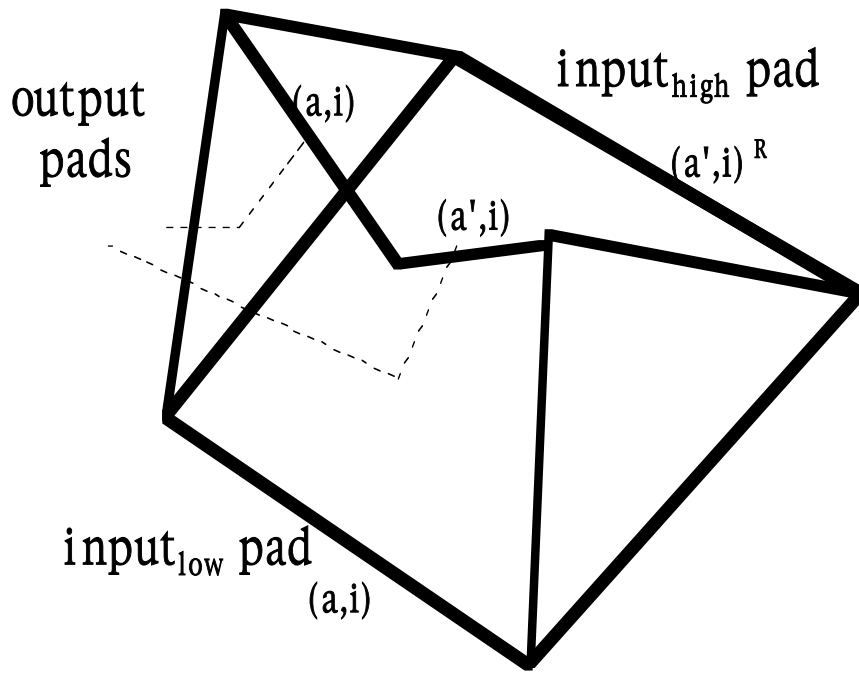


Figure 28: Alternative 3D polyhedral tile for perfect shuffle.