

```

__global__ void convolution_1D_basic_kernel(float *N, float *P, int Mask_Width,
int Width) {

int i = blockIdx.x*blockDim.x + threadIdx.x;
__shared__ float N_ds[(TILE_SIZE + MAX_MASK_WIDTH - 1)];

int n = Mask_Width/2;

int halo_index_left = (blockIdx.x - 1)*blockDim.x + threadIdx.x;
if (threadIdx.x >= blockDim.x - n) {
    N_ds[threadIdx.x - (blockDim.x - n)] =
        (halo_index_left < 0) ? 0 : N[halo_index_left];
}

N_ds[n + threadIdx.x] = N[blockIdx.x*blockDim.x + threadIdx.x];

int halo_index_right = (blockIdx.x + 1)*blockDim.x + threadIdx.x;
if (threadIdx.x < n) {
    N_ds[n + blockDim.x + threadIdx.x] =
        (halo_index_right >= Width) ? 0 : N[halo_index_right];
}

__syncthreads();

float Pvalue = 0;
for(int j = 0; j < Mask_Width; j++) {
    Pvalue += N_ds[threadIdx.x + j]*N[j];
}
P[i] = Pvalue;
}

```