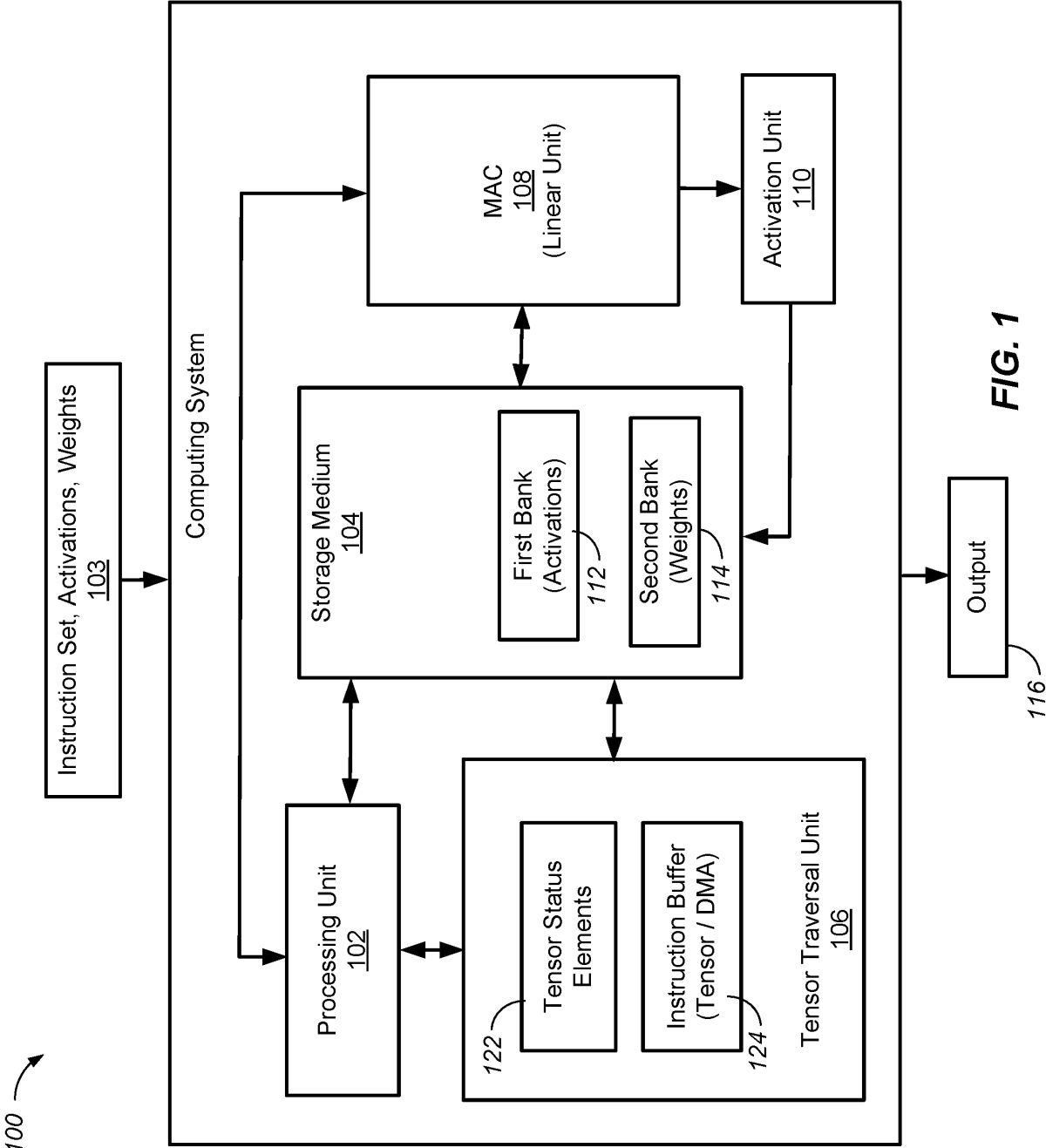
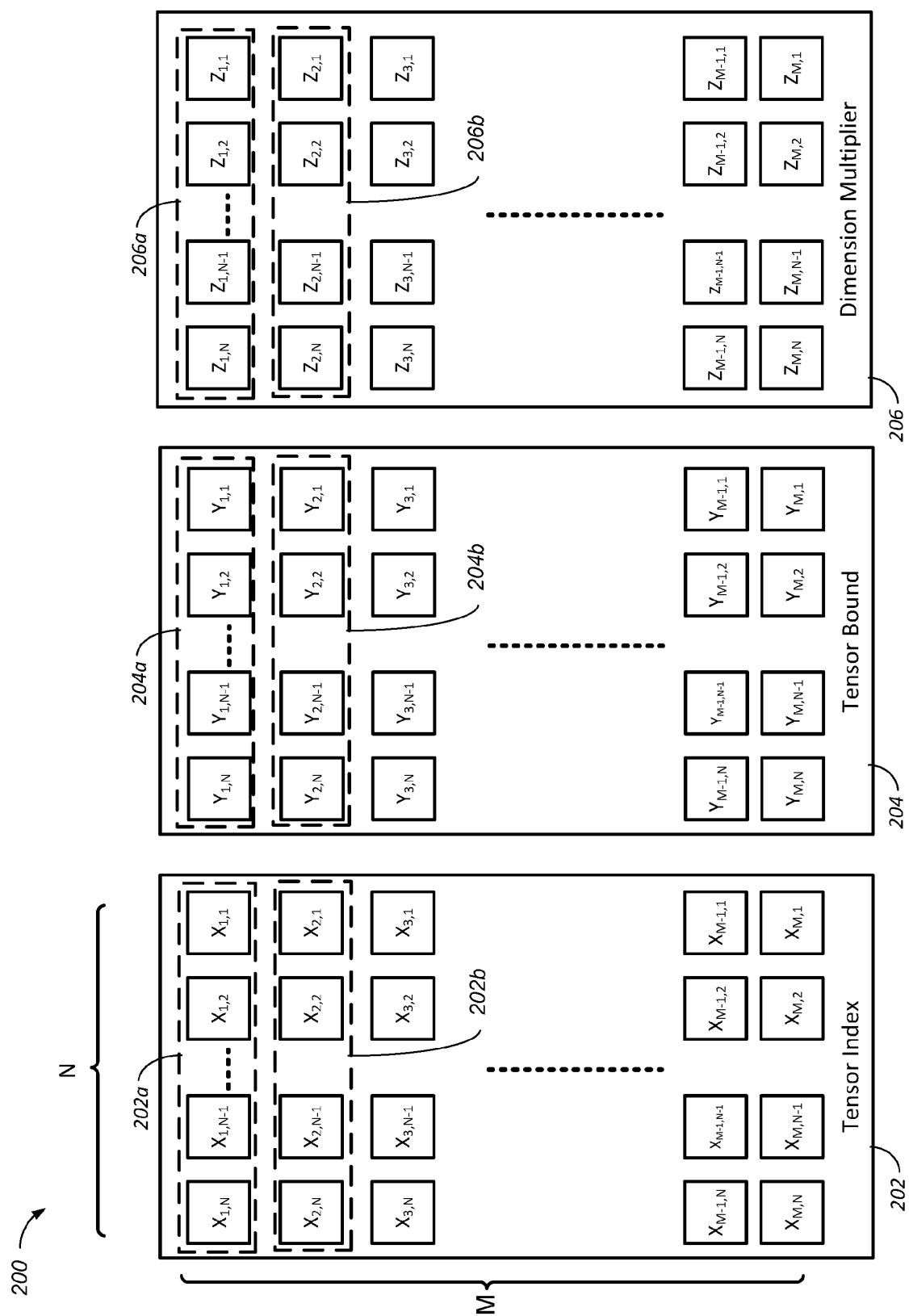






300

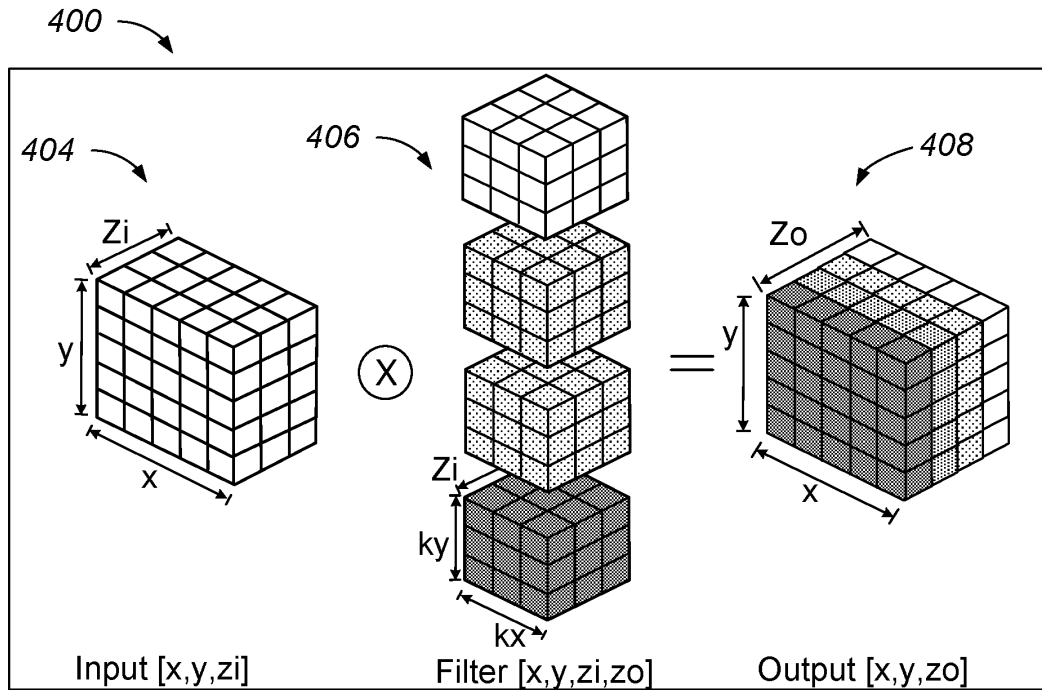
Operation Code (Opcode)	Operation Type	Operation Description
0	TensorOp	Computations at the convolution and/or FC layers
1	TensorOp	Computations at the max pooling layers
2	TensorOp	Computations at the average pooling layers
3	TensorOp	Computations at the depth-wise convolution and/or element-wise multiply layers
4	DMAOp.In	Compute tile (system 100) receives input activations and other data from instruction set 103 (via an example data bus) and place data in first bank 112
5	DMAOp.Out	Compute tile reads data from first bank 112 and writes the data to an example data bus to output the data from system 100
6	DMAOp.W-N	Compute tile moves data between the first bank 112 (N-mem) and the second bank 114 (W-mem). With this operation, data is moved from W-mem to N-mem
7	DMAOp.N-W	Compute tile moves data between the first bank 112 (N-mem) and the second bank 114 (W-mem). With this operation, data is moved from N-mem to W-mem
8	DMAOp.R-bus	Compute tile writes data to second bank 114 after retrieving the data from instruction set 103 (via an example data bus); or operation to read data from second bank and write the data to the bus to output the data from the compute tile
9*	DMAOp.InFeed*	Higher level controller (external to compute tile) sends activations and weights/parameter to compute tile
10*	DMAOp.OutFeed*	Higher level controller moves inference workload results from classifier tile to external I/O device
11	TileFenceOp	Operation to prevent any new instructions from being issued until pending instructions in the respective instruction buffers are retired
12	ScalarFenceOp	Operation to prevent issuance of new instructions until a threshold tilefenceop count is reached
* - Associated, in part, with operation performed by higher level controller device that is external to compute tile(s)		

302

304

306

FIG. 3

**FIG. 4A**

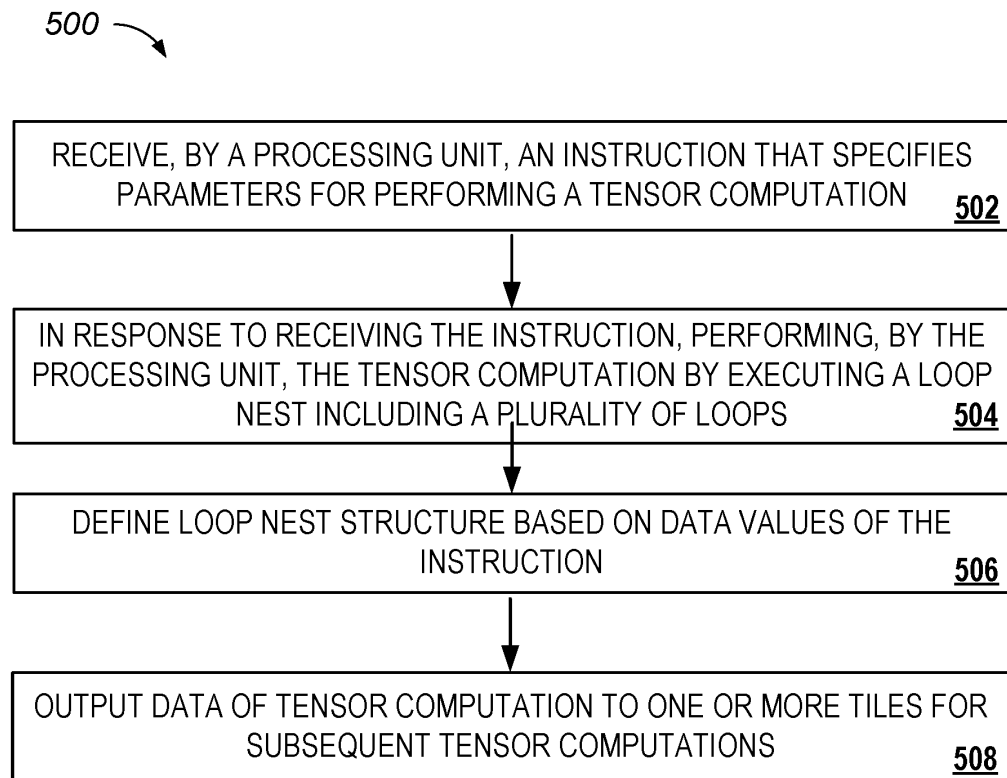
402

```

output [ : ] [ : ] [ : ] = 0
for xx in range (1, 50 , stride=10) :
    for yy in range (1, 50 , stride=10) :
        for x in range (xx, xx+10) :
            for y in range (yy, yy+10) :
                for kx in range (1, 5) :
                    for ky in range (1, 5) :
                        for zi in range (1, 3) :
                            for zo in range (1, 16) :
                                output [ x ] [ y ] [ zo ] += input [ x+kx ] [ y+ky ]

```

FIG. 4B

**FIG. 5**