This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <root>
    Every newly generated result will be appended after existed ones as a
    'simulation' under the 'root'. New result file will be generated if there is no
    existed one.
  – <simulation>
      All the information of a design point is collected under a 'simulation'. User
      can set 'result_print_mode' in the setting file to tune how detail the result
      file is. This documentation will take the 'concise' version as an example.
      <this_documentation_generated_time/>
      2020-08-18 10:51:23
    – <layer>
        A summary of the neural network layer under-test.
        <layer_spec/>
        Layer dimension notation: 'B' for batch size, 'K' for output channel,'C' for
        input channel, 'OX/OY' for output X/Y dimension, 'IX/IY' for input X/Y
        dimension, 'FX/FY' for filter kernel X/Y dimension, 'SX/SY' for stride on X/Y
        dimension, 'SFX/SFY' for dilated convolution on X/Y dimension.
        <total_MAC_operation/>
        Total multiply-accumulate operation of the neural network layer.
        <total_data_size_element/>
        Total number of data element of weight ('W'), input ('I'), and output ('O') in
        the neural network layer.
      </layer>
    – <hw_spec>
        Hardware specification includes two parts: 'PE array' and 'memory
        hierarchy'.
      – <PE_array>
          <precision_bit/>
          The data precision in bit used in logic computing and data storing for
          each operand (W, I, O). Note that for O, the partial sum (partial_O) and
          the final sum (fianl_O) could have different data precision requirement.
          <array_size/>
          2D dimensional PE array size (column and row).
        </PE_array>
      – <memory_hierarchy>
          <mem_name_in_the_hierarchy/>
          The overall memory hierarchy is represented as three operands' memory
          hierarchy. Data format example: <W/>['rf512', 'sram128kb',
```

'sram16Mb'] indicates that W has three memory levels to store data, from the innermost level (closest to MAC logic) to the outermost level are 'rf512', 'sram128kb', and 'sram16Mb', whose name are defined by user in the memory_pool file. Same rule is applied to operands I and O.
<**mem_size_bit**/>
Each memory module's size in bit of the memory hierarchy.
<**mem_bw_bit_per_cycle_or_mem_wordlength**/>
Memory bandwidth defines how many bit of each memory module can be accessed per cycle. It is used interchangeably with memory wordlength in the tool. Data format example: 'W': [[32.0, 32.0], [128.0, 128.0], [128, 128]] indicates the memory read and write bandwidth / wordlength at three W memory levels, from the innermost level (closest to MAC logic) to the outermost level.
<**mem_access_energy_per_word**/>
The energy of accessing one word at each memory level. Data format example: 'W': [[1.858, 2.573], [38.176, 41.952], [240, 240]] indicates the memory read and write cost per word for three W memory levels, from the innermost level (closest to MAC logic) to the outermost level.
<**mem_type**/>
Three types of memory are currently supported in the tool: single-port double-buffered (sp_db), dual-port single-buffered (dp_sb), and dual-port double-buffered (dp_db). Data format example: W': ['dp_sb', 'dp_sb', 'dp_sb'] indicates the memory type for each W memory level.
<**mem_share**/>
It defines which operands' which memory levels are physically one memory module. Data format example: {1: [('O', 'sram16Mb'), ('I', 'sram16Mb'), ('W', 'sram16Mb')]} indicates in the memory hierarchy, memory module 'sram16Mb' stores all three operands W, I, O. The memory module's name ('sram16Mb') is corresponding to the name listed in 'mem_name_in_the_hierarchy'.
<**mem_area_single_module**/>
Memory area at each memory level for each operand.
<**mem_unroll**/>
Memory unrolling factor at each memory level for each operand
</**memory_hierarchy**>
</**hw_spec**>
− <**results**>
Results including four parts: basic information, energy, performance, and area.
− <**basic_info**>
<**spatial_unrolling**/>
It is ordered from left to right the MAC level, the innermost memory level, and all the way up to the outermost memory level (the largest memory in the system). Note that it is one level more than the pure

```
          memory hierarchy. Data format example: <W/>[[], [[('OY', 13)], [('FY', 3),
          ('C', 4)]], [], []] indicates that for operand W, 'OY', 'FY', and 'C' are
          unrolled 13, 3, and 4 times at the innermost memory level, in which 'OY'
          is unrolled along one array dimension and 'FY' & 'C' are unrolled along
          another array dimension.
          <temporal_mapping/>
          It is ordered from left to right the innermost memory level, and all the
          way up to the outermost memory level. Data format example:
          <I/>[[('OX', 13), ('K', 4), ('C', 2), ('FX', 3), ('K', 6)], [('C', 12), ('K', 16)],
          [('C', 2)]] indicates that for operand I, loops ('OX', 13), ('K', 4), ('C', 2),
          ('FX', 3), ('K', 6) are scheduled at the innermost memory level with ('OX',
          13) as the innermost one; loops ('C', 12), ('K', 16) are scheduled at one
          memory level above; loop ('C', 2) is scheduled at the outermost memory
          level. In the case that no loop is scheduled at a certain memory level, an
          empty '[]' is shown.
        </basic_info>
      - <energy>
          <total_energy/>
          Total energy currently includes two parts: 'memory access energy' and
          'mac energy'.
          <mem_energy_breakdown/>
          Energy spent for each operand at each level, starting from the
          innermost memory level.
          <mac_energy/>
        </energy>
      - <performance>
          Performance includes two parts: 'mac array utilization' and 'latency'.
          <mac_array_utilization/>
          The overall MAC array utilization without considering the initial stage
          data loading.
          <latency/>
          The total latency cycles without considering the initial stage data
          loading.
        </performance>
        <area/>
        Total memory hierarchy area.
      </results>
    </simulation>
  </root>
```