

```

__global__ void MatrixMulKernel(float* d_M, float* d_N, float* d_P, int Width) {

    // Calculate the row index of the d_P element and d_M
    int Row = blockIdx.y*blockDim.y+threadIdx.y;

    // Calculate the column index of d_P and d_N
    int Col = blockIdx.x*blockDim.x+threadIdx.x;

    if ((Row < Width) && (Col < Width)) {
        float Pvalue = 0;
        // each thread computes one element of the block sub-matrix
        for (int k = 0; k < Width; ++k) {
            Pvalue += d_M[Row*Width+k]*d_N[k*Width+Col];
        }
        d_P[Row*Width+Col] = Pvalue;
    }
}

```