```
//
// (c) 2007 Vasily Volkov @ UC Berkeley
//

// GPU kernel: compute C = alpha A B' + beta C
__global__ void sgemmNT( const float *A, int lda,
                         const float *B, int ldb,
                         float* C, int ldc, int k,
                         float alpha, float beta   )
{
    int inx = threadIdx.x;
    int iny = threadIdx.y;
    int ibx = blockIdx.x * 32;
    int iby = blockIdx.y * 32;

    A += ibx + inx + __mul24( iny, lda );
    B += iby + inx + __mul24( iny, ldb );
    C += ibx + inx + __mul24( iby + iny, ldc );

    float c[16] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
    for( int i = 0; i < k; i += 4 )
    {
        __syncthreads();
        __shared__ float a[4][32];
        __shared__ float b[4][32];

        a[iny][inx] = A[i*lda];
        a[iny+2][inx] = A[(i+2)*lda];

        b[iny][inx]   = B[i*ldb];
        b[iny+2][inx] = B[(i+2)*ldb];
        __syncthreads();

        for( int j = 0; j < 4; j++ )
        {
            float _a = a[j][inx];
            float *_b = &b[j][0] + iny;

            c[0] += _a*_b[0];
            c[1] += _a*_b[2];
            c[2] += _a*_b[4];
            c[3] += _a*_b[6];
            c[4] += _a*_b[8];
            c[5] += _a*_b[10];
            c[6] += _a*_b[12];
            c[7] += _a*_b[14];
            c[8] += _a*_b[16];
            c[9] += _a*_b[18];
            c[10] += _a*_b[20];
            c[11] += _a*_b[22];
            c[12] += _a*_b[24];
            c[13] += _a*_b[26];
            c[14] += _a*_b[28];
            c[15] += _a*_b[30];
        }
    }

    for( int i = 0; i < 16; i++, C += 2*ldc )
        C[0] = alpha * c[i] + beta * C[0];
}


void ourSgemm (char transa, char transb,
               int m, int n, int k,
               float alpha,
               const float *A, int lda,
               const float *B, int ldb,
               float beta,
               float *C, int ldc)
{
    assert( (transa == 'N' || transa == 'n') &&
            (transb == 'T' || transb == 't') &&
            ((m|n|k|lda|ldb)&31) == 0,
            "unsupported parameters in ourSgemm()" );

    dim3 grid( m/32, n/32, 1 );
    dim3 threads2( 32, 2, 1 );
    sgemmNT<<<grid, threads2>>>( A, lda,
                                 B, ldb,
                                 C, ldc,
                                 k, alpha, beta );
}
```