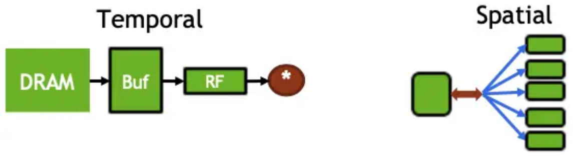
Dataflow



- **Temporal reuse:** the same data is used more than once over time by the same consumer.
- **Spatial reuse:** the same data is used by more than one consumer at different spatial locations of the hardware.

Output-stationary

- **input:** N, C_{in}, H_{in}, W_{in}
- **kernel:** C_{out}, C_{in}, H_{k}, W_{k}
- **output:** N, C_{out}, H_{out}, W_{out}

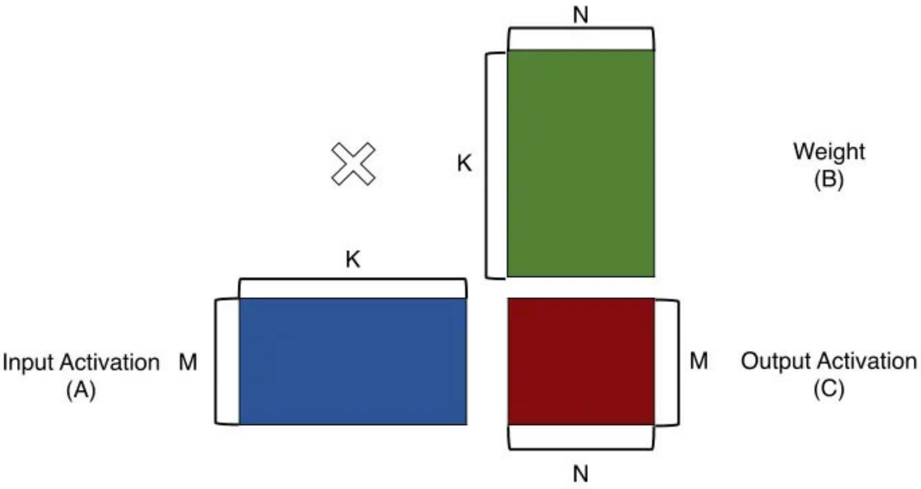
```
# padding=0, stride=1
for n in range(batch_size):
    for c_out in range(C_out):
        for h_out in range(H_out):
            for w_out in range(W_out):
                # per output element
                output[n,c_out,h_out,w_out] = 0
                for h_k in range(H_k):
                    for w_k in range(W_k):
                        for c_in in range(C_in):
                            output[n,c_out,h_out,w_out] += input[n,c_in,h_out+h_k,w_out+w_k]*
                                                          kernel[c_out,c_in,h_k,w_k]
```

Weight-stationary

```
for n in range(batch_size):
  for h_k in range(H_k):
    for w_k in range(W_k):
      for c_in in range(C_in):
        for c_out in range(C_out):
          curr_k = kernel[c_out, c_in, h_k, w_k]
          for h_out in range(H_out):
            for w_out in range(H_out):
              output[n,c_out,h_out,w_out] += input[n,c_in,h_out+h_k,w_out+w_k]*curr_k
```

Datapath Optimization

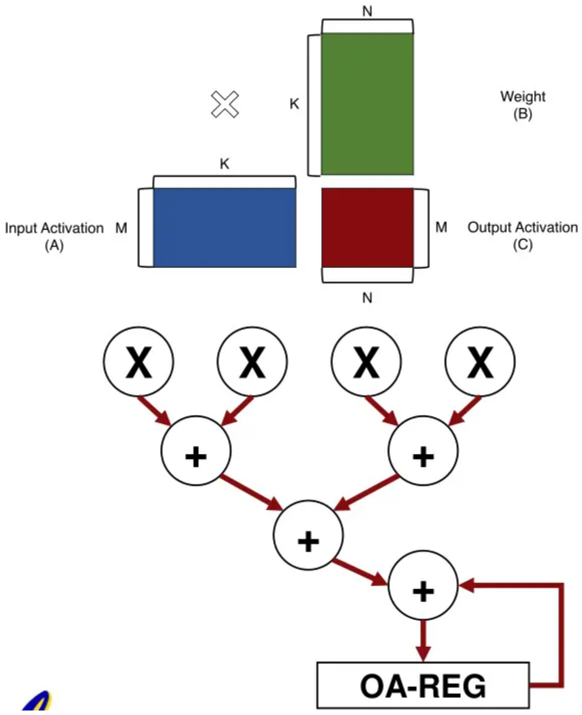
Matrix Multiply



```
for m in range(M):
  for n in range(N):
    OA[m,n] = 0
    for k in range(K):
      OA[m,n] += IA[m,k] * W[k,n]
    OA[m,n] = Activation(OA[m,n])
```

Spatial, K

```
for m in range(M):
  for n in range(N):
    OA[m,n] = 0
    spatial_for k in range(K):
      OA[m,n] += IA[m,k] * W[k,n]
    OA[m,n] = Activation(OA[m,n])
```

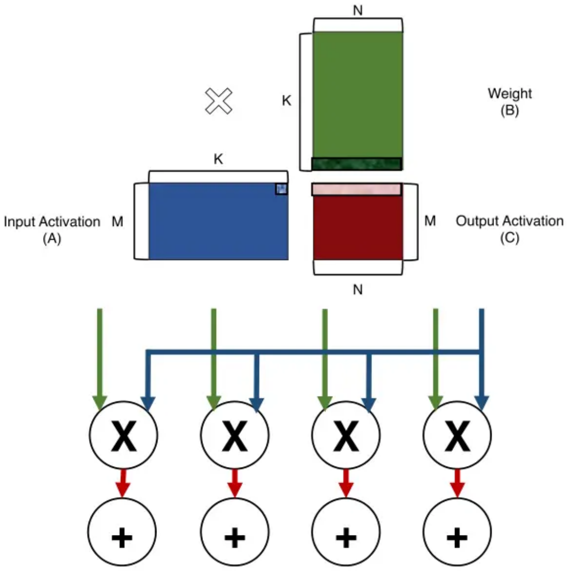


- Adder Tree

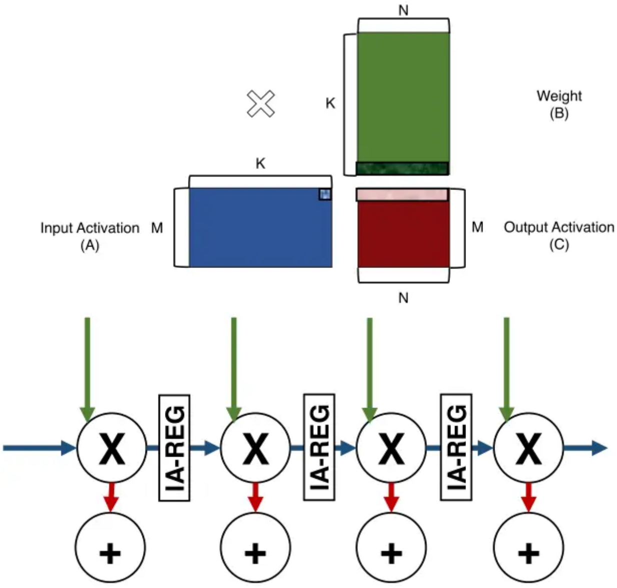
- Example: NVDLA, DianNao

Spatial, N

```
for m in range(M):
  OA[m,:] = 0
  for k in range(K):
    OA[m,:]+= IA[m,k] * W[k,:]
```

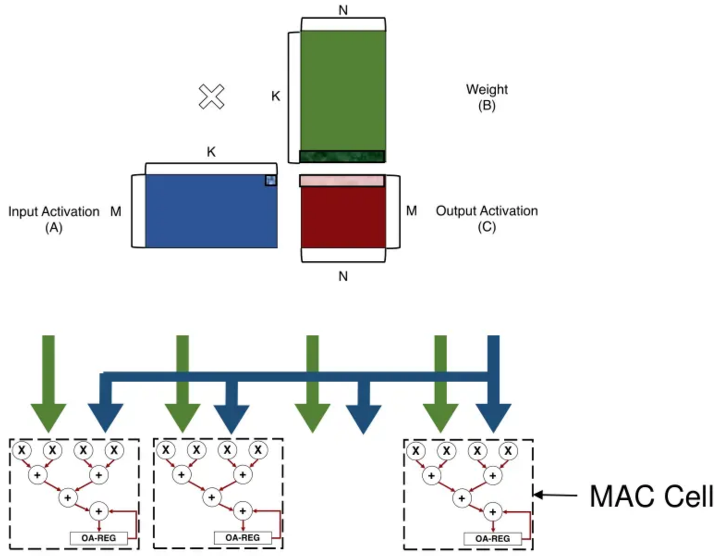


- Direct-wiring multicast
- Example: NVDLA, DianNao



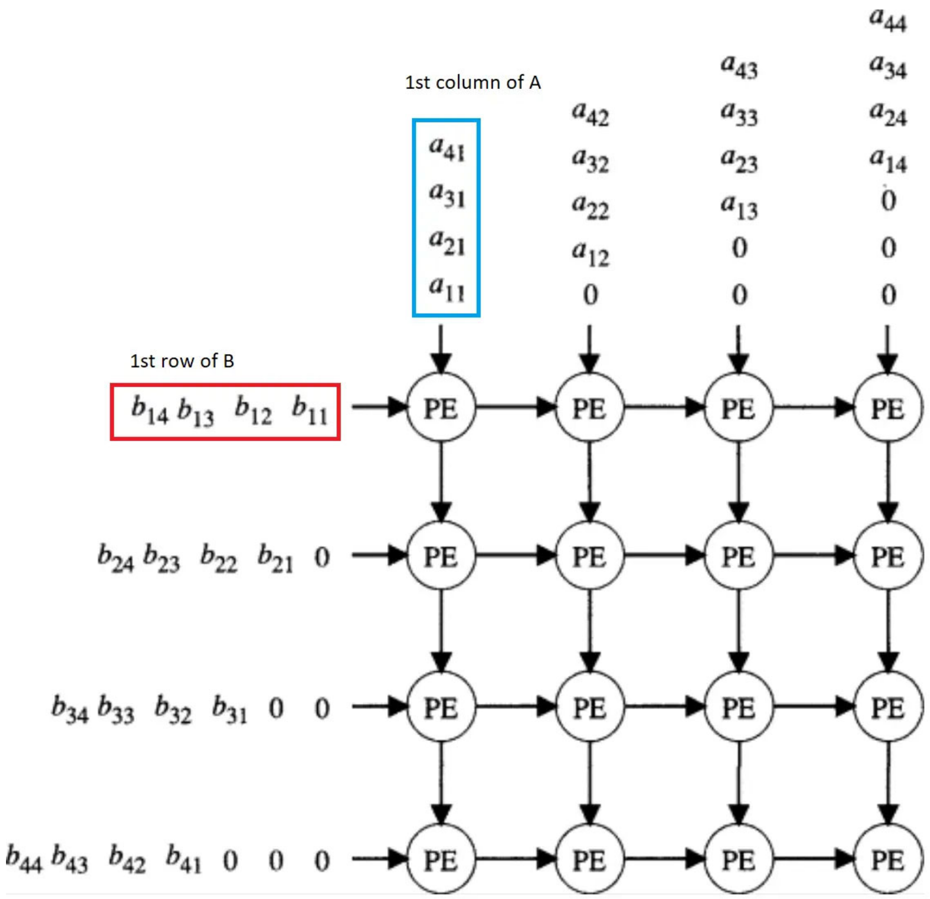
- Systolic multicast
- Example: TPU, Gemmini

Combined : NVDLA



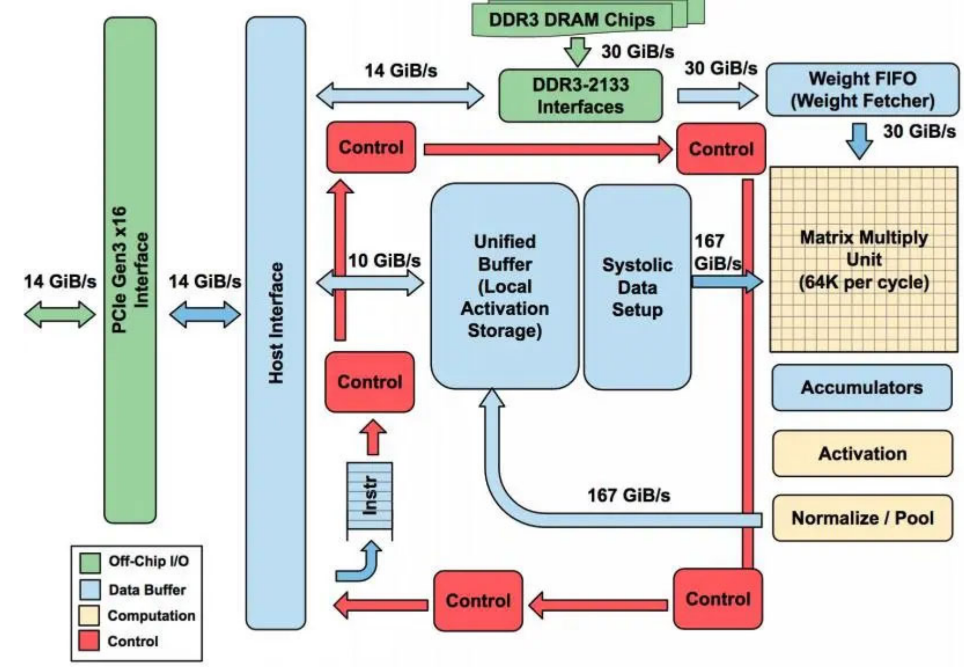
Systolic Array

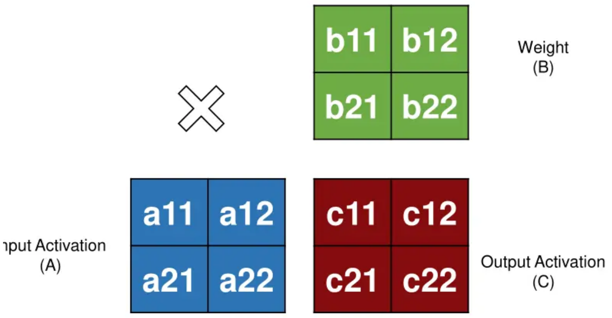
Output-stationary

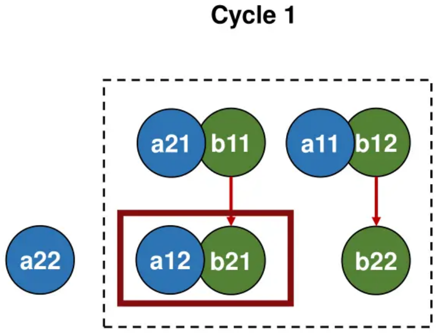

1st column of A
a_{41} a_{31} a_{21} a_{11}
a_{42} a_{32} a_{22} a_{12} 0
a_{43} a_{33} a_{23} a_{13} 0 0
a_{44} a_{34} a_{24} a_{14} 0 0 0
1st row of B
b_{14} b_{13} b_{12} b_{11}
b_{24} b_{23} b_{22} b_{21} 0
b_{34} b_{33} b_{32} b_{31} 0 0
b_{44} b_{43} b_{42} b_{41} 0 0 0
PE

- 每个PE计算4次乘法, 4次累加
- 10 个周期完成计算

Weight-stationary: TPU







$$A \times B \rightarrow C$$

- 脉动矩阵的第 j 列固定 B 的第 j 列，计算 C 的第 j 列
- C 中的每个值从脉动矩阵第一行落下
- 4个周期完成计算

https://inst.eecs.berkeley.edu/~ee290-2/sp21/