1 Parameter Conventions

1.1 Images

N	Image Dimension: $N \times N$
C	Channels per Pixel
A	Word Width for each Channel (Activation Precision)

With respect to a specific convolutional layer:

- N, C, A refer to the input image, and
- N', C', A' refer to the output image.

1.2 Kernels

K	Kernel Dimension: $K \times K$
S	Stride
W	Word Width of Weights

1.3 Compute Layout

P	Parallel PEs (mutually independent)
Q	Simultaneous Products (added onto the same sum)

2 Cost Functions

2.1 Block RAM

Block RAM utilization is typically well predictable. Achieving the theoretical bounds derived below should be a key goal of the network generator.

The use of memory blocks is always subject to address space and data word fragmentation. This is reflected by the ceilings in the equations below. The actually suffered waste of resources should be monitored in practice so as to identify severe penalties and think up suitable remedies if encountered.

2.1.1 Weight Memory

Overall Content Size:	$C' \cdot K^2 \cdot C \cdot W$ bits
Parallel Access Width:	$P \cdot Q \cdot W$ bits
Implied Layout:	$\frac{C \cdot C' \cdot K^2}{P \cdot Q} \times P \cdot Q \cdot W$

Requiring just one port at each point in time allows to use RAMB18 halves with an effective layout of 512×18. This implies an ideal BRAM demand of:

$$\text{RAMB18}_{Weights} = \frac{1}{2} \times \left\lceil \frac{C \cdot C' \cdot K^2}{512 \cdot P \cdot Q} \right\rceil \times \left\lceil \frac{P \cdot Q \cdot W}{18} \right\rceil$$

Notes:

- HLS is likely to have difficulties with packing half-utilized RAM blocks together.
- HLS will typically only utilize 16/32 bits of the BRAM data ports.
- The current implementation structures the weight memory in a way that P ends up outside the ceiling.
  This is fixable and should be approached if it helps to reduce fragmentation significantly.

Effectively, the current state of affairs demands:

$$\text{RAMB18}_{Weights} = P \times \left\lceil \frac{C \cdot C' \cdot K^2}{512 \cdot P \cdot Q} \right\rceil \times \left\lceil \frac{Q \cdot W}{32} \right\rceil$$

2.1.2 Line Buffers

Line buffers are required for re-arranging and duplicating input data to be fed into the multi-matrix-vector-multiply units. The buffer maintains a history of K lines of the input image.

Overall Content Size:	$K \cdot N \cdot C \cdot A$ bits
Parallel Access Width:	$Q \cdot A$ bits
Implied Layout:	$K \cdot N \cdot \frac{C}{Q} \times Q \cdot A$

The line buffers are ideally filled and drained concurrently, which requires the underlying BRAM to be configured in SDP mode. This restricts its granularity to 512×36 and implies a BRAM demand of:

$$\text{RAMB18}_{LineBuffers} = \left\lceil \frac{K \cdot N \cdot C}{512 \cdot Q} \right\rceil \times \left\lceil \frac{Q \cdot A}{36} \right\rceil$$

Notes:

- HLS will typically only utilize 16/32 bits of the BRAM data ports.
- The current implementation works the line buffer on the granularity of the stride S, which might introduce more severe fragmentation.
  Thomas is challenging this approach and will have to prove that it is doable otherwise if significant fragmentation is observed in practice.

The current state of affairs is thus:

$$\underline{\text{RAMB18}_{LineBuffers} = \left(\frac{K}{S} + 1\right) \cdot \left\lceil \frac{S \cdot N \cdot C}{512 \cdot Q} \right\rceil \times \left\lceil \frac{Q \cdot A}{32} \right\rceil}$$

2.2 Logic

The logic utilization of HLS synthesis results is estimated by models that are tuned and validated by experimental heuristic data.

2.2.1 PE

The logic consumption of a PE can be modeled in terms of its constituents:

Q-element Dot Product	linear growth in $Q \cdot W \cdot A$
Accumulation	linear growth in accumulator width, which is something like $W + A + log_2C + 2 \cdot log_2K$
Counters	various progress counters and associated comparators, which grow logarithmically with the high-level problem complexity, i.e. with $log_2N + log_2C + log_2C'$

This already yields an initial model with the parameters α, β, γ:

$$\text{LUT}_{PE} = \alpha\left(Q \cdot W \cdot A\right) + \beta\left(W + A + log_2C + 2 \cdot log_2K\right) + \gamma\left(log_2N + log_2C + log_2C'\right)$$

Note: The accuracy of the predicted composed cost may suffer from the massive inlining performed within the HLS code base, which causes the boundaries of the constituents to blur.

2.2.2 Sliding Window Generator

2.2.3 Stream Width Adaptation

2.2.4 Host Interface and Global Control