```
//threadIdx.x=0...31
//Loop 0:
ld.sm.128 rA[0],smA[0][threadIdx.x];// for loop 0
ld.sm.128 rB[0],smB[0][threadIdx.x];// for loop 0
ld.sm.128 rB[2],smB[2][threadIdx.x];// for loop 0
dfma accum[0][0],rA[0],rB[0],accum[0][0]; dfma accum[0][1],rA[0],rB[1],accum[0][1];
dfma accum[1][0],rA[1],rB[0],accum[1][0]; dfma accum[1][1],rA[1],rB[1],accum[1][1];
ld.sm.128 rA[2],smA[2][threadIdx.x]; // for loop 0
dfma accum[0][3],rA[0],rB[2],accum[0][2]; dfma accum[0][3],rA[0],rB[3],accum[0][3];
dfma accum[1][2],rA[1],rB[2],accum[1][2]; dfma accum[1][3],rA[1],rB[3],accum[1][3];
ld.sm.128 rA[0],smA[0][threadIdx.x]; // for loop 1
dfma accum[2][0],rA[2],rB[0],accum[2][0]; dfma accum[2][1],rA[2],rB[1],accum[2][1];
dfma accum[3][0],rA[3],rB[0],accum[3][0]; dfma accum[3][1],rA[3],rB[1],accum[3][1];
ld.sm.128 rB[0],smA[0][threadIdx.x]; // for loop 1
dfma accum[2][2],rA[2],rB[2],accum[2][2]; dfma accum[2][3],rA[2],rB[3],accum[2][3];
dfma accum[3][2],rA[3],rB[2],accum[3][2]; dfma accum[3][3],rA[3],rB[3],accum[3][3];
//Loop 1:
ld.sm.128 rA[0],smA[0][threadIdx.x]; //for loop 1
dfma accum[0][0],rA[0],rB[0],accum[0][0]; dfma accum[0][1],rA[0],rB[1],accum[0][1];
dfma accum[1][0],rA[1],rB[0],accum[1][0]; dfma accum[1][1],rA[1],rB[1],accum[1][1];
ld.sm.128 rA[2],smA[2][threadIdx.x]; // for loop 1
...//omitted other instructions for space because they're
cloned from loop 0
```