SUMMATION (HYPERCUBE SIMD):

```
 Parameter  $n$   {Number of elements to add}
            $p$   {Number of processing elements}
Global      $j$
Local       $local.size, local.value[1 .. \lceil n/p \rceil], sum, tmp$

{
  for all $p_i$, where $0 \leq i \leq p-1$ do
          if $i < (n \text{ modulo } p)$ then
              $local.size \leftarrow \lceil n/p \rceil$
          else
              $local.size \leftarrow \lfloor n/p \rfloor$
          endif
              $sum \leftarrow 0$;
  endfor

 for ($j = 1$ ; $j <= \lceil n/p \rceil$ ; $j++$ ) do
         for all $p_i$, where $0 \leq i \leq p-1$ do
           if $local.size \geq j$ then
               $sum \leftarrow sum + local.value[j]$
           endif
         endfor
endfor

for ( $j = \log p - 1$; $j >= 0$; $j--$) do
   for all $p_i$, where $0 \leq i \leq p-1$ do
        if $i < 2^j$ then
             $tmp \Leftarrow [i+2^j]sum$
             $sum \leftarrow sum + tmp$
          endif
     endfor
endfor
}
```

Analyze the time complexity.

SUMMATION (SHUFFLE-EXCHANGE SIMD):

```
 Parameter  $n$   (Number of elements to add)
            $p$   (Number of processing elements)
Global      $j$
Local       $local.size, local.value[1 \ldots \lceil n/p \rceil], sum, tmp$

{
 for all $p_i$ , where  $0 \le i \le p-1$ do
      if  $i < (n \text{ modulo } p)$ then
             $local.size \leftarrow \lceil n/p \rceil$
      else
               $local.size \leftarrow \lfloor n/p \rfloor$

   endif
          $sum \leftarrow 0$;
 endfor

for  ($j$ =1 ; $j$<= $\lceil n/p \rceil$;  $j$++  )  do
    for  all $p_i$ ,   where $0 \le i \le p-1$ do
         if $local.size \ge j$ then
          $sum \leftarrow sum + local.value[j]$
         endif
    endfor
endfor

for ( $j$ =0; $j$<= log $p$-1;  $j$++) do
  for all $p_i$ , where $0 \le i \le p-1$ do
      $shuffle(sum) \Leftarrow sum$
        $exchange(tmp) \Leftarrow sum$
        $sum \leftarrow sum + tmp$
  endfor
endfor
}                                   Analyze the time complexity
```

SUMMATION (2-D MESH SIMD):

```
 Parameter  $l$   (Mesh has size $l* l$)
 Global     $i$
Local       $tmp, sum, local.size, local.value \lceil n/p \rceil$
{
  for all $p_{i,j}$, where  $1 \le i, j \le l$ do
           if  ( $i < (n$ modulo $p)$  &&  $j==1$ ) then
                  $local.size \leftarrow \lceil n/p \rceil$
           else
                      $local.size \leftarrow \lfloor n/p \rfloor$
       endif
              $sum \leftarrow 0$;
  endfor

for  ($k$ =1 ; $k$<= $\lceil n/p \rceil$;  $k$++  )  do
        for  all $p_{i,j}$  where $1 \le i, j \le l$ do
                 if $local.size \ge k$   then
                  $sum \leftarrow sum + local.value[k]$
                endif
        endfor
endfor

 for  ( $i = l-1$;  $i >=1$; $i$  --)  do
      for all $p_{j,i}$  where  $1 \le j \le l$ do
(Processing elements in column  $i$ active)
        $tmp \Leftarrow east(sum)$
        $sum \leftarrow sum + tmp$
      endfor
  endfor
for  ($i$ = $l - 1$;  $i >= 1$ ; $i$ --) do
      for all $p_{i,1}$   do
         $tmp \Leftarrow south(sum)$
         $sum \leftarrow sum + tmp$
       endfor
  endfor
  }
```