Implementation of the Jet Algorithm

Sam Silverstein

Anders Ferm

Torbjörn Söderström

We are rewriting the Jet algorithm from scratch

- Full-size algorithm (11x7)
- Produce portable VHDL code
- Take advantage of modern design tools (FPGA Advantage)
- Chance to examine and reevaluate algorithm elements and organisation

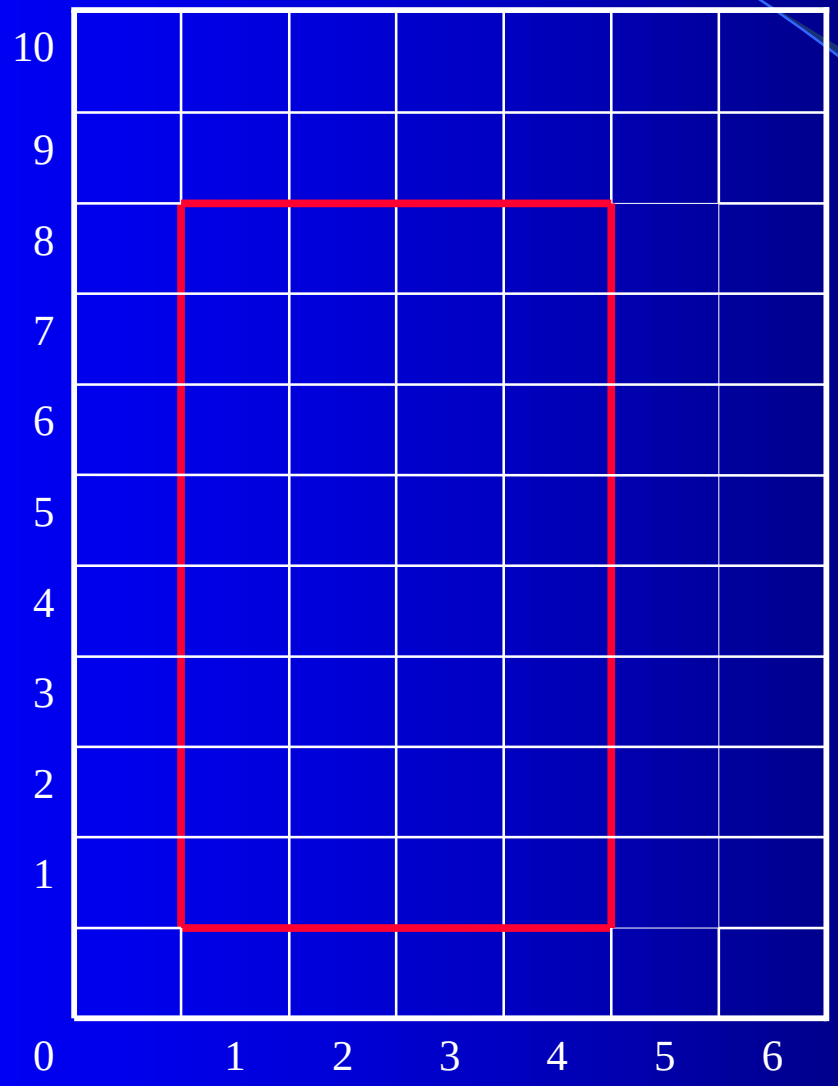
Strategy: Compact Design

- Design logic blocks that use as few resources as possible
  - 5-bit serial vs. 10-bit parallel
- Eliminate duplicated operations
  - Adder trees
  - Nearest neighbor comparison

Some examples from the algorithm

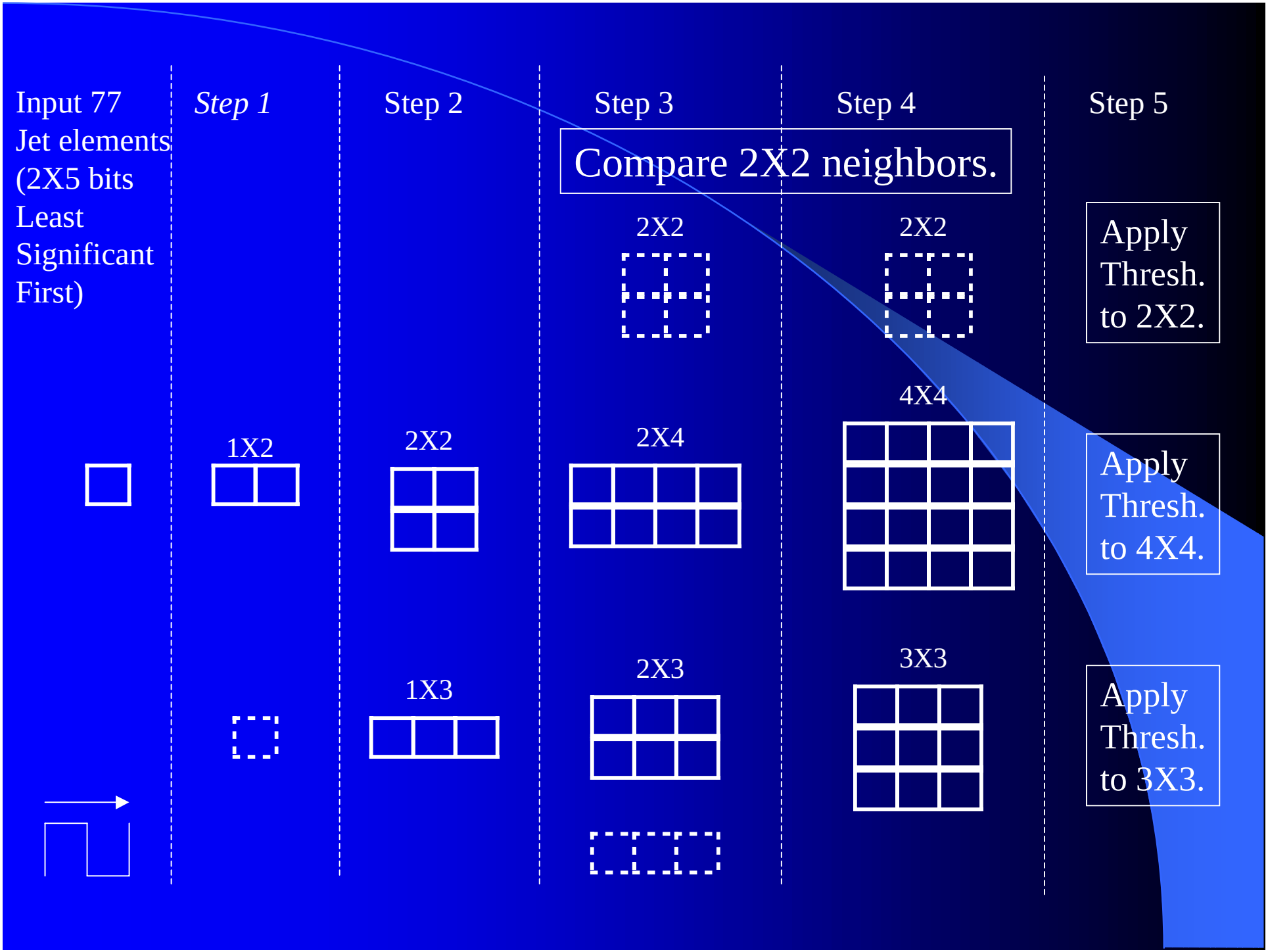
- The adder trees
- Local maximum identification

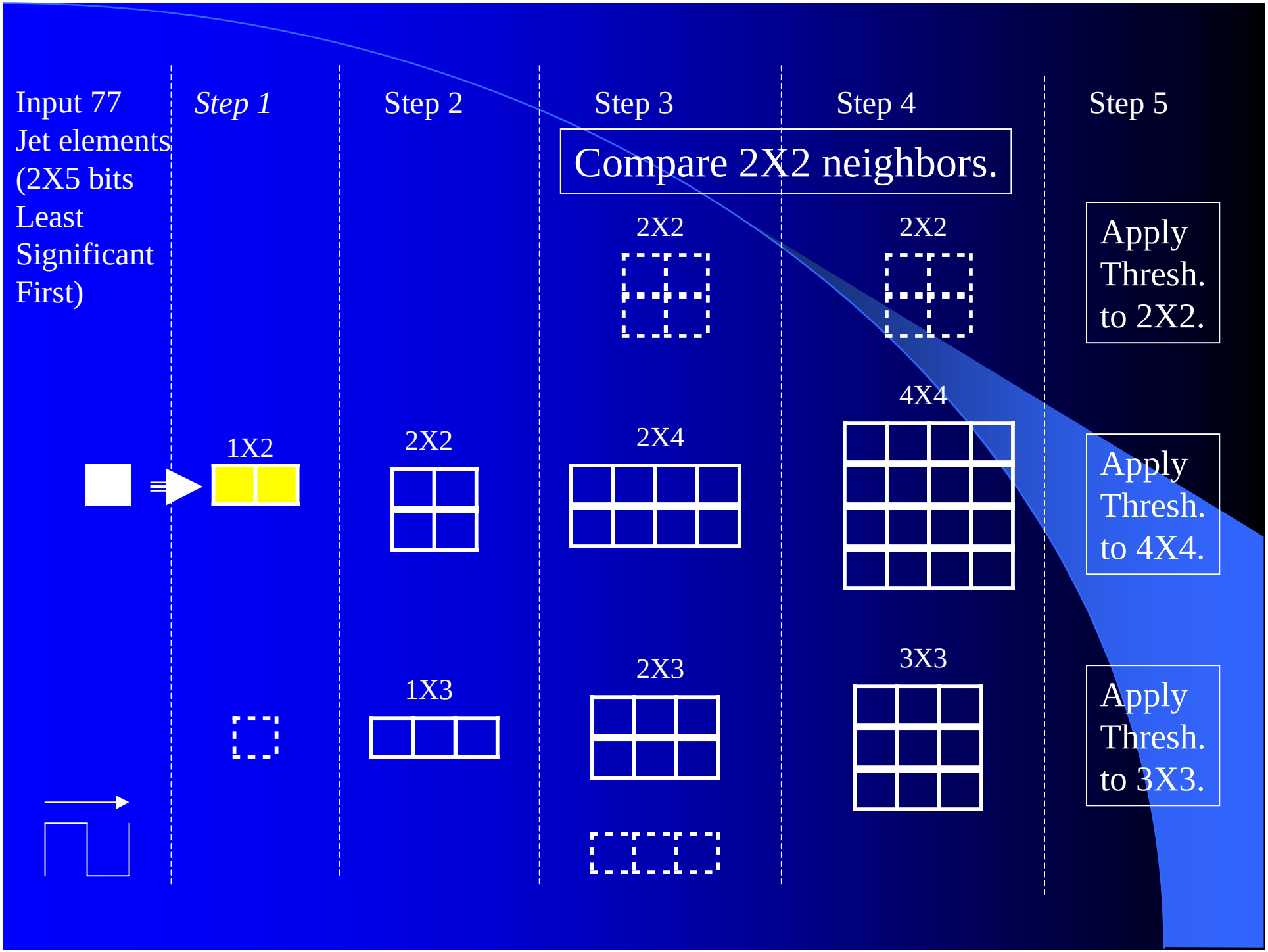
The Jet adder trees



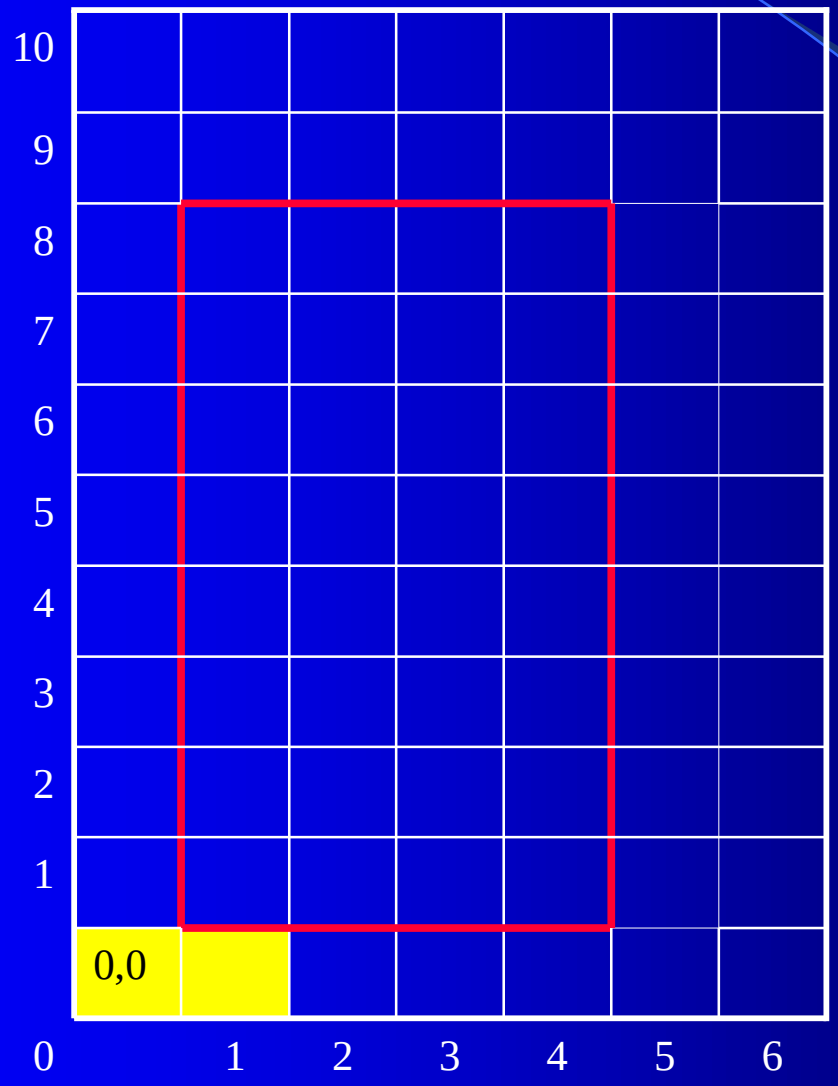
77 Jet elements

32 Core elements

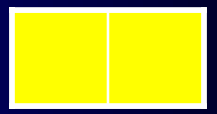

Input 77 Jet elements (2X5 bits Least Significant First)
Step 1
Step 2
Step 3
Step 4
Step 5
Compare 2X2 neighbors.
2X2
2X2
Apply Thresh. to 2X2.
4X4
1X2
2X2
2X4
Apply Thresh. to 4X4.
3X3
2X3
1X3
Apply Thresh. to 3X3.


Input 77 Jet elements (2X5 bits Least Significant First)
Step 1
Step 2
Step 3
Step 4
Step 5
Compare 2X2 neighbors.
2X2
2X2
Apply Thresh. to 2X2.
4X4
1X2
2X2
2X4
Apply Thresh. to 4X4.
1X3
2X3
3X3
Apply Thresh. to 3X3.

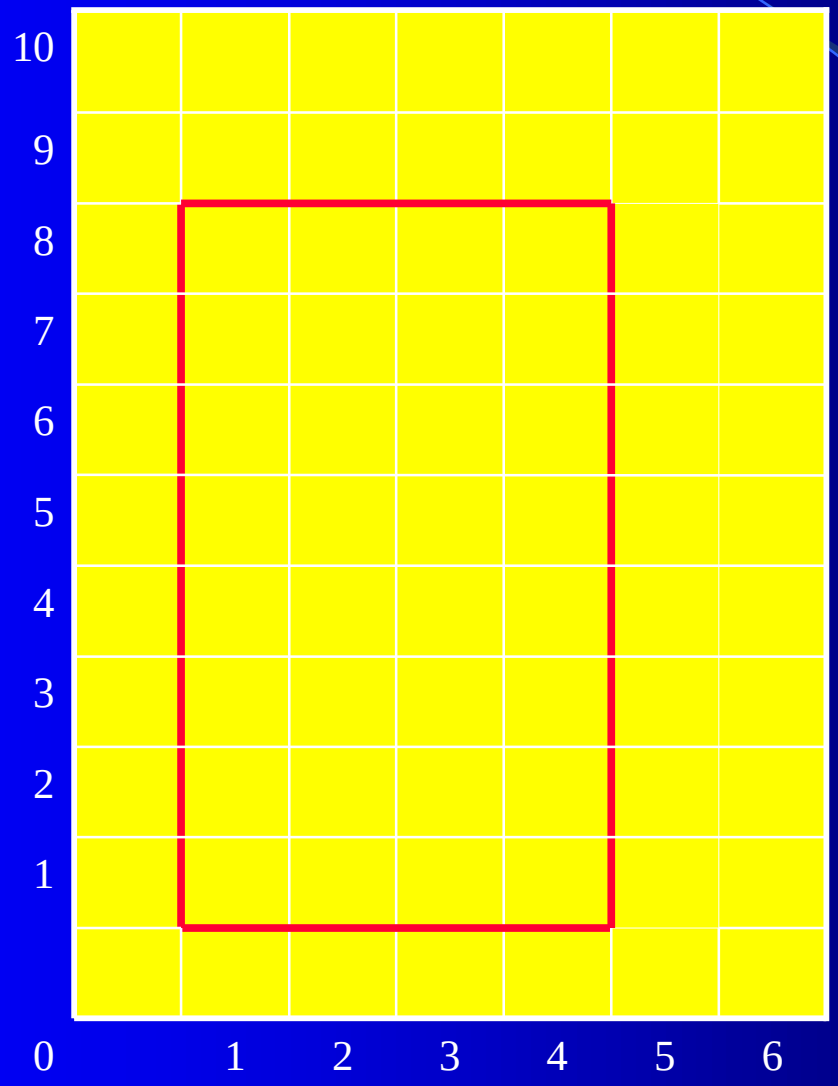
1x2 Jet element Sum (Step 1)



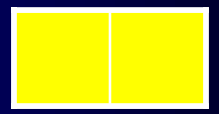
Total 1 summation

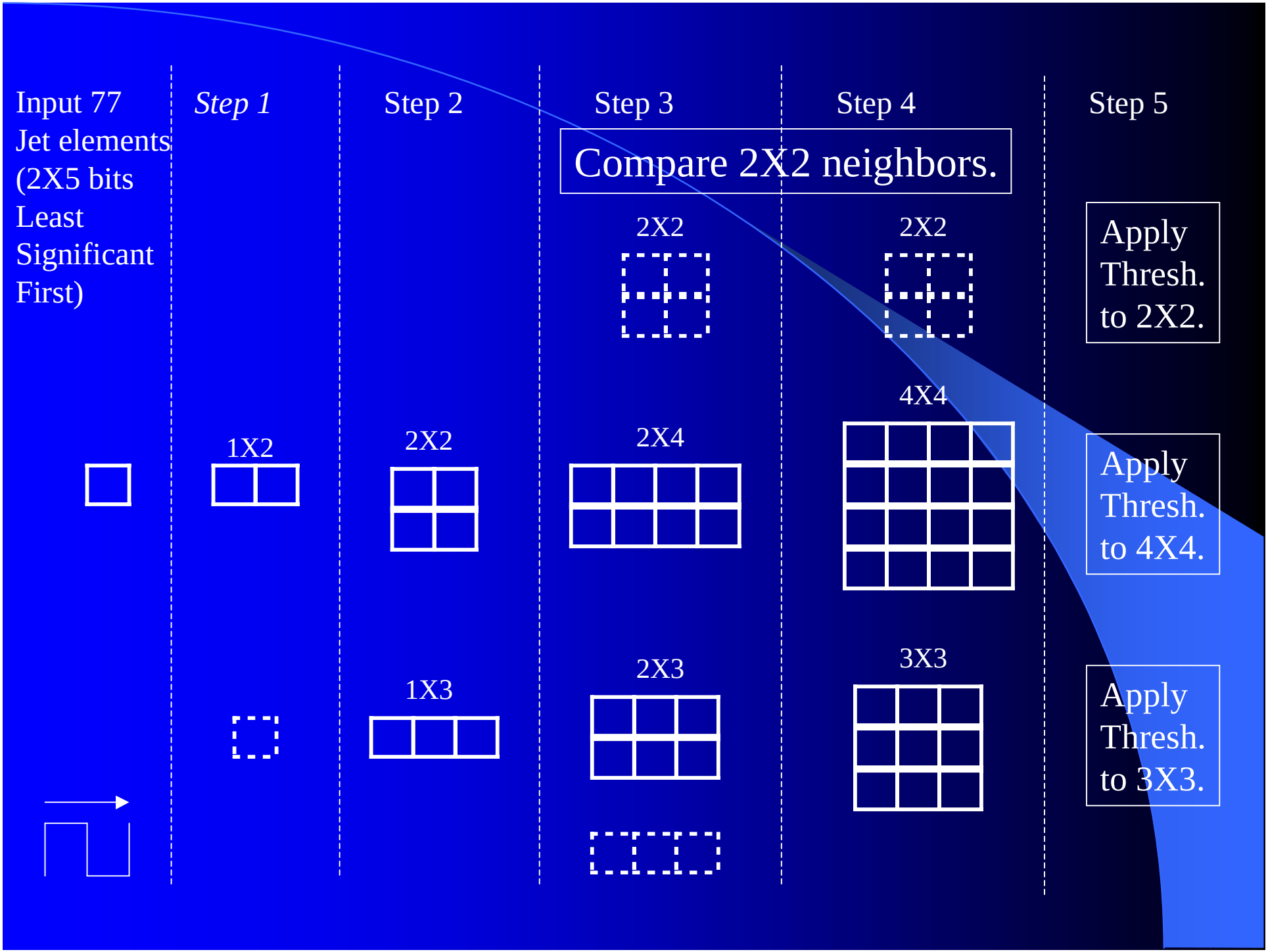


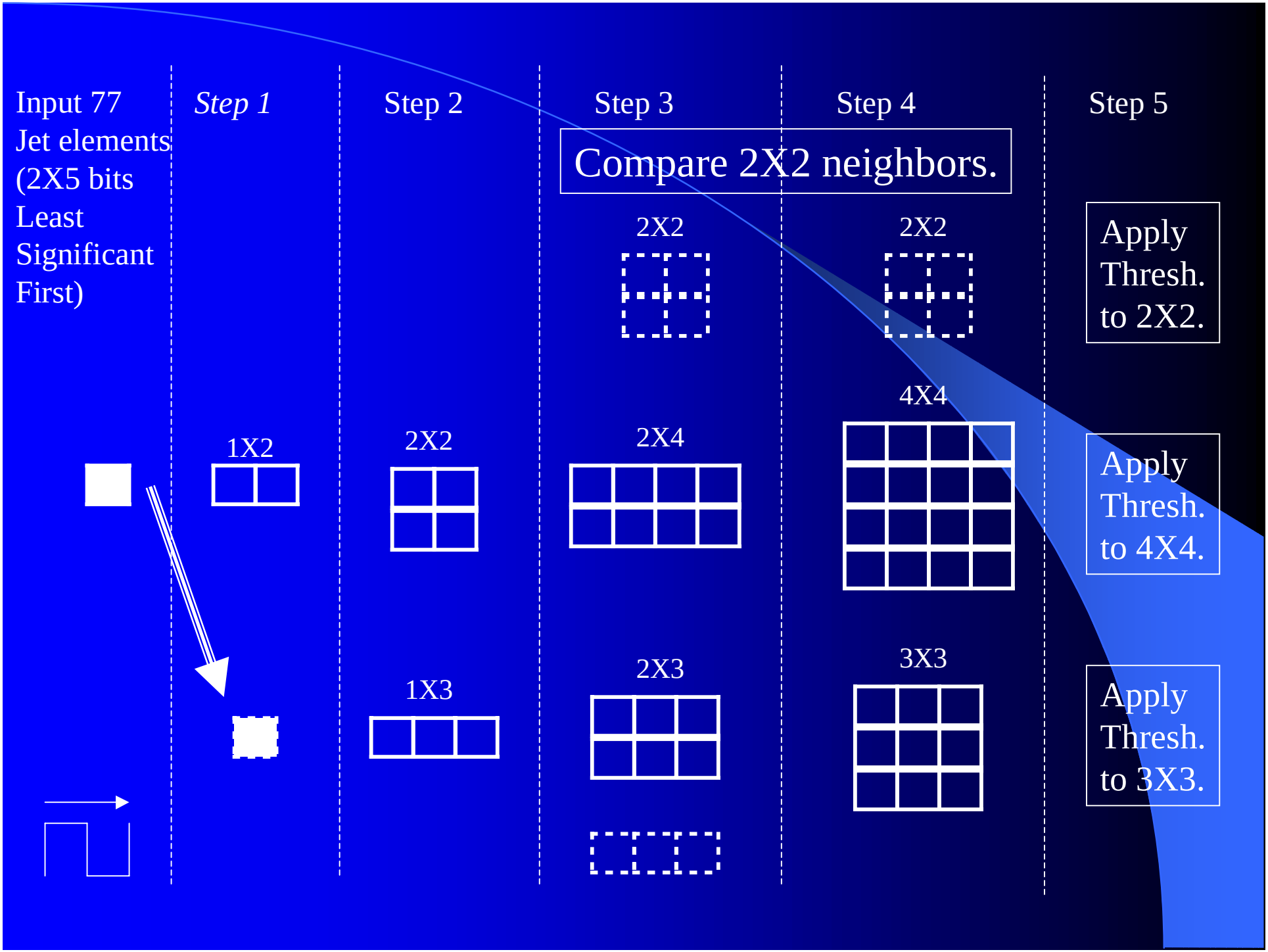
1x2 Jet element Sum (Step 1)



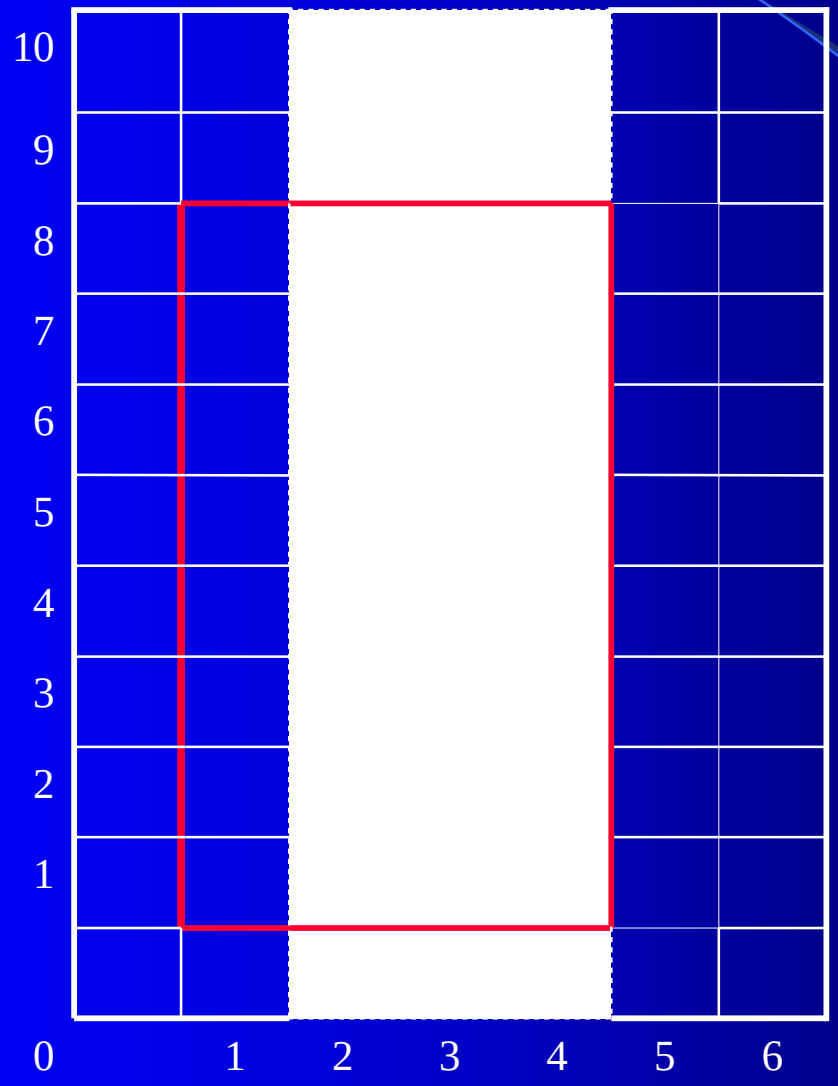
Total 11x6 summations



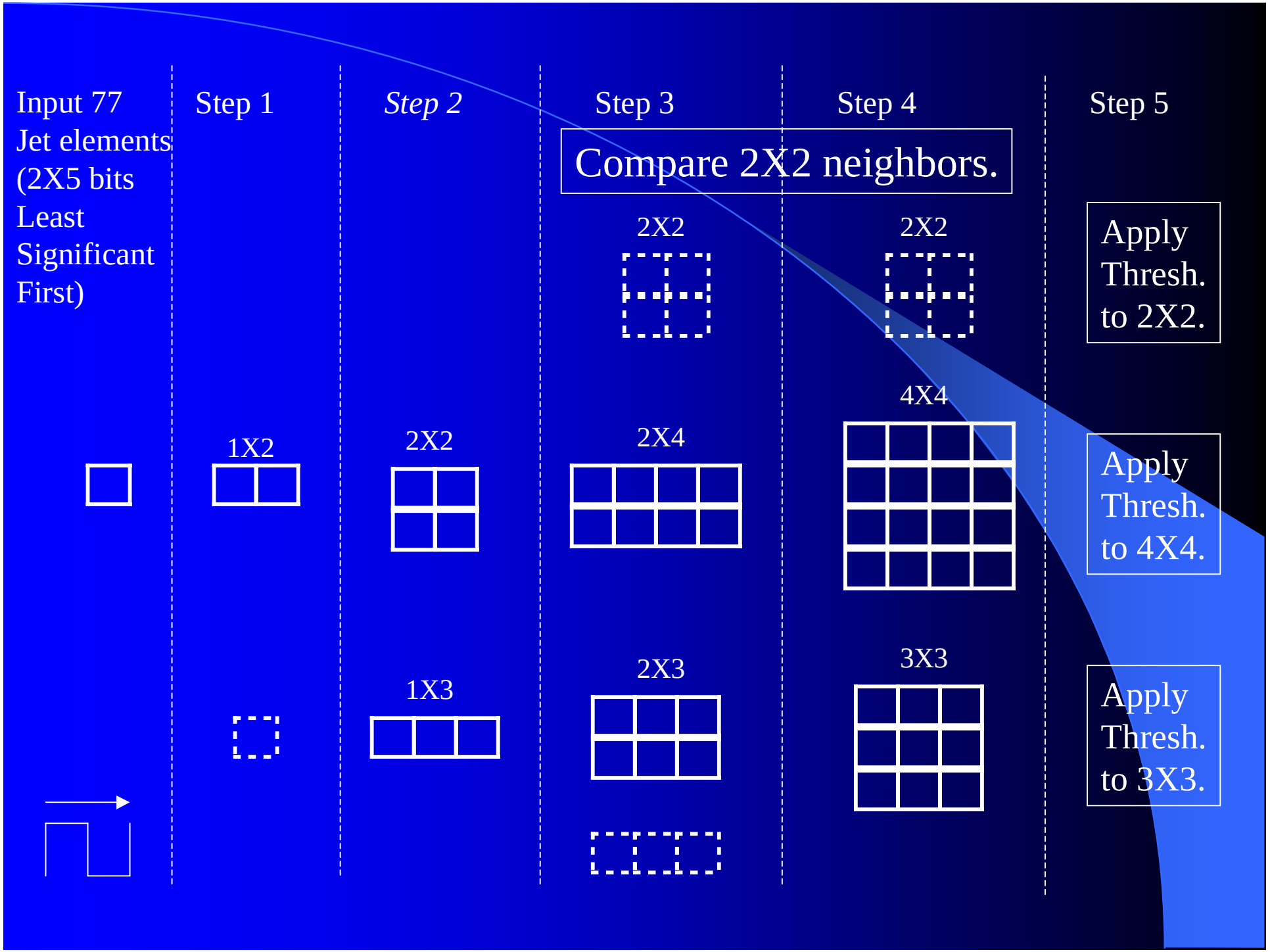

Input 77 Jet elements (2X5 bits Least Significant First)
Step 1
Step 2
Step 3
Step 4
Step 5
Compare 2X2 neighbors.
2X2
2X2
Apply Thresh. to 2X2.
4X4
1X2
2X2
2X4
Apply Thresh. to 4X4.
3X3
2X3
1X3
Apply Thresh. to 3X3.

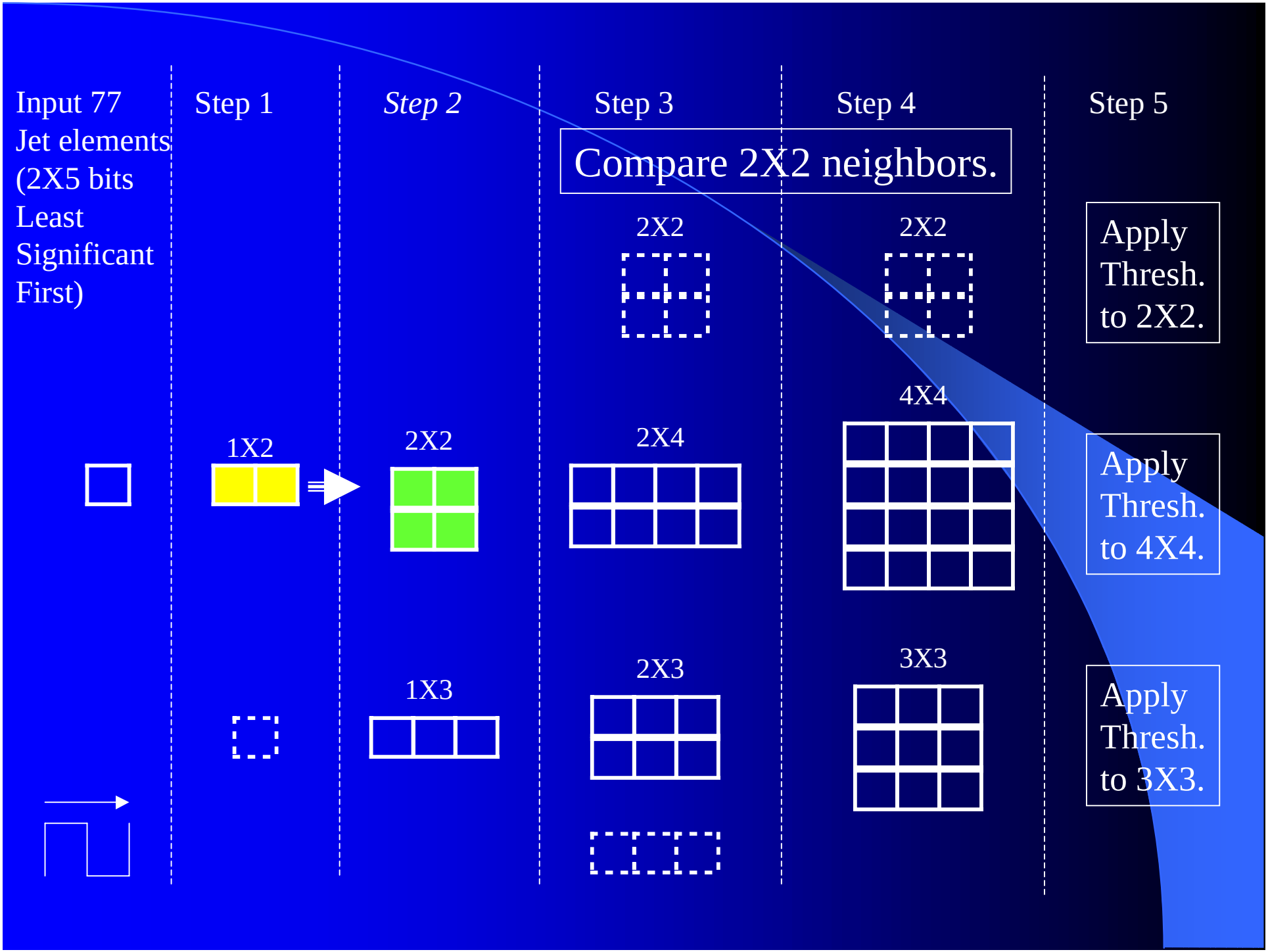

Input 77 Jet elements (2X5 bits Least Significant First)
Step 1
Step 2
Step 3
Step 4
Step 5
Compare 2X2 neighbors.
2X2
2X2
Apply Thresh. to 2X2.
4X4
1X2
2X2
2X4
Apply Thresh. to 4X4.
2X3
3X3
1X3
Apply Thresh. to 3X3.

1x1 Jet element Save (Step 1)

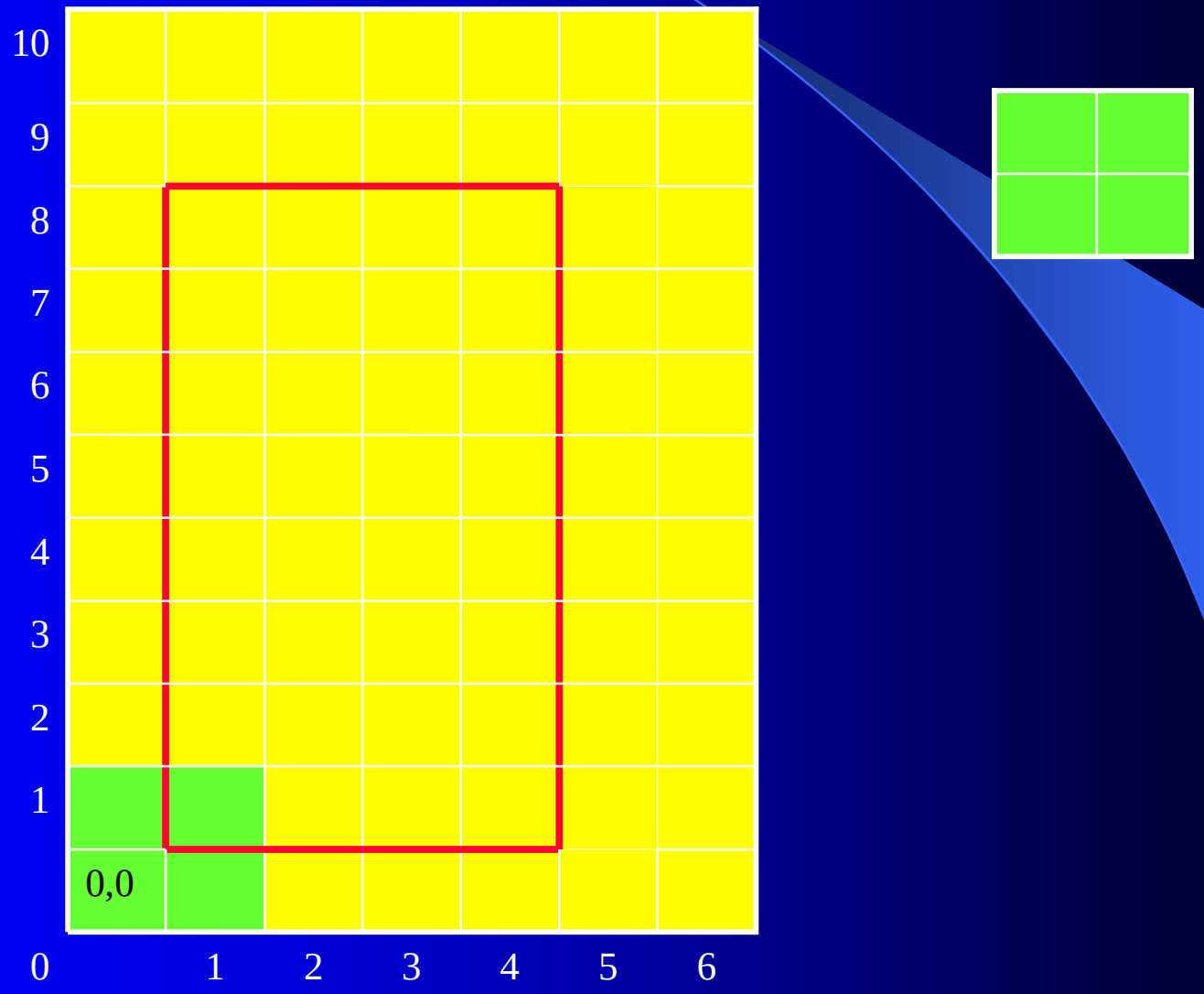


Save 30 Jet elements
for the 1X3 summations.

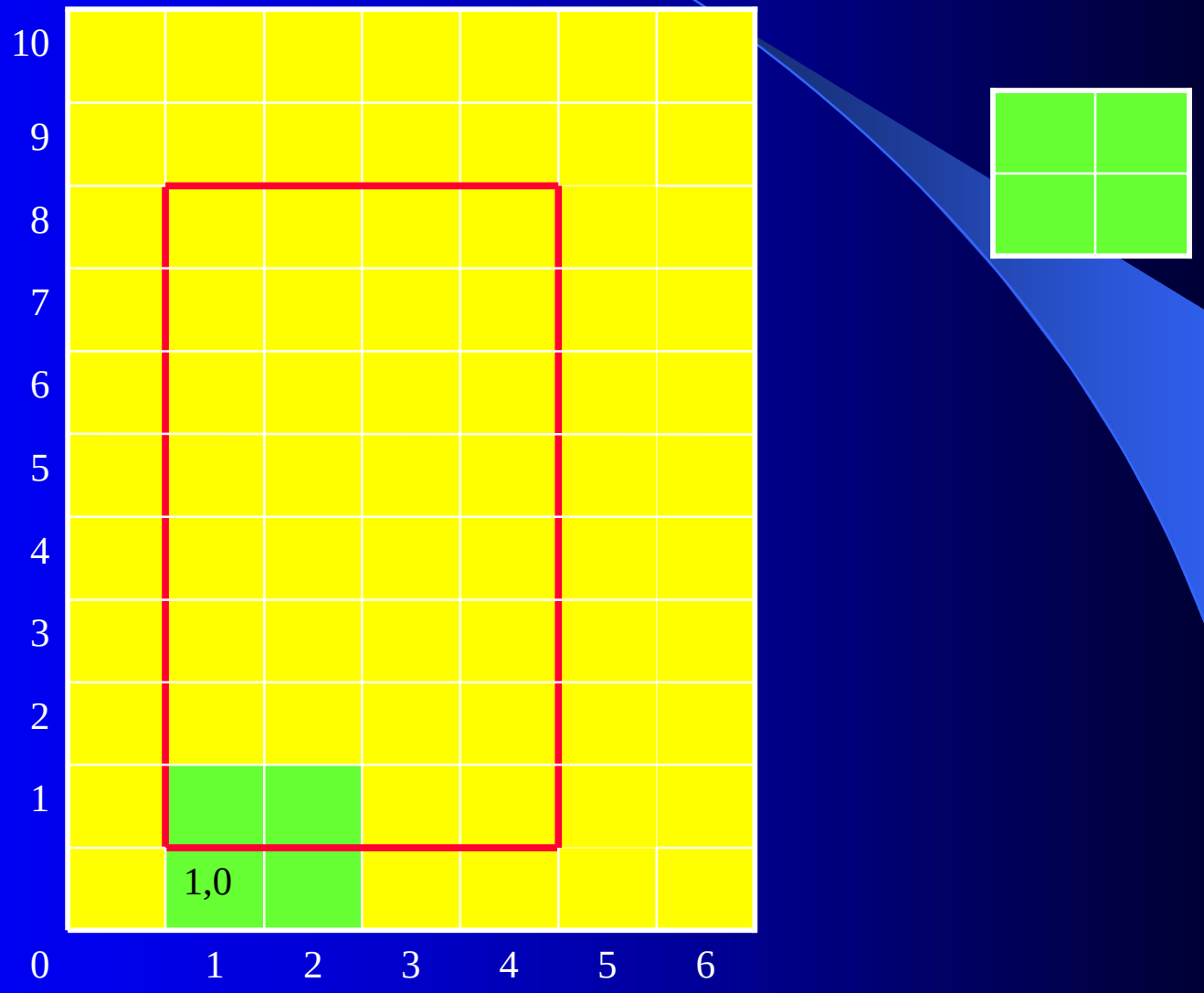

Input 77 Jet elements (2X5 bits Least Significant First)
Step 1
Step 2
Step 3
Step 4
Step 5
Compare 2X2 neighbors.
2X2
2X2
Apply Thresh. to 2X2.
4X4
1X2
2X2
2X4
Apply Thresh. to 4X4.
3X3
2X3
1X3
Apply Thresh. to 3X3.


Input 77 Jet elements (2X5 bits Least Significant First)
Step 1
Step 2
Step 3
Step 4
Step 5
Compare 2X2 neighbors.
2X2
2X2
Apply Thresh. to 2X2.
4X4
1X2
2X2
2X4
Apply Thresh. to 4X4.
3X3
2X3
1X3
Apply Thresh. to 3X3.

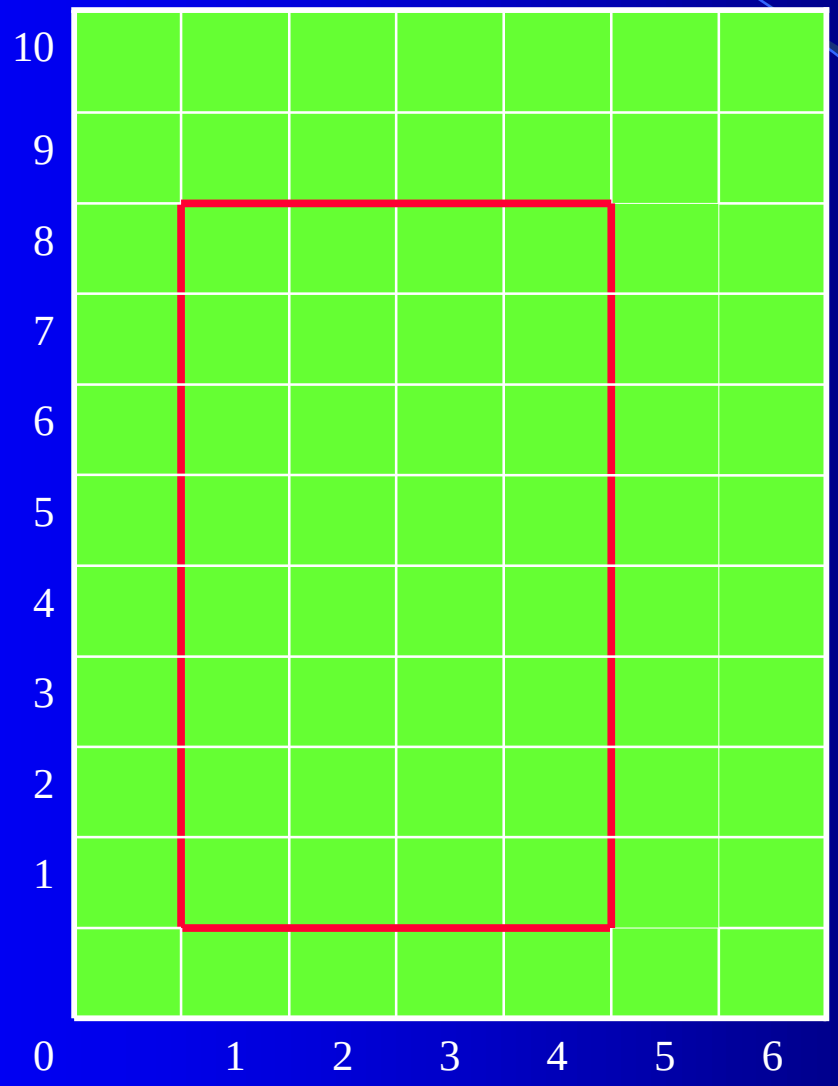
2x2 Jet element Sum (Step 2)



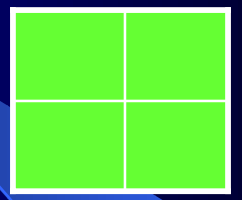
2x2 Jet element Sum (Step 2)

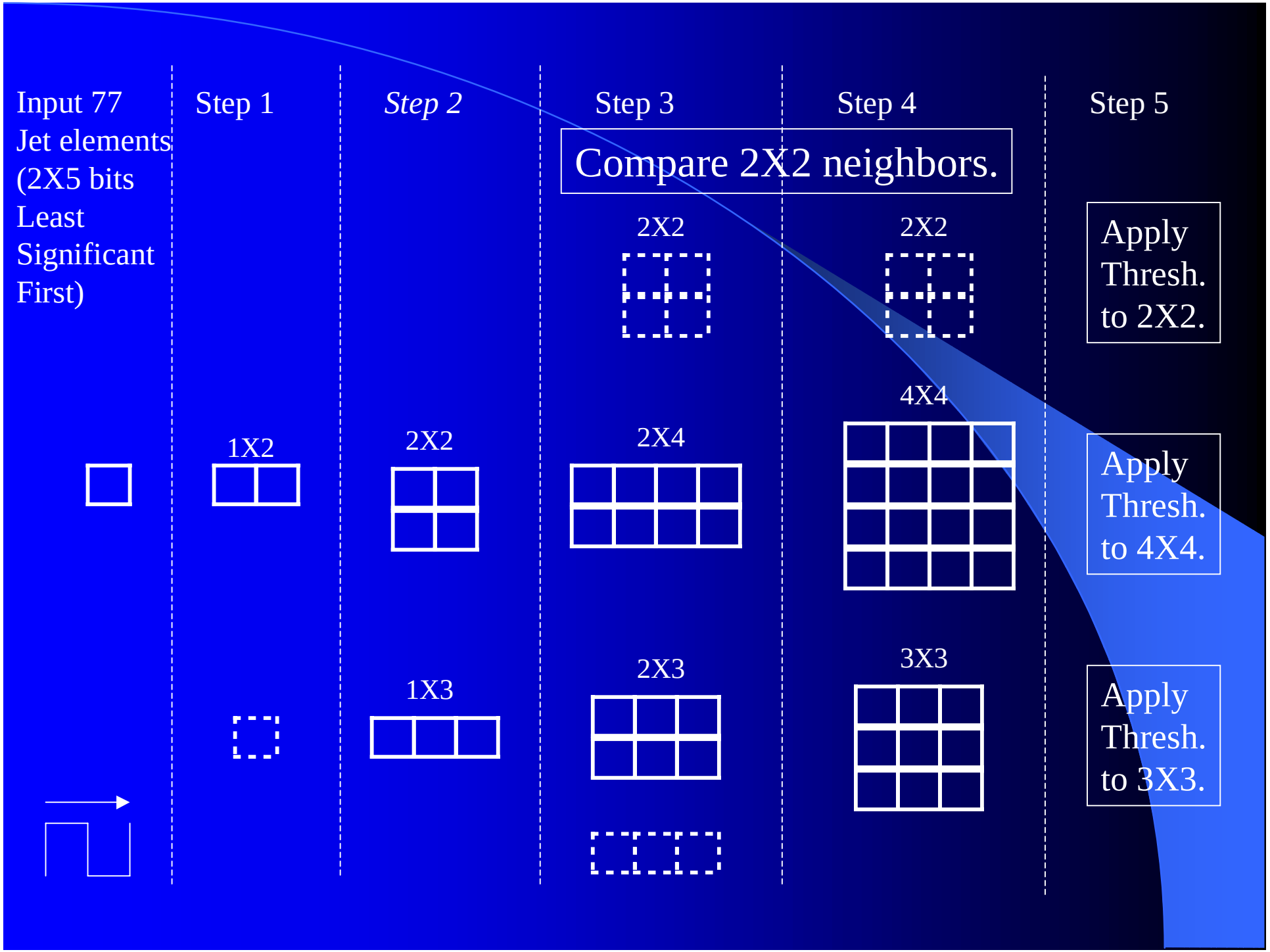


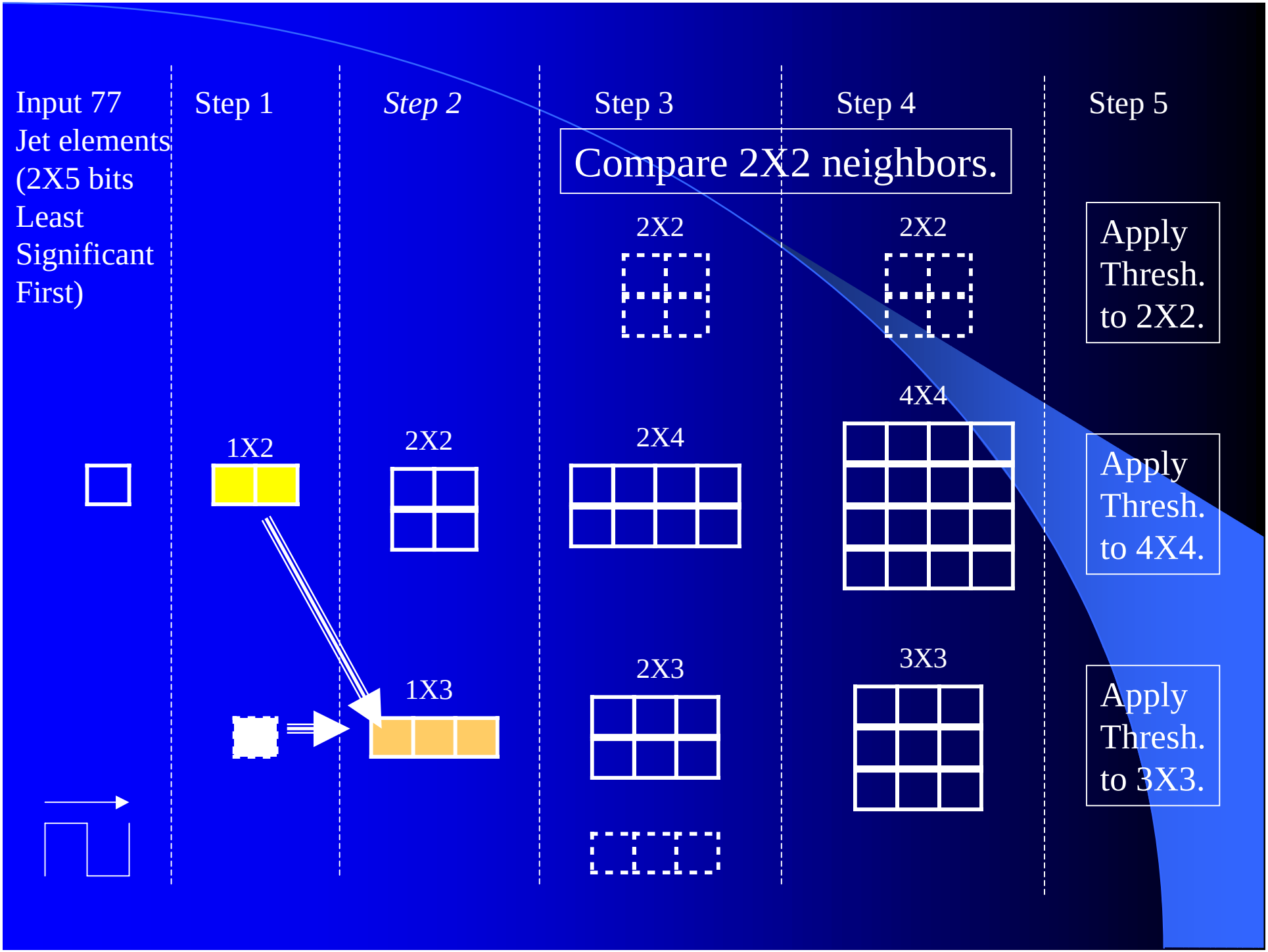
2x2 Jet element Sum (Step 2)



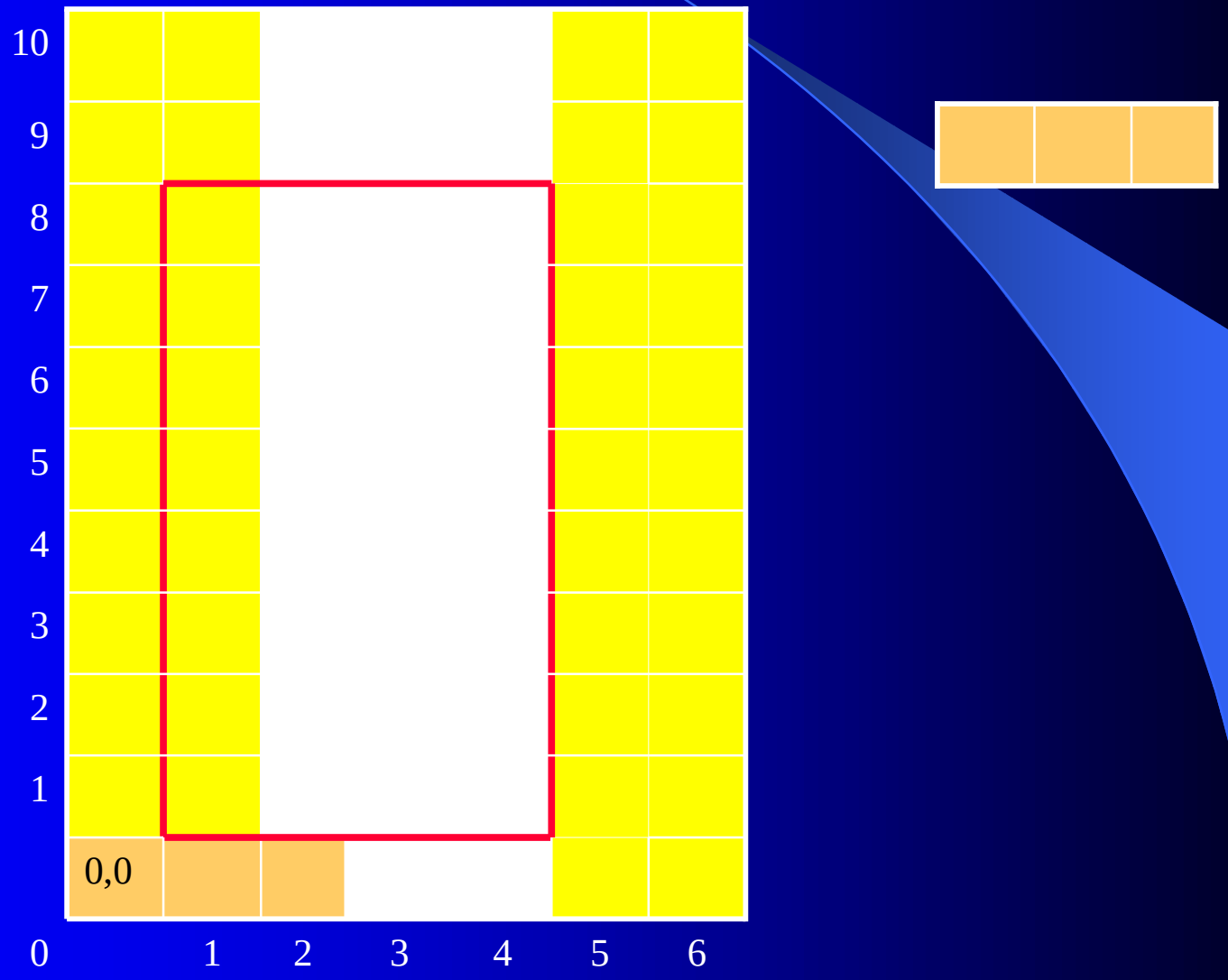
Total 10x6 summations



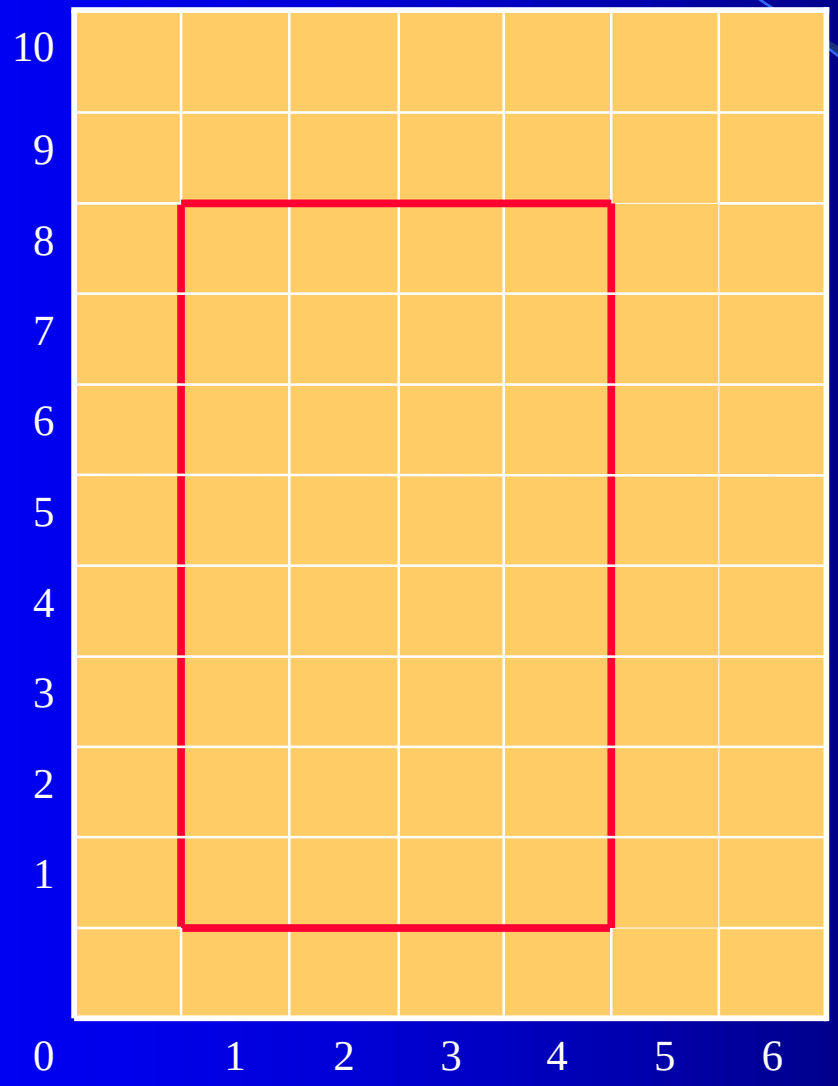

Input 77 Jet elements (2X5 bits Least Significant First)
Step 1
Step 2
Step 3
Step 4
Step 5
Compare 2X2 neighbors.
2X2
2X2
Apply Thresh. to 2X2.
4X4
1X2
2X2
2X4
Apply Thresh. to 4X4.
3X3
2X3
1X3
Apply Thresh. to 3X3.


Input 77 Jet elements (2X5 bits Least Significant First)
Step 1
Step 2
Step 3
Step 4
Step 5
Compare 2X2 neighbors.
2X2
2X2
Apply Thresh. to 2X2.
4X4
1X2
2X2
2X4
Apply Thresh. to 4X4.
1X3
2X3
3X3
Apply Thresh. to 3X3.

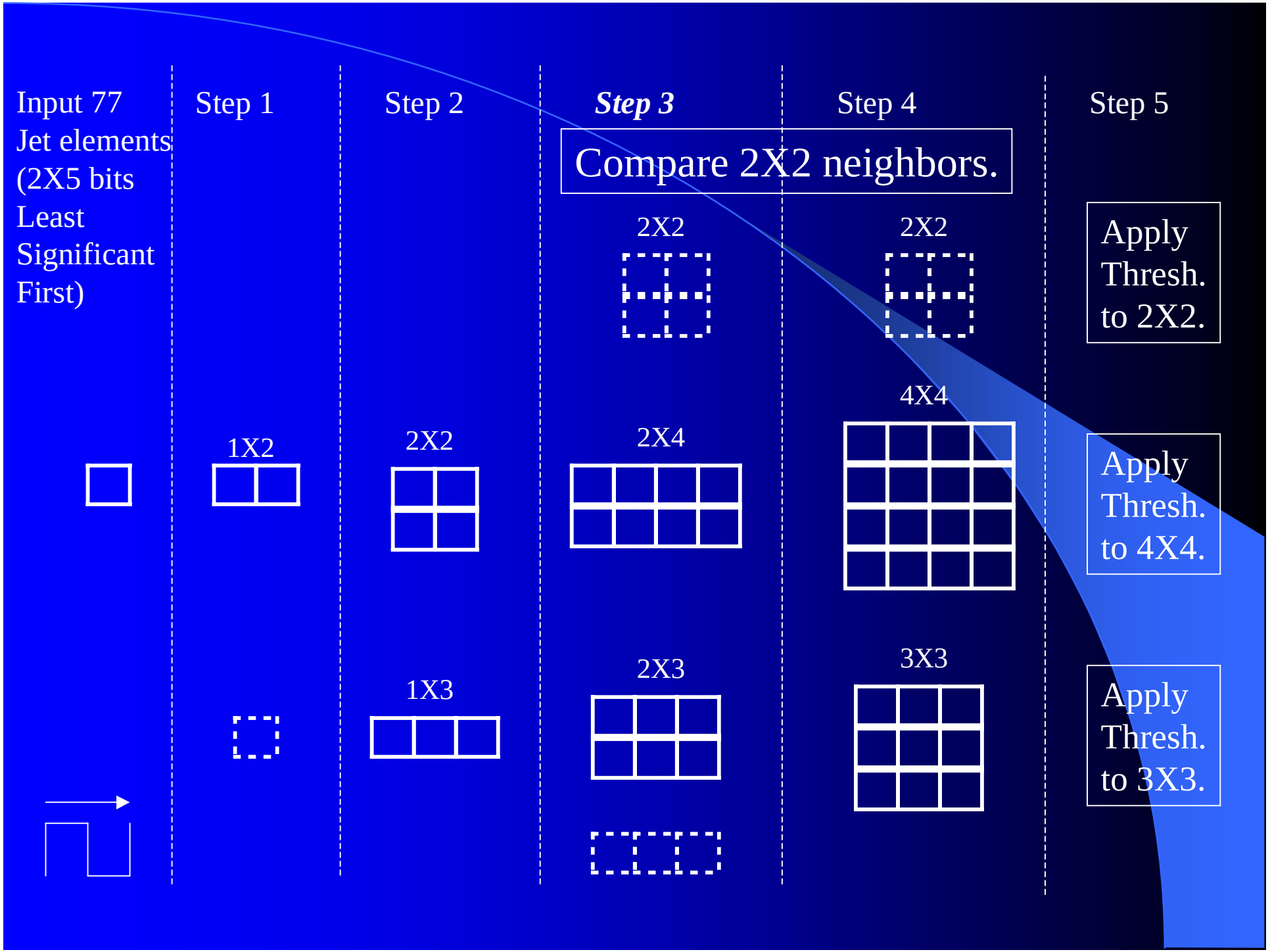
1x3 Jet element Sum (Step 2)



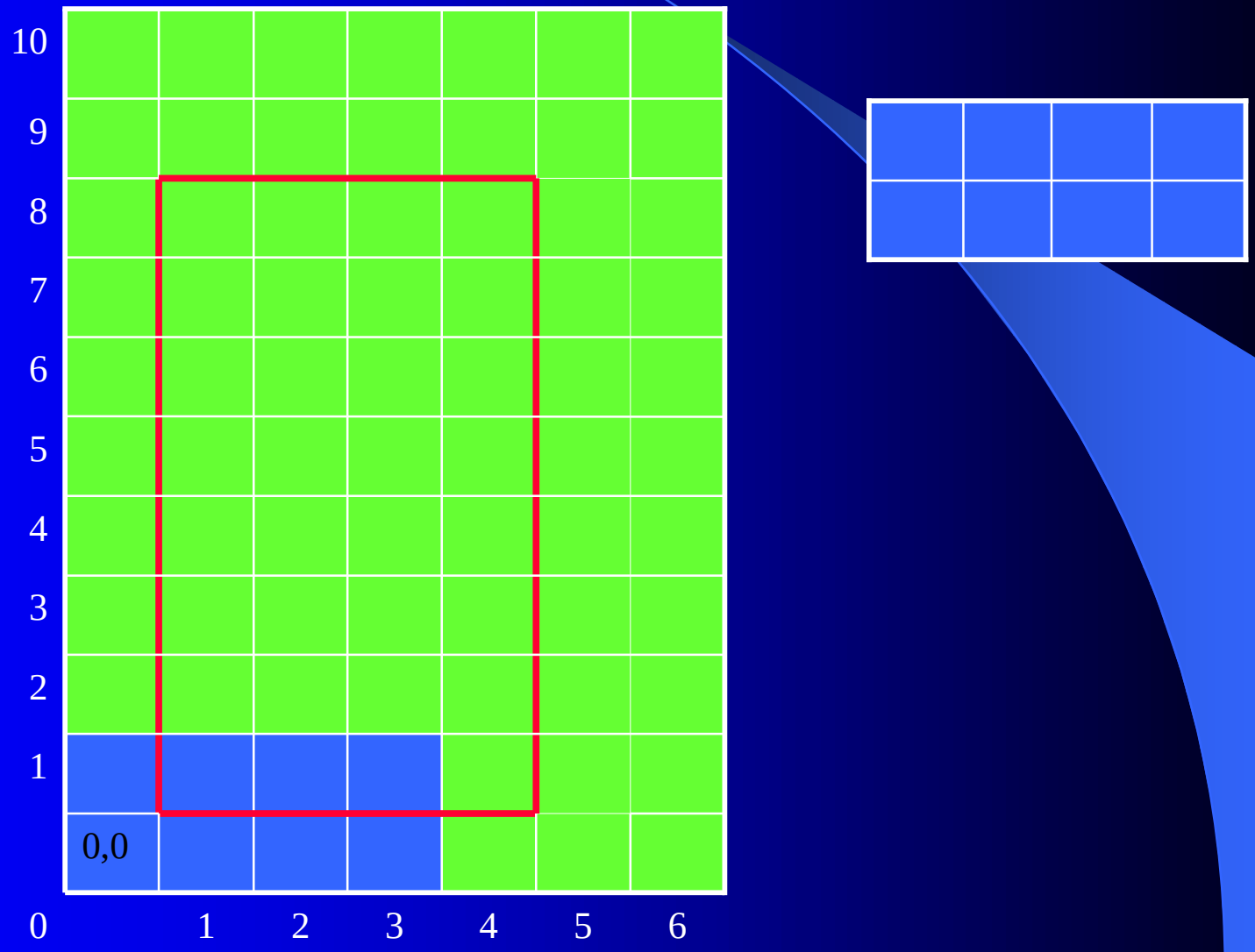
1x3 Jet element Sum (Step 2)



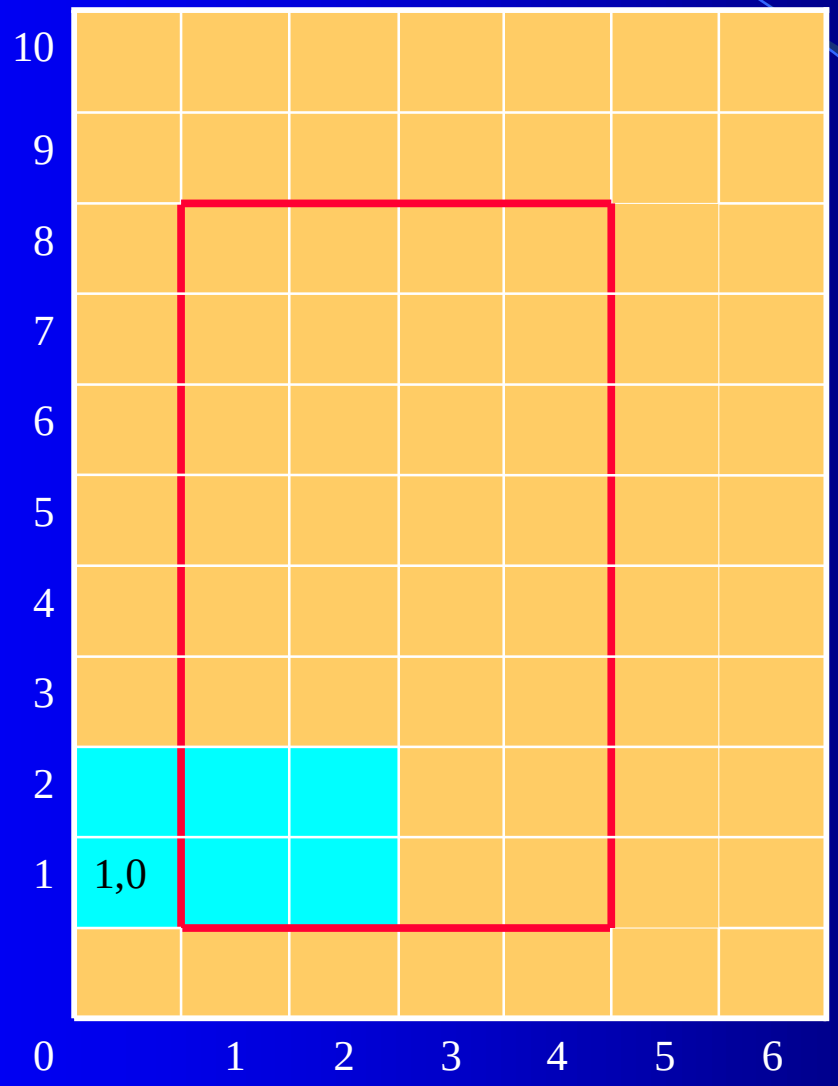
Total 11x5 summations


Input 77 Jet elements (2X5 bits Least Significant First)
Step 1
Step 2
Step 3
Step 4
Step 5
Compare 2X2 neighbors.
2X2
2X2
Apply Thresh. to 2X2.
4X4
1X2
2X2
2X4
Apply Thresh. to 4X4.
3X3
2X3
1X3
Apply Thresh. to 3X3.

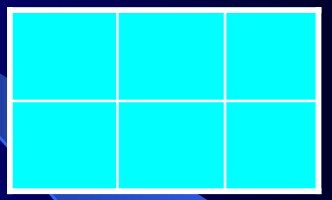
2x4 Jet element Sum (Step 3)

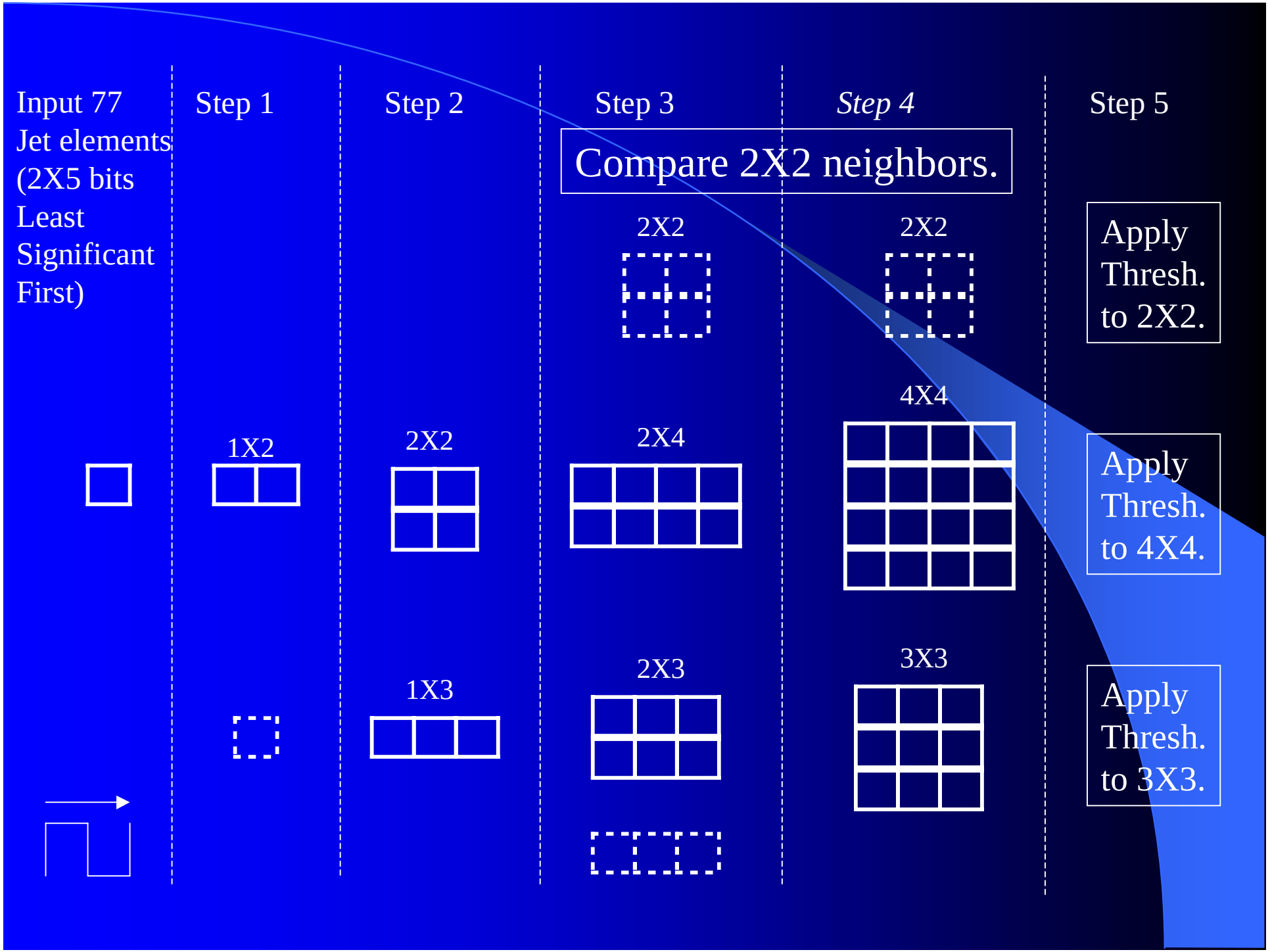


2x3 Jet element Sum (Step 3)

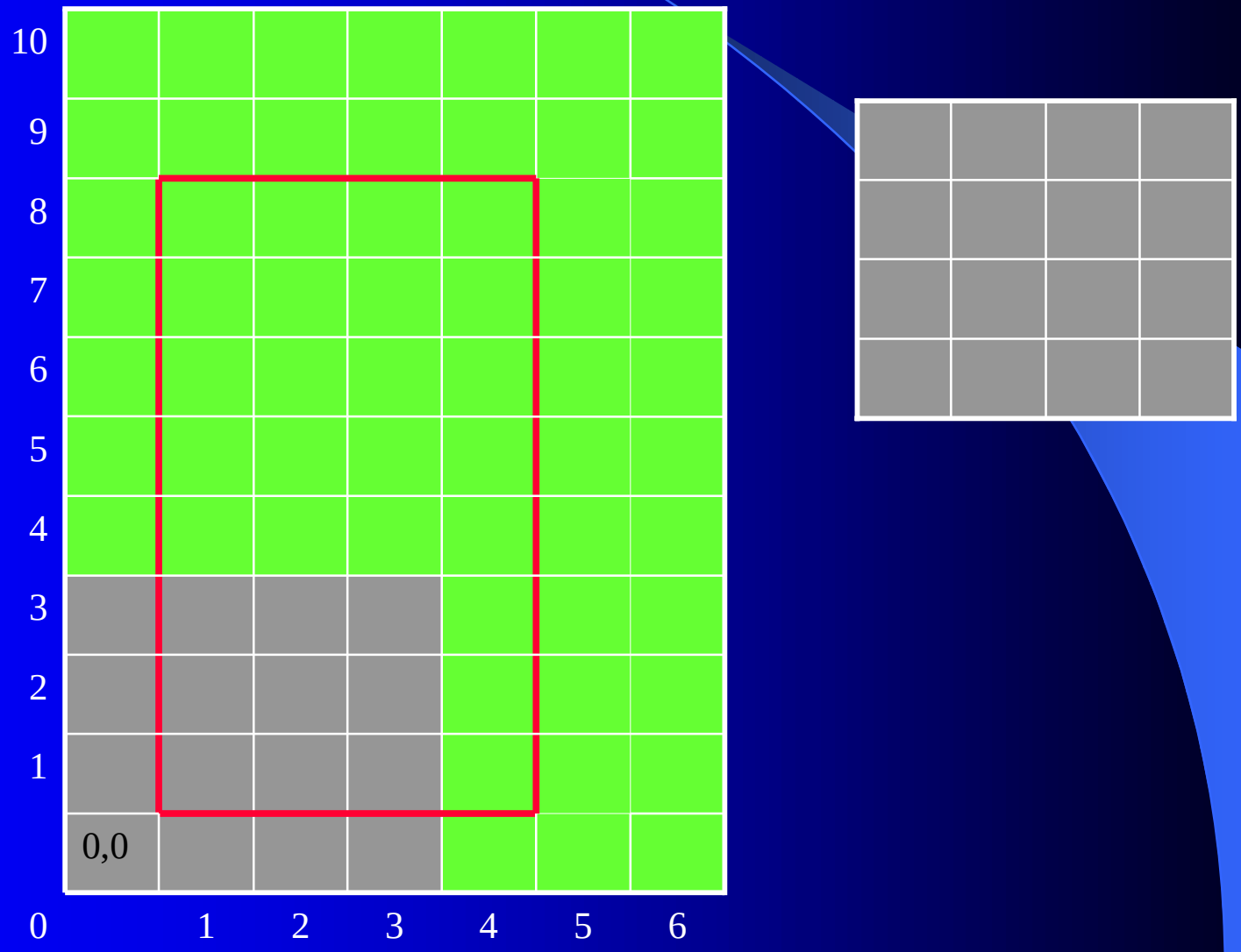


Total 1 summation

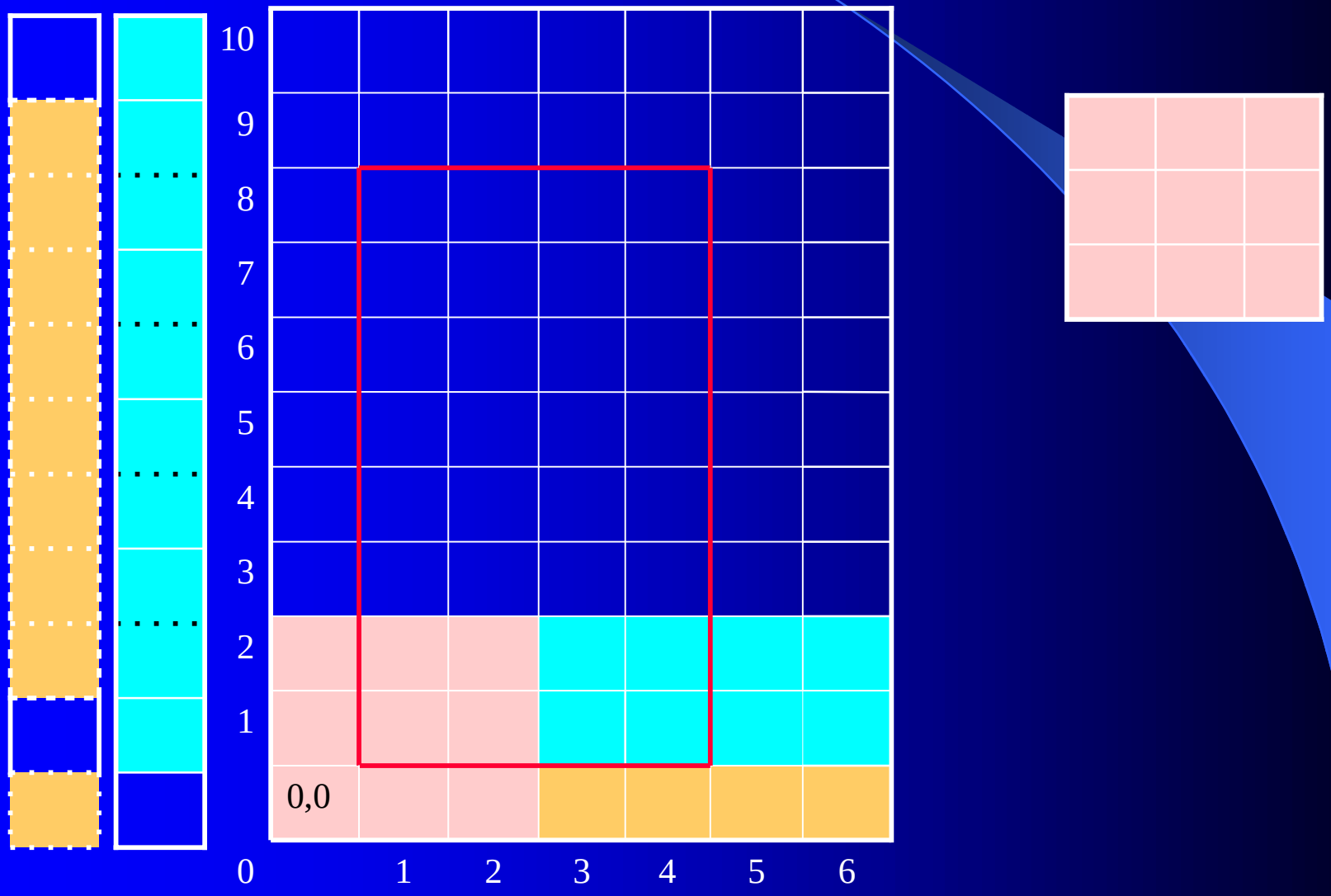


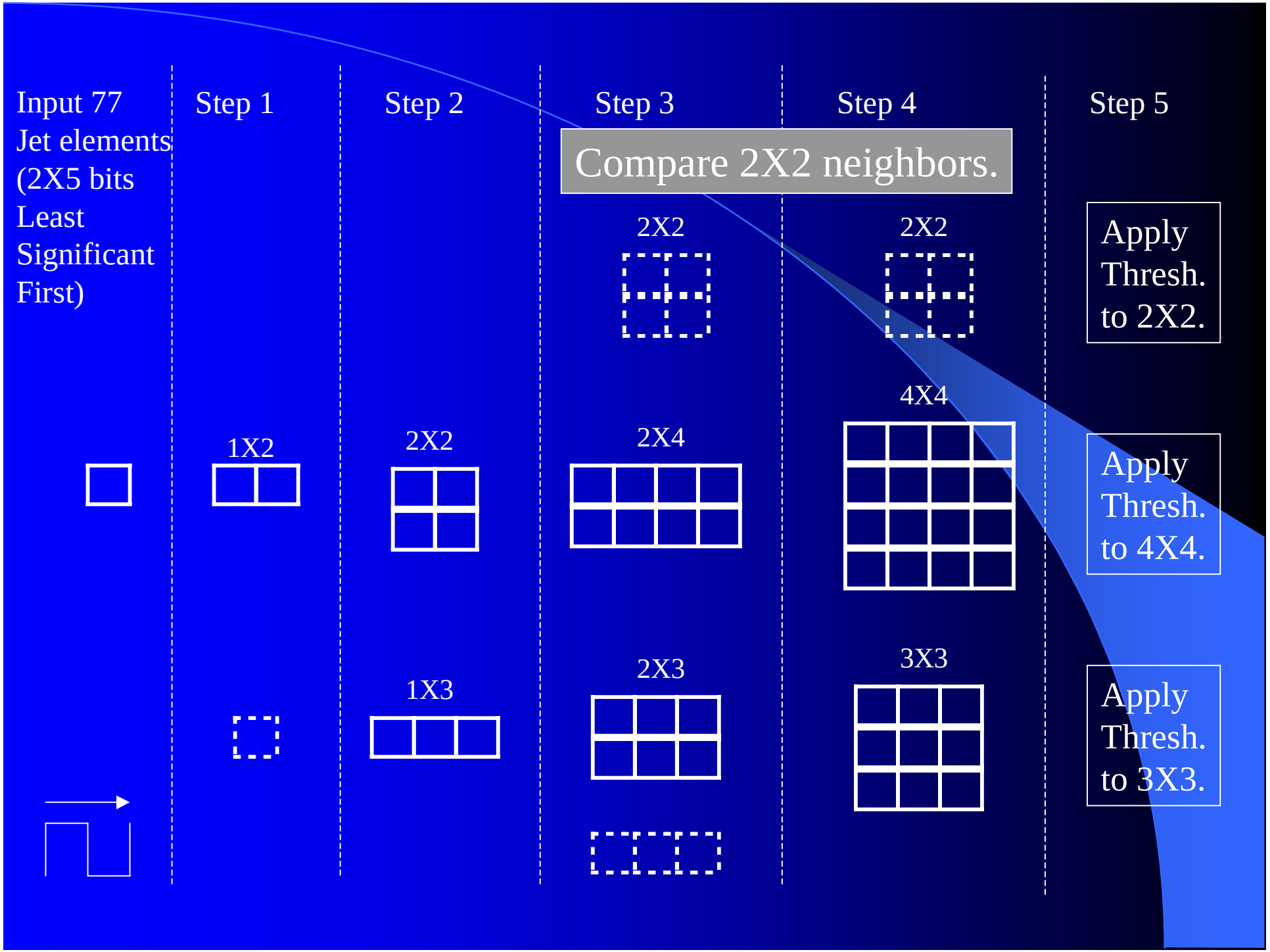

Input 77 Jet elements (2X5 bits Least Significant First)
Step 1
Step 2
Step 3
Step 4
Step 5
Compare 2X2 neighbors.
2X2
2X2
Apply Thresh. to 2X2.
4X4
1X2
2X2
2X4
Apply Thresh. to 4X4.
3X3
2X3
1X3
Apply Thresh. to 3X3.

4x4 Jet element Sum (Step 4)

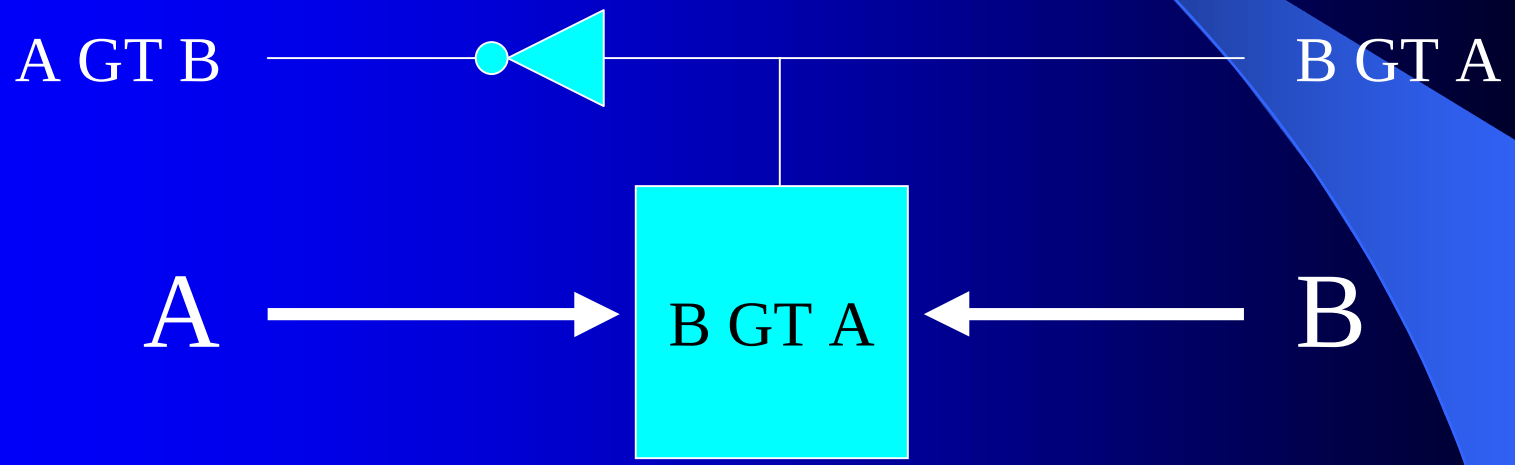


3x3 Jet element Sum (Step 4)



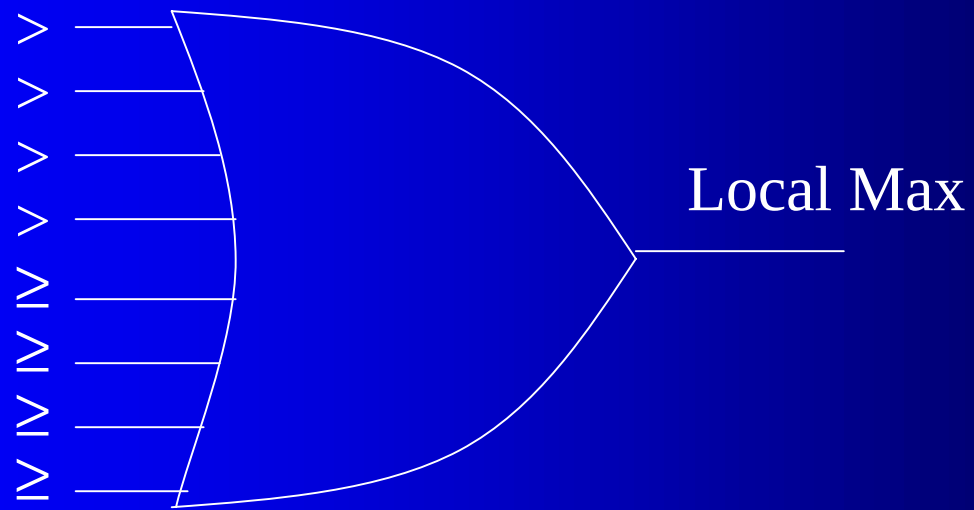

Input 77 Jet elements (2X5 bits Least Significant First)
Step 1
Step 2
Step 3
Step 4
Step 5
Compare 2X2 neighbors.
2X2
2X2
Apply Thresh. to 2X2.
4X4
1X2
2X2
2X4
Apply Thresh. to 4X4.
2X3
3X3
1X3
Apply Thresh. to 3X3.

Comparators can be shared



Sam Silverstein:

Local maximum is OR of eight comparisons



Status of algorithm

- Adder trees: completed
- Local maximum ID: completed
- Jet cluster selection: completed
- Threshold comparison: completed
- Level-1, ROI result reporting: nearly done
- Test vector generation: starting to write C++ code

Things left to do

- Finish algorithm
- Synthesize for Virtex - size estimate!
- Simulate and verify with test vectors
- Integrate with rest of algorithm on main processor FPGA
- Produce test vectors for slice tests

Some final words

- We are approaching the point where we need to interface with other work
  - Integration of Jet algorithm on JEM main processor FPGA
  - Testing of algorithm: HDMC?
- Who should Anders and Torbjörn be contacting?