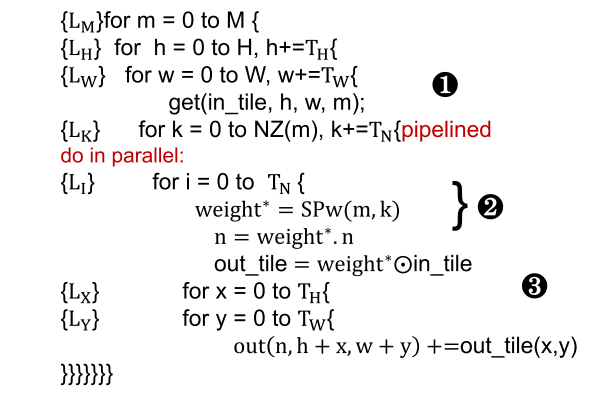
**An Efficient Hardware Accelerator for Sparse Convolutional Neural Networks on FPGAs**

1. Contributions
   1. a dataflow with element-matrix multiplication as the key operation
   2. a weight layout which can enable efficient onchip memory access.
   3. a set of architecture optimization techniques for sparse CNNs.
2. Weight-oriented dataflow
   1. gather the necessary input pixels into an input tile.

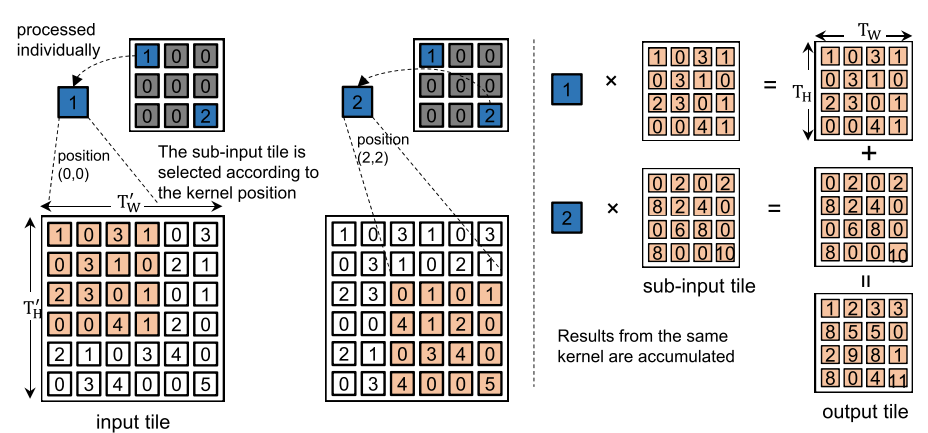


Compute size of input feature map

* 1. weights are multiplied with the input tile in parallel.
  2. the multiplication results will be accumulated the output pixels.

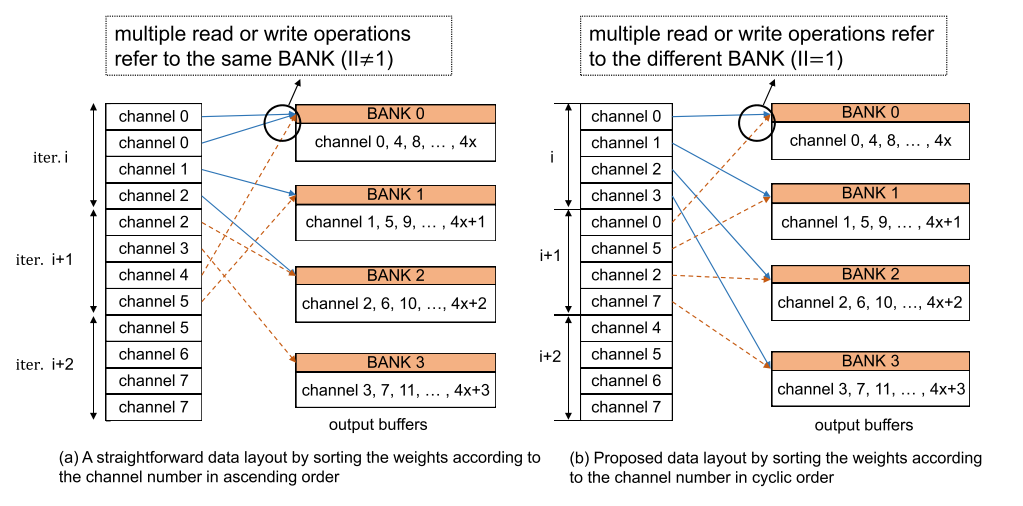


Pseudo code



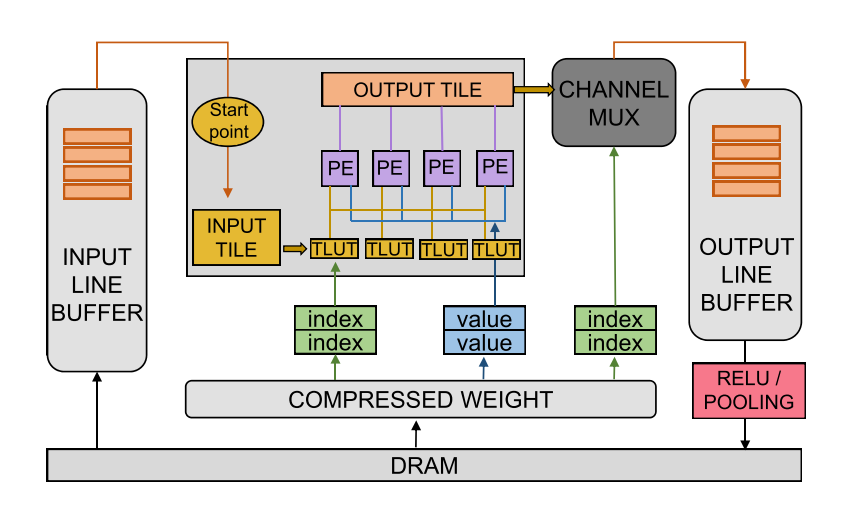
Inner computation of the dataflow

1. Weight layout
   1. compress the sparse weights into two arrays
      1. SPw array: where the nonzero weights in the same input channel are compressed into a vector(store in COO format, a 5-tuple (n, r, s, value, valid)).
      2. NZ array: which records the number of non-zero weights in each input channel.
   2. the weight layout is rearranged to cooperate with the partitioned output buffers.



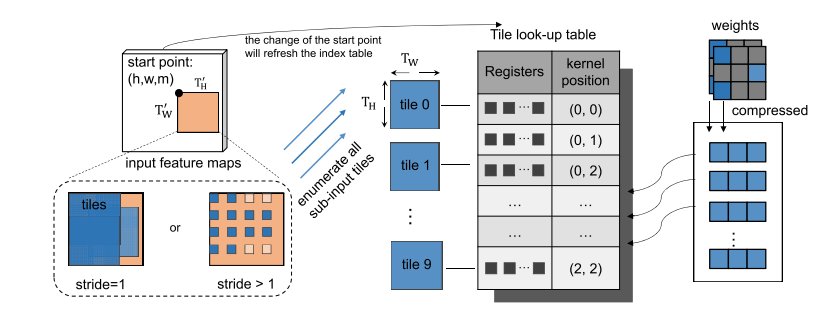
Weight layout in the output channel dimension

1. Architecture
   1. Overview



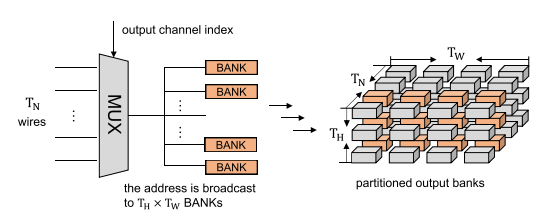
Architecture overview

* 1. TLUT module



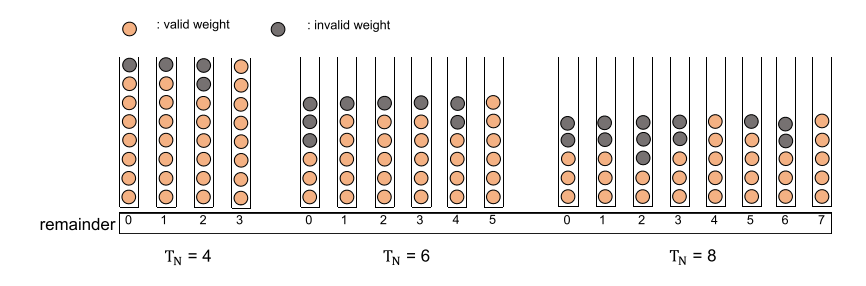
Tile look-up table to locate the sub-input tile

* 1. CMUX module



Channel multiplexer to locate the output channel address

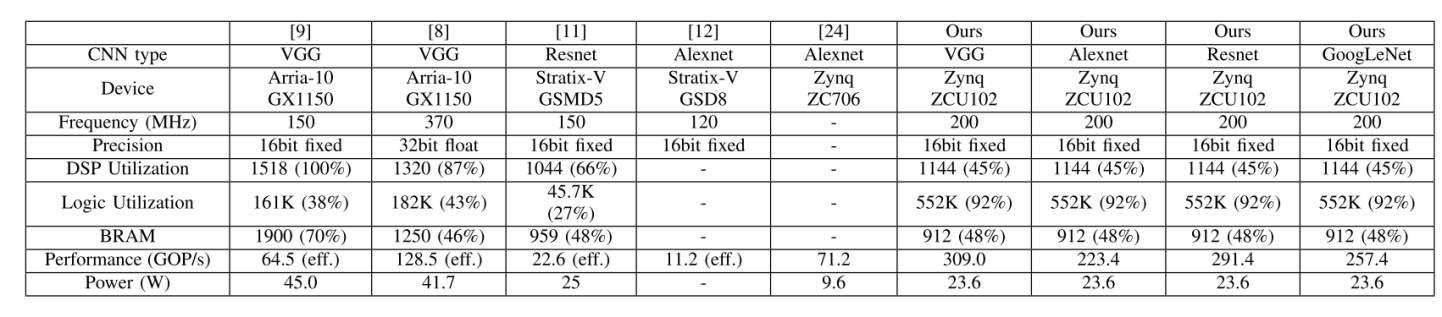
* 1. Balance the load



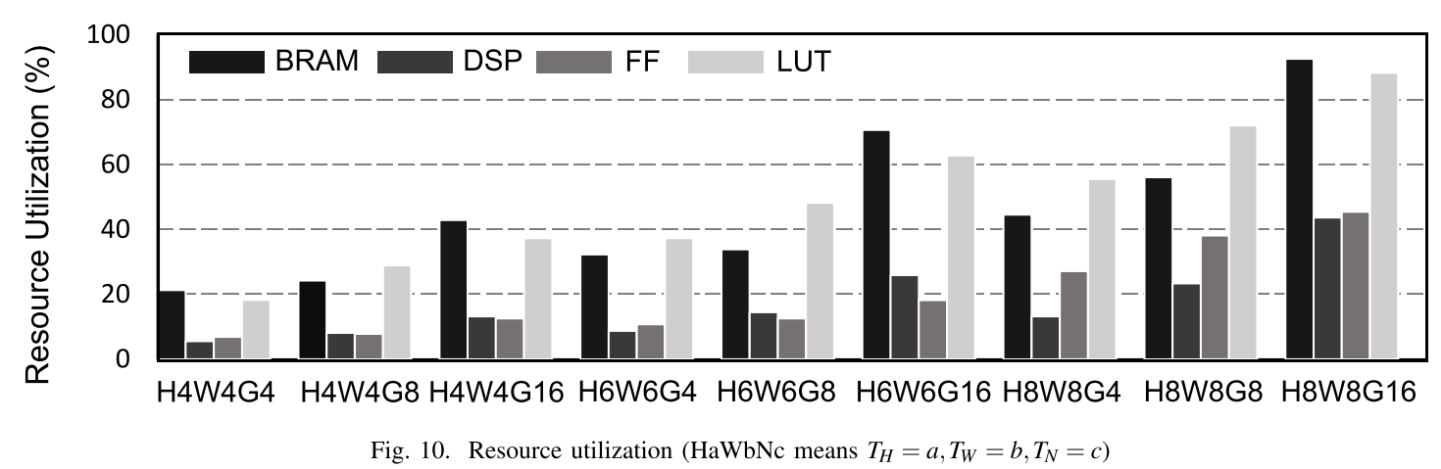
Invalid computation under proposed weight layout

* 1. Transform all the kernels to 3x3 kernel to unify the structure of the tile look-up table

1. Results



Comparison



Resource Utilization