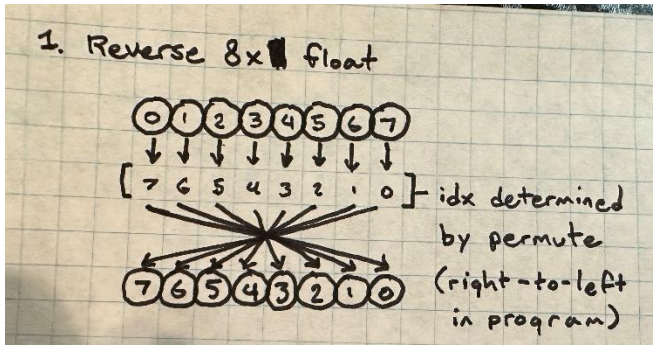
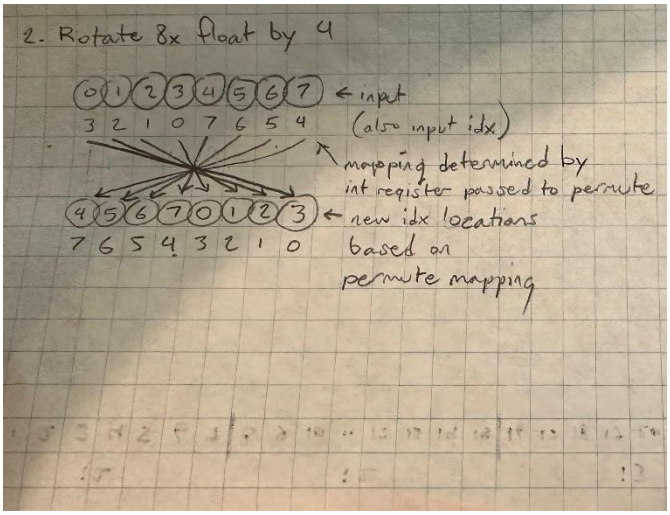
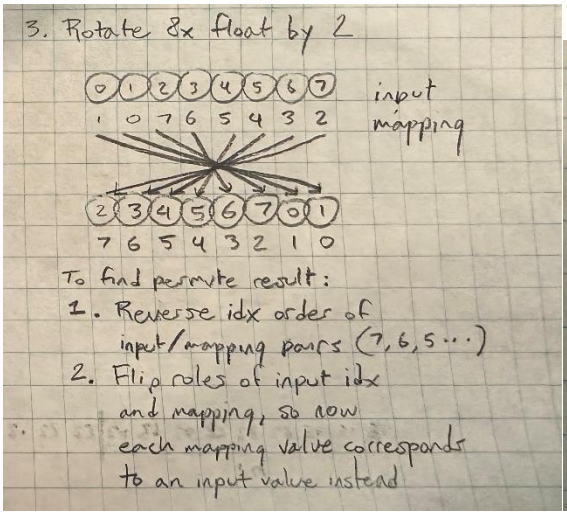
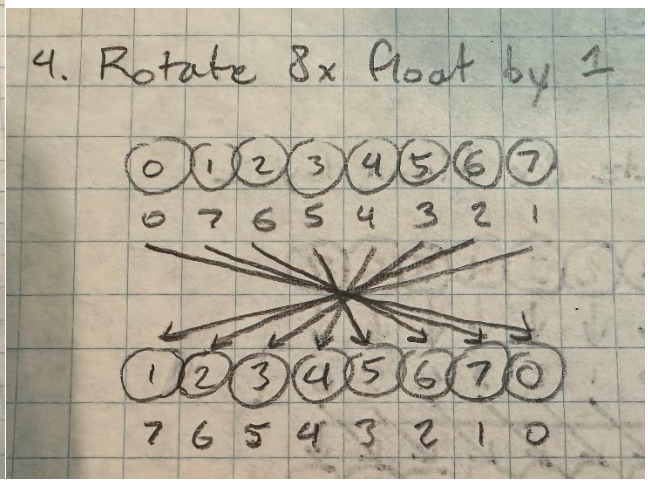
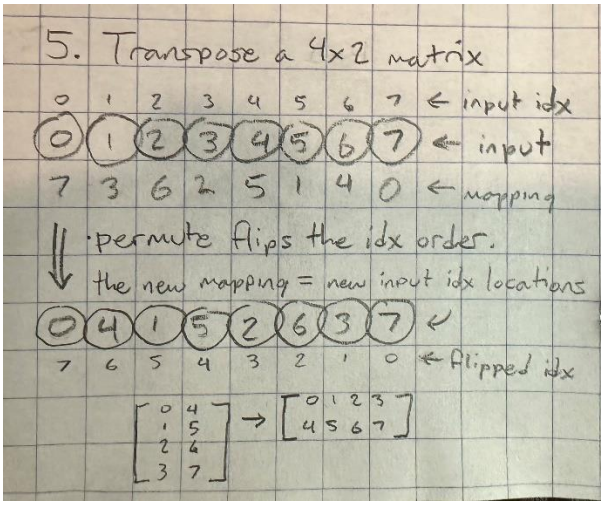
Daniel Yowell
Dr. Veras
PDN
9/26/2023
PG05 Figures

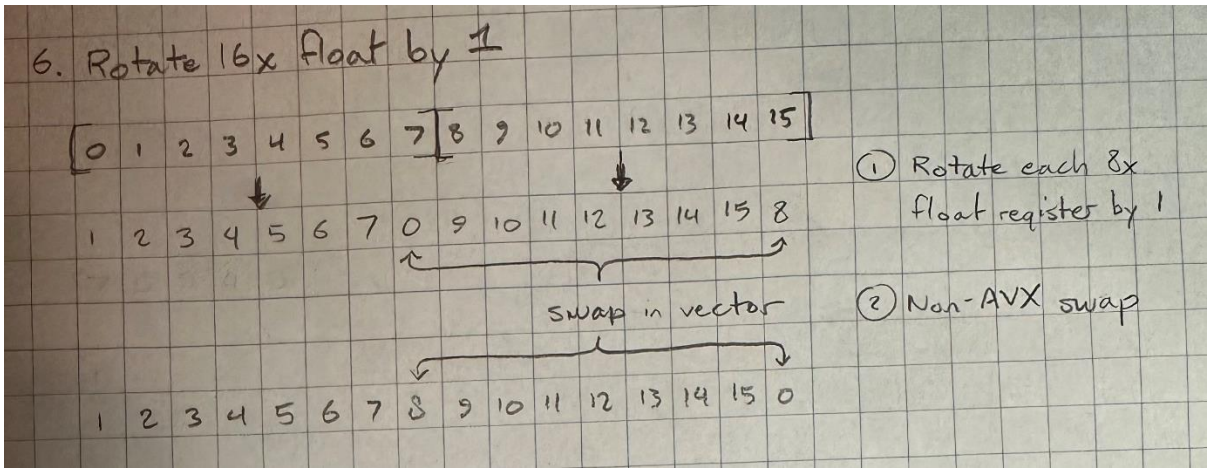




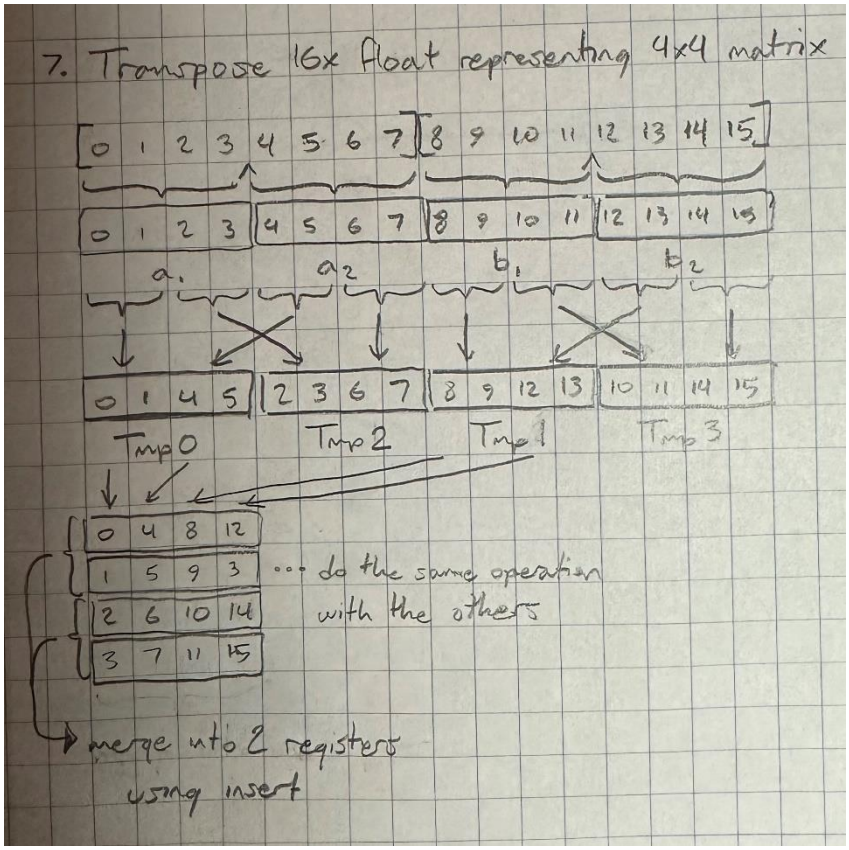




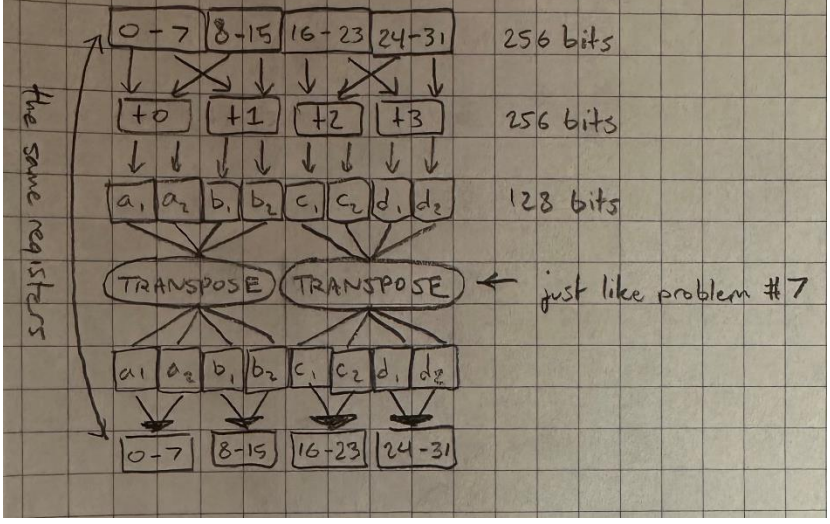

5. Transpose a 4x2 matrix
0 1 2 3 4 5 6 7 ← input idx
0 1 2 3 4 5 6 7 ← input
7 3 6 2 5 1 4 0 ← mapping
· permute flips the idx order.
the new mapping = new input idx locations
0 4 1 5 2 6 3 7
7 6 5 4 3 2 1 0 ← flipped idx
[0 4; 1 5; 2 6; 3 7] → [0 1 2 3; 4 5 6 7]


6. Rotate 16x float by 1
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
1 2 3 4 5 6 7 0 9 10 11 12 13 14 15 8
swap in vector
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 0
① Rotate each 8x float register by 1
② Non-AVX swap

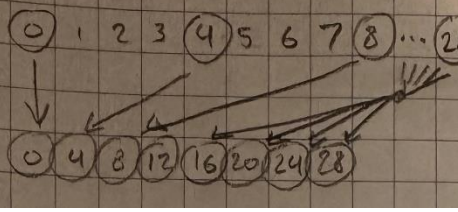
7. Transpose 16x float representing 4x4 matrix



8. Transpose 8x4 matrix represented by 8x floats

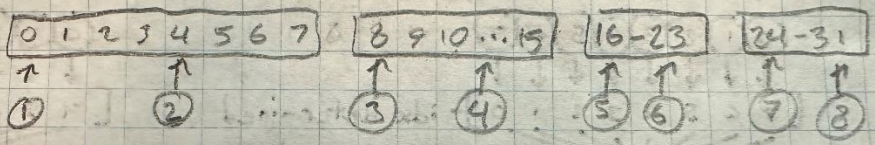


9. Gather at stride – 8x float



at every index * scale increment, select a value from src array

10. 8x float stride



To do this, use permute (within lane) and insert

11. 8x8 float matrix-vector using 8x registers