Spiking Neural Network Feature Trees

This page outlines Backend Development Cost (BDC) analysis of Spiking Neural Network accelerators. The Feature Tree Template (<FT>) used for this analysis is available below. We used the provided <FT> to model the following hardware:

- Intel Loihi

Feature Tree Template (<FT>)

- **System Root**
  combine = SUM; *weight* = 1
  - **Memory Hierarchy**
    combine = SUM; *weight* = 1
    - **Implicit Data Movement**: No. of cache levels
      stage_mask = [0, 0, 1, 1]; *scale* = Linear; *weight* = 1
    - **Explicit Data Movement**: No. of scratchpad levels
      stage_mask = [1, 1, 1, 1]; *scale* = Linear; *weight* = 1
    - **Software Coherency**: No. of software coherrent levels
      stage_mask = [0, 1, 0, 1]; *scale* = Linear; *weight* = 1
    - **Storage Properties**: No. of storage properties [Activations, Weights, Output]
      stage_mask = [1, 0, 1, 0]; *scale* = Linear; *weight* = 1
  - **Node Types Set**
    combine = SUM; *weight* = 1
    - **Node**
      combine = SUM; *weight* = 1
      - **Data Movement**
        combine = SUM; *weight* = 1
        - **Granularity**: No. of movement packet sizes
          stage_mask = [1, 0, 1, 0]; *scale* = Linear; *weight* = 1
        - **Patterns**: No. of movement patterns [Peer-to-Peer, Broadcast, Scatter]
          stage_mask = [0, 1, 0, 1]; *scale* = Linear; *weight* = 1
      - **Control**
        combine = SUM; *weight* = 1
        - **Latency Hiding**: No. of latency hiding widgets [Threads, Double Buffers]
          stage_mask = [0, 0, 1, 1]; *scale* = Linear; *weight* = 1
        - **Data Dependency**: Sync. support between data producer and consumer
          stage_mask = [1, 0, 1, 0]; *scale* = Linear; *weight* = 1
      - **Datapaths Set**
        combine = SUM; *weight* = 1
        - **DataPath**
          combine = SUM; *weight* = 1
          - **Operation Dimensions**: No. of dimensions of data operations
            stage_mask = [1, 0, 0, 0]; *scale* = Linear; *weight* = 1
          - **Unmaskable Dimensions**: No. of inner operation dimensions without masking support
            stage_mask = [1, 0, 0, 0]; *scale* = Linear; *weight* = 1
          - **Memory Levels**: No. of memory units in datapath [Input, Output, Internal]
            stage_mask = [1, 0, 1, 0]; *scale* = Linear; *weight* = 1
          - **Latency Hiding**: No. of latency hiding widgets [Threads, Double Buffers]
            stage_mask = [0, 0, 1, 0]; *scale* = Linear; *weight* = 1
  - **Network and Synchronization**
    combine = SUM; *weight* = 1
    - **Latency**: No. of latency domains visible to node
      stage_mask = [0, 0, 0, 1]; *scale* = Linear; *weight* = 1

- **Bandwidth**: No. of bandwidth domains visible to node
  stage_mask = [0, 0, 0, 1]; *scale* = Linear; *weight* = 1
- **Topology Positions**: No. of topology positions in node layout
  stage_mask = [0, 0, 0, 1]; *scale* = Linear; *weight* = 1
- **Sync Capability**: No. of inter-node sync. capabilities [Atomics, Interrupt]
  stage_mask = [0, 1, 0, 1]; *scale* = Linear; *weight* = 1