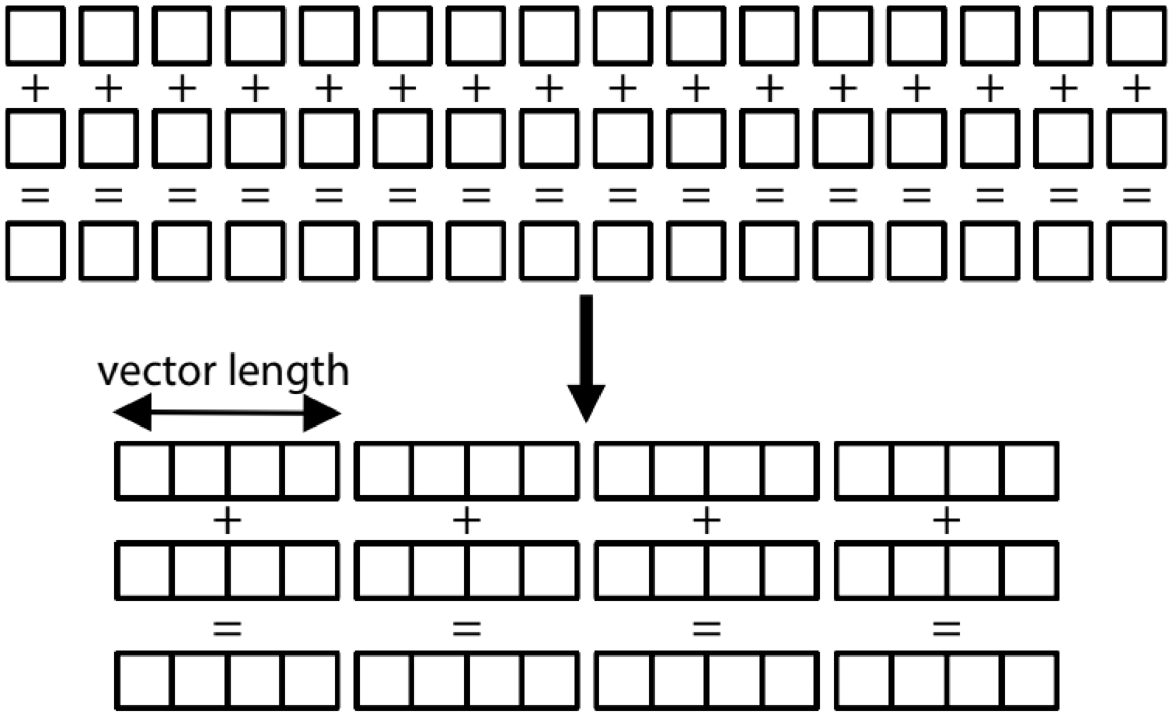
An example of vectorization – Vector instructions improve the performance by processing multiple data items concurrently.

```
for (int i=0; i<16; i++)
    a[i] = b[i] + c[i];
```



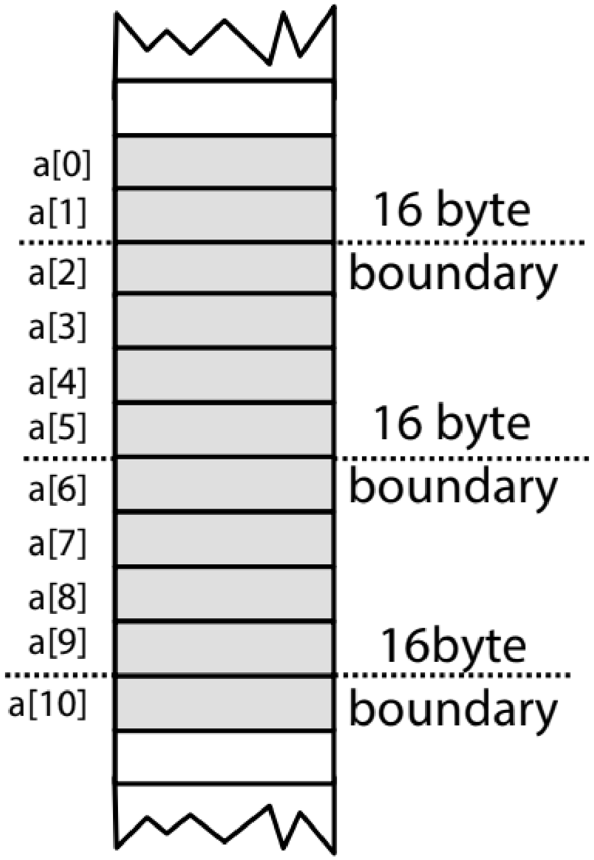
Scalar instructions

32 loads
16 adds
16 stores

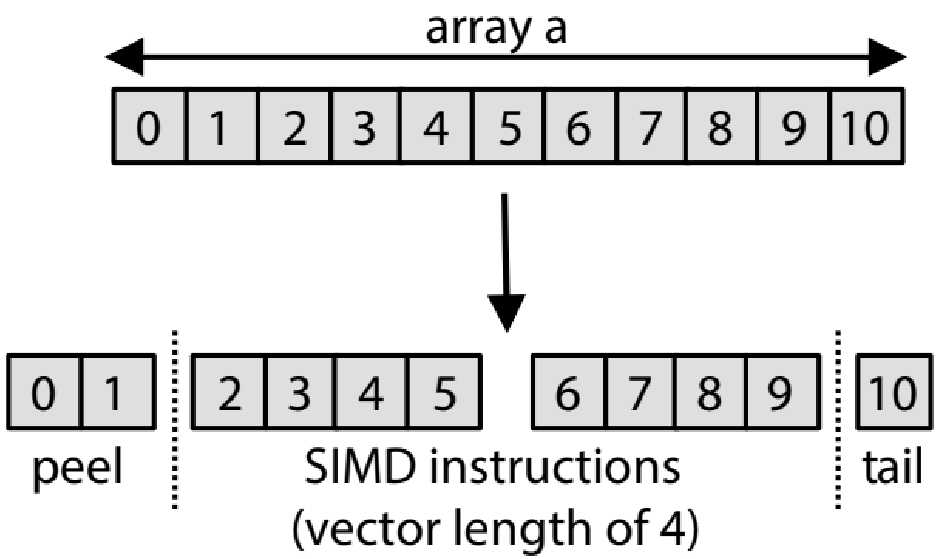
SIMD instructions

8 loads
4 adds
4 stores

Loop modifications needed to vectorize a loop using SIMD instructions – To enforce alignment and ensure the remaining loop length is a multiple of the vector length, the compiler may need to peel off iterations and also treat the tail end separately.

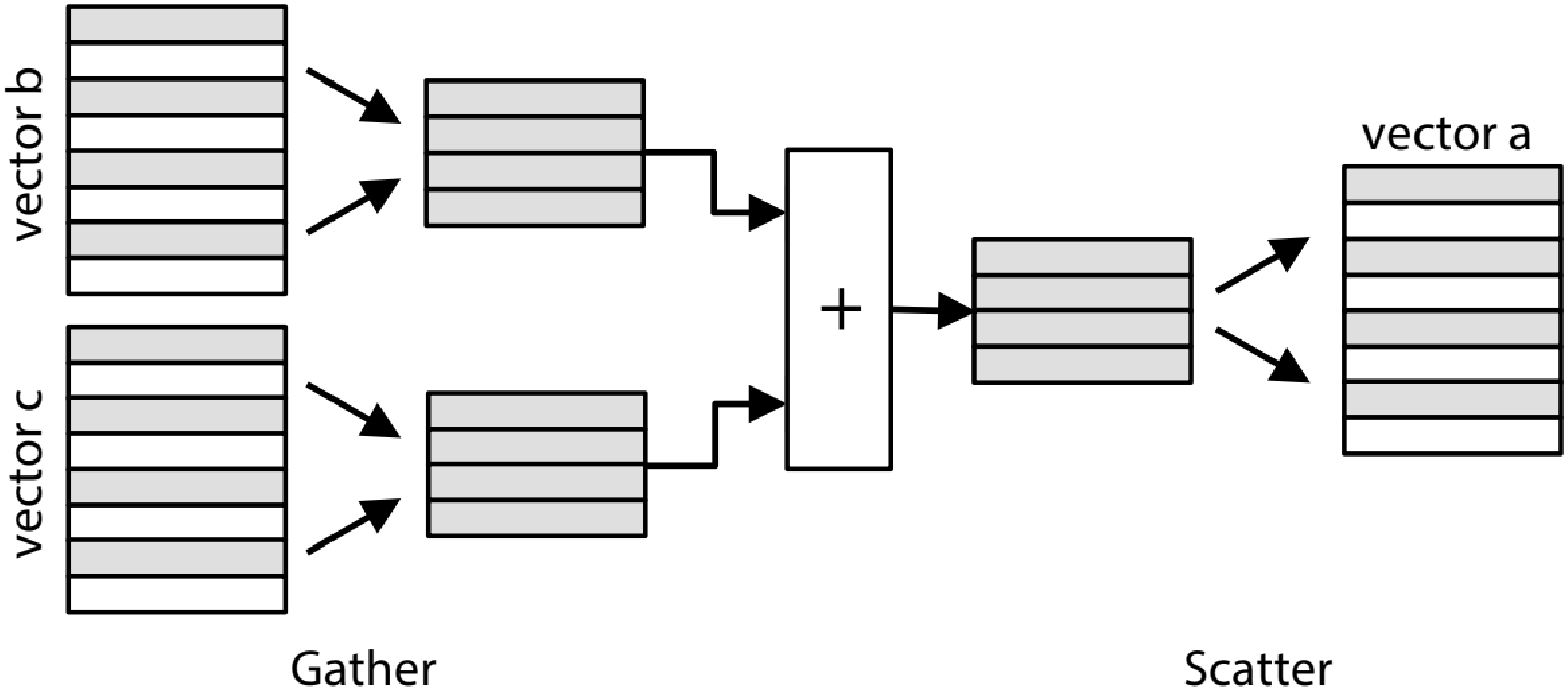


```
for (int i=0; i<11; i++)
    a[i] = 1.0;
```



Gather and scatter scalar data elements in and out of a vector – Scalar data elements are gathered into vectors, operated on as a vector, and then scattered back to their destination locations.

```
for (int i=0; i<8; i+=2)
    a[i] = b[i] + c[i];
```



Combining thread and SIMD parallelism – Thread and SIMD parallelism are used to execute a loop.

```
#pragma omp for simd
for (int i=0; i<32; i++)
    a[i] += 1;
```

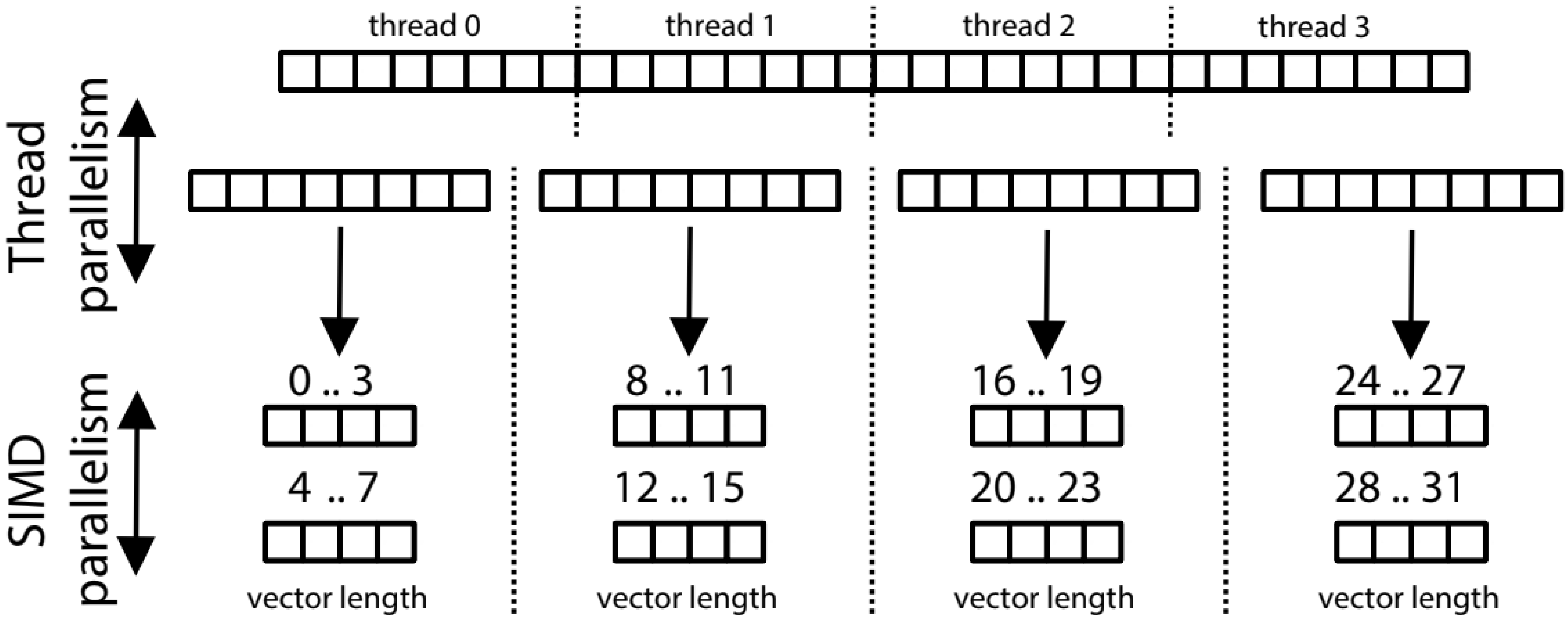
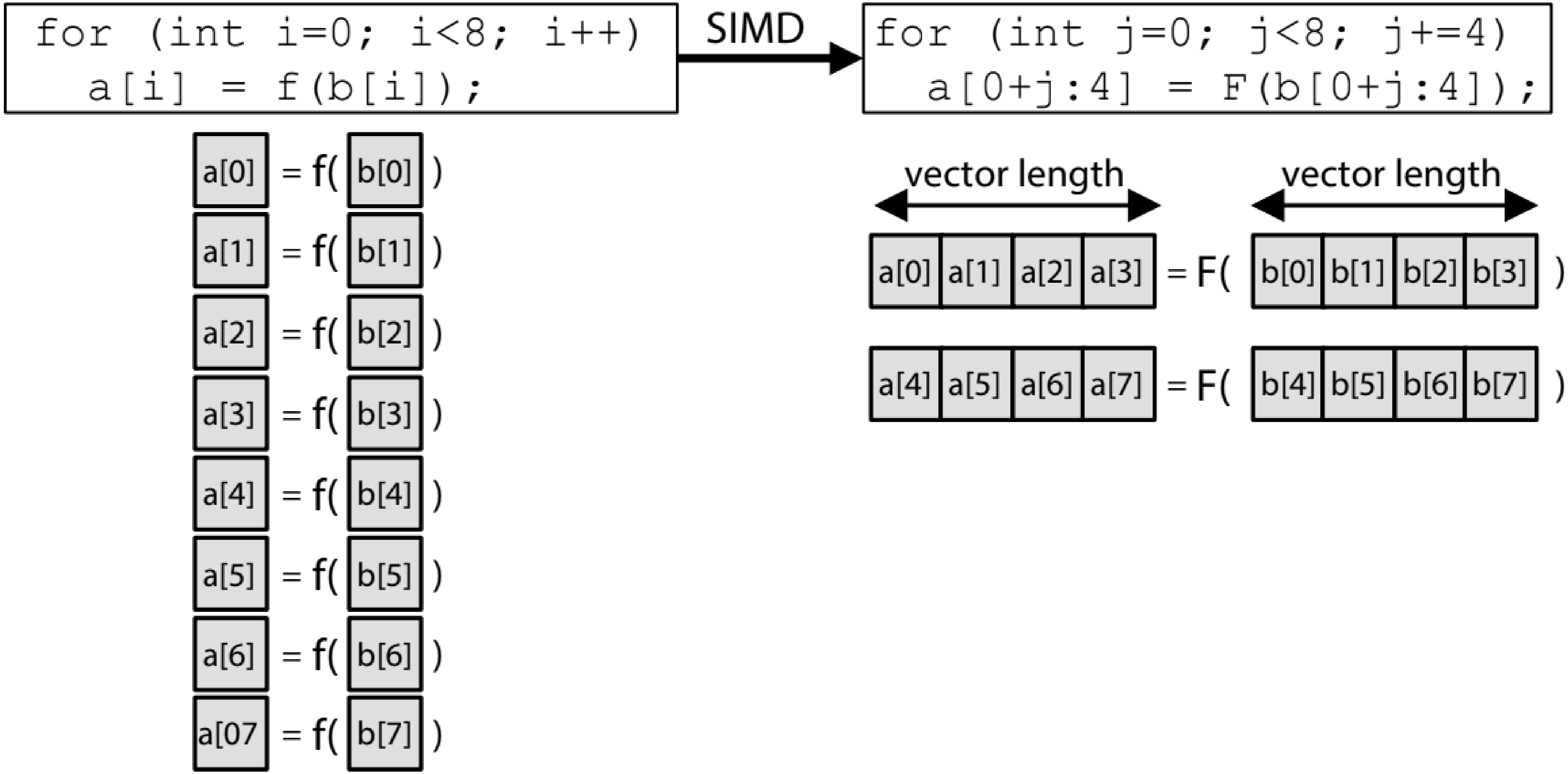


Illustration of function calls with SIMD – The scalar function f() is modified and renamed to F() in this example. This function supports vector input arguments and returns an entire vector.



```
for (int i=0; i<8; i++)
  a[i] = f(b[i]);
```

```
for (int j=0; j<8; j+=4)
  a[0+j:4] = F(b[0+j:4]);
```

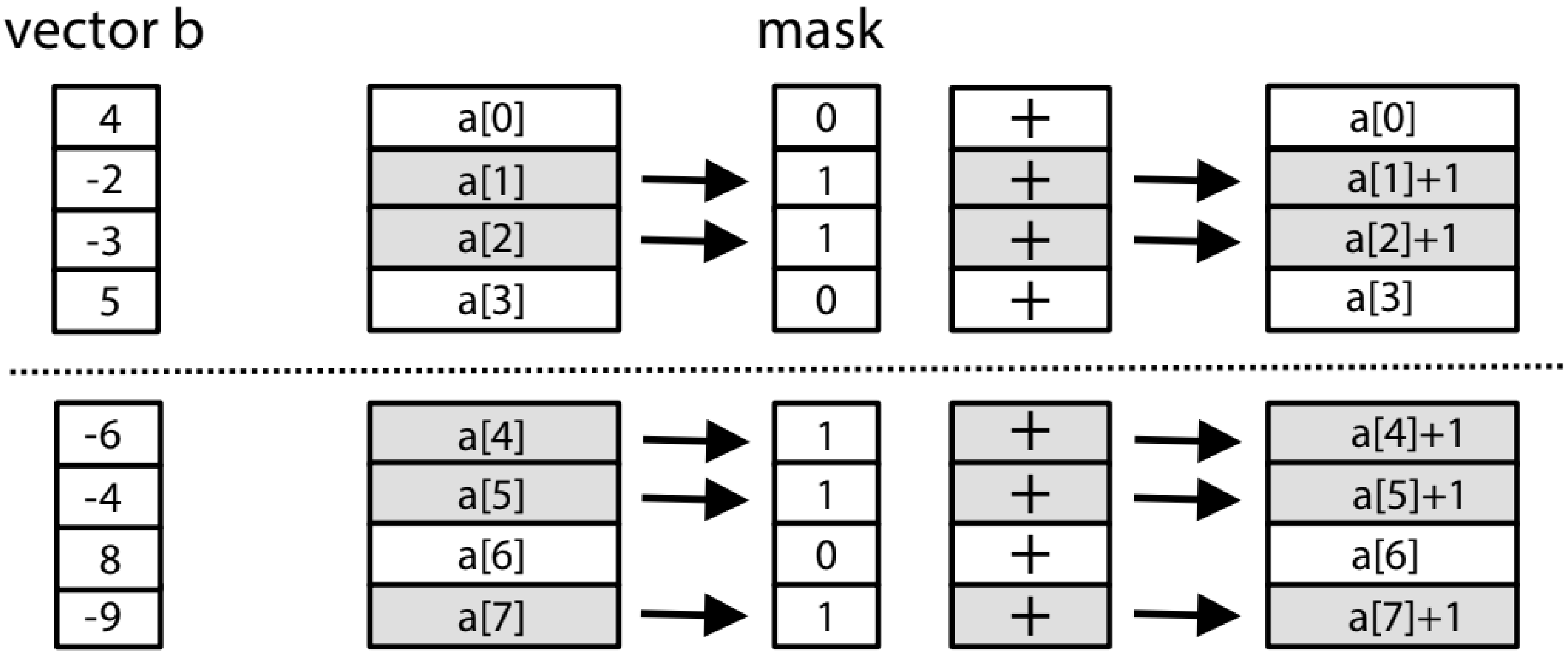
a[0] = f(b[0])
a[1] = f(b[1])
a[2] = f(b[2])
a[3] = f(b[3])
a[4] = f(b[4])
a[5] = f(b[5])
a[6] = f(b[6])
a[07 = f(b[7])

a[0] a[1] a[2] a[3] = F(b[0] b[1] b[2] b[3])

a[4] a[5] a[6] a[7] = F(b[4] b[5] b[6] b[7])

Conditional control flow converted to masked vector instructions – A vector mask predicate is used to enable or disable an operation on a given vector element. If the indexed mask is 1, the operation occurs. When it is 0, the operation is masked off.

```
     for (int i=0; i<8; i++)
          if (b[i] < 0) a[i] += 1;
```



vector b		mask		
4	a[0]	0	+	a[0]
-2	a[1]	1	+	a[1]+1
-3	a[2]	1	+	a[2]+1
5	a[3]	0	+	a[3]
-6	a[4]	1	+	a[4]+1
-4	a[5]	1	+	a[5]+1
8	a[6]	0	+	a[6]
-9	a[7]	1	+	a[7]+1