High Performance Computing Systems

- Numeric Representations

Doug Shook

Review: Numeric Representations

- Integers
- Fixed Point
- Floating Point
- Limitations?

Limitations

- Overflow / underflow
- "Gaps"

Normalization

- Consider:

  .5 x 10^2 == .05 x 10^3

- Problems with this?

- Solutions?

Normalization

- The first digit of the mantissa will always be 1
  - Implications of this?
    - Storage?
    - Underflows?

Representation errors

- What approaches can we take for numbers that cannot be represented?
  - Which approach is best?

Machine Precision

- The error can be quantified as maximum relative error
  - Maximum attainable accuracy
- Implications of this?
- Exercise: write a program that computes epsilon

IEEE Standard

- Declared standardized mantissa sizes
- Specified rounding behavior
- Defines two special quantities for exceptional cases

IEEE Standard - Rounding

- Rounding determines accuracy of operations
- Defined as follows:
  - Compute exact answer
  - Round to the *nearest* representable number
- Also requires use of guard digits

Round-off Error

- Each operation has error associated with it
  - Addition
  - Subtraction
  - Multiplication

- Implications?

Compilation of floating point

- Some of the laws of math don't apply to floating point....
  - How might this affect compiler optimization?
- Different kinds of reproducibility:
  - Bitwise
  - "Scientific"
- What about parallel execution?

FPGA Floating Point Acceleration

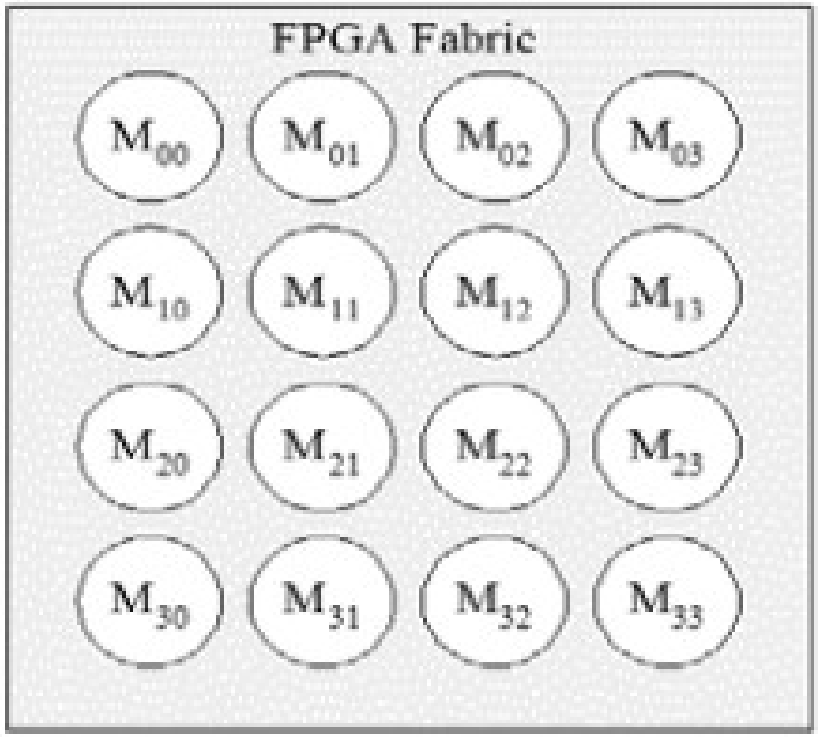
- HPC nodes are becoming more and more heterogeneous
- FPGAs struggle with floating point...
  - DSPs
- Cappello and Strenski, "A Practical Measure of FPGA Floating Point Acceleration for High Performance Computing", 2013

FPGA Floating Point Acceleration

- Xilinx XC7VX690T
  - 3600 DSPs
- What is the best performance that can be achieved?
  - Matrix Multiply
  - How many DSPs to implement?
  - Best configuration?
  - Highest clock speed?

Matrix Multiply

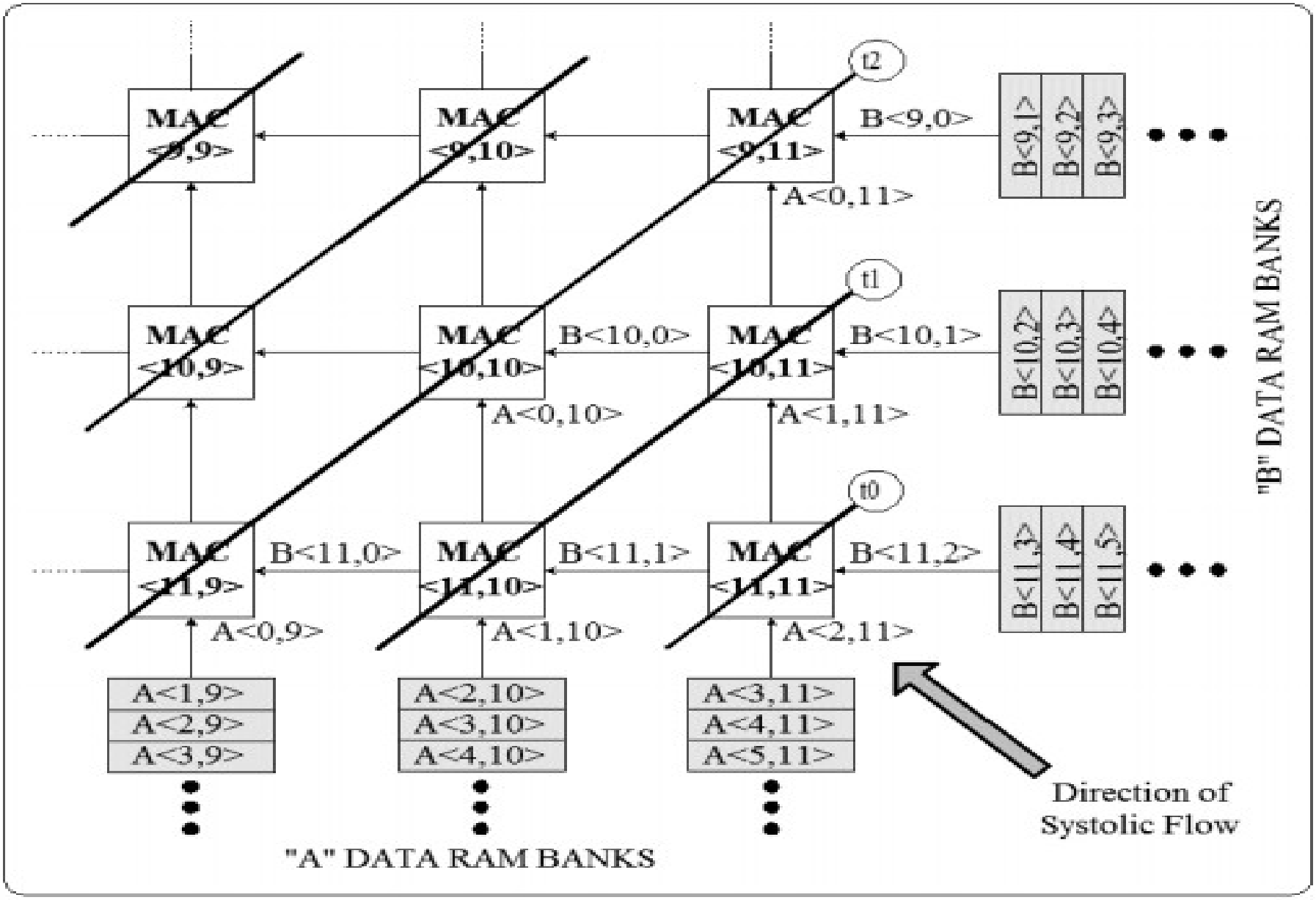
$$\underset{\mathbf{A}}{\overset{4x4}{\begin{bmatrix} A_{00} & A_{01} & A_{02} & A_{03} \\ A_{10} & A_{11} & A_{12} & A_{13} \\ A_{20} & A_{21} & A_{22} & A_{23} \\ A_{30} & A_{31} & A_{32} & A_{33} \end{bmatrix}}} \times \underset{\mathbf{B}}{\overset{4x4}{\begin{bmatrix} B_{00} & B_{01} & B_{02} & B_{03} \\ B_{10} & B_{11} & B_{12} & B_{13} \\ B_{20} & B_{21} & B_{22} & B_{23} \\ B_{30} & B_{31} & B_{32} & B_{33} \end{bmatrix}}} = \underset{\mathbf{C}}{\begin{bmatrix} A_{0x}B_{x0} & A_{0x}B_{x1} & A_{0x}B_{x2} & A_{0x}B_{x3} \\ A_{1x}B_{x0} & A_{1x}B_{x1} & A_{1x}B_{x2} & A_{1x}B_{x3} \\ A_{2x}B_{x0} & A_{2x}B_{x1} & A_{2x}B_{x2} & A_{2x}B_{x3} \\ A_{3x}B_{x0} & A_{3x}B_{x1} & A_{3x}B_{x2} & A_{3x}B_{x3} \end{bmatrix}}$$



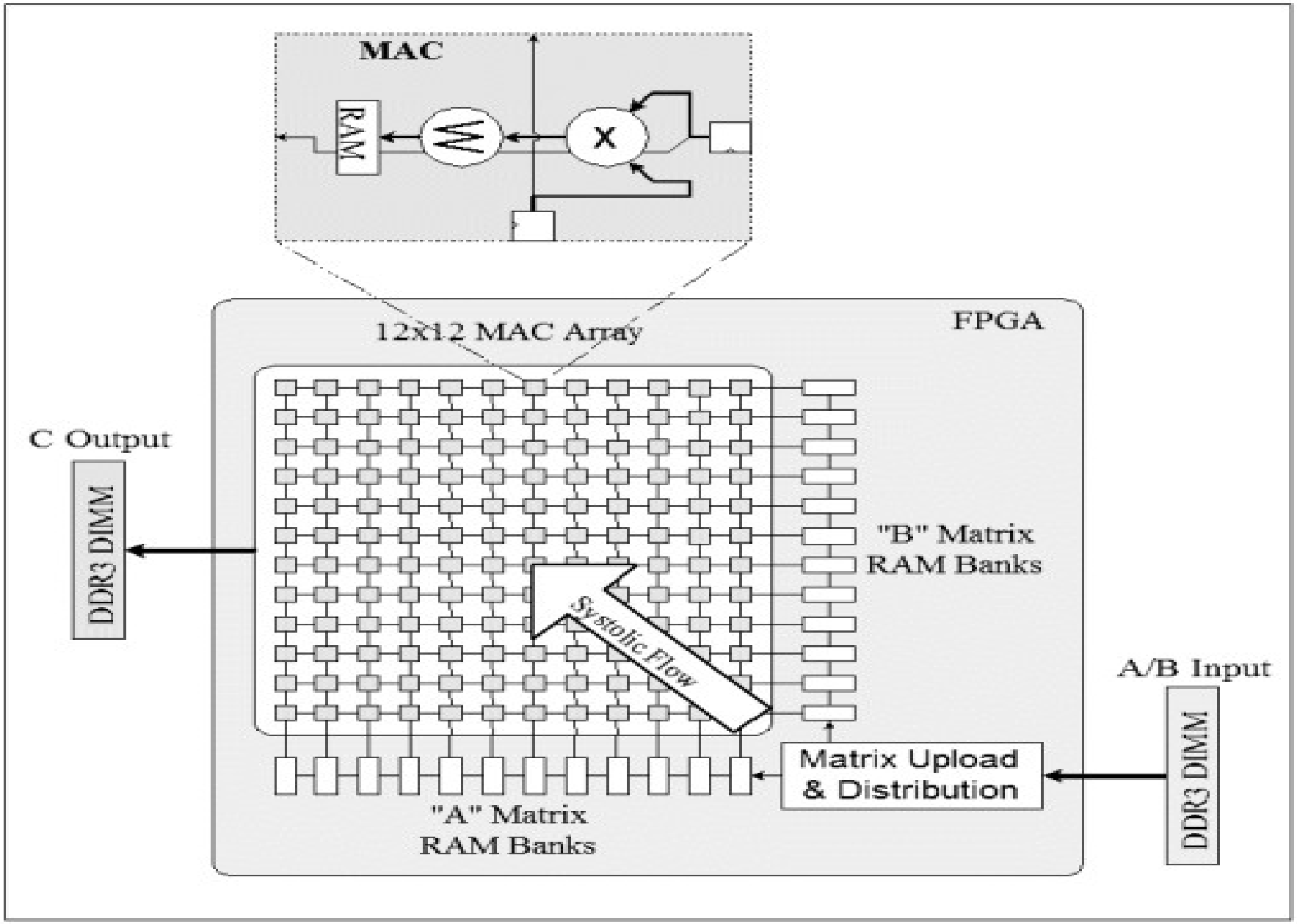
Dot products 1:1 with MACs

M = MAC

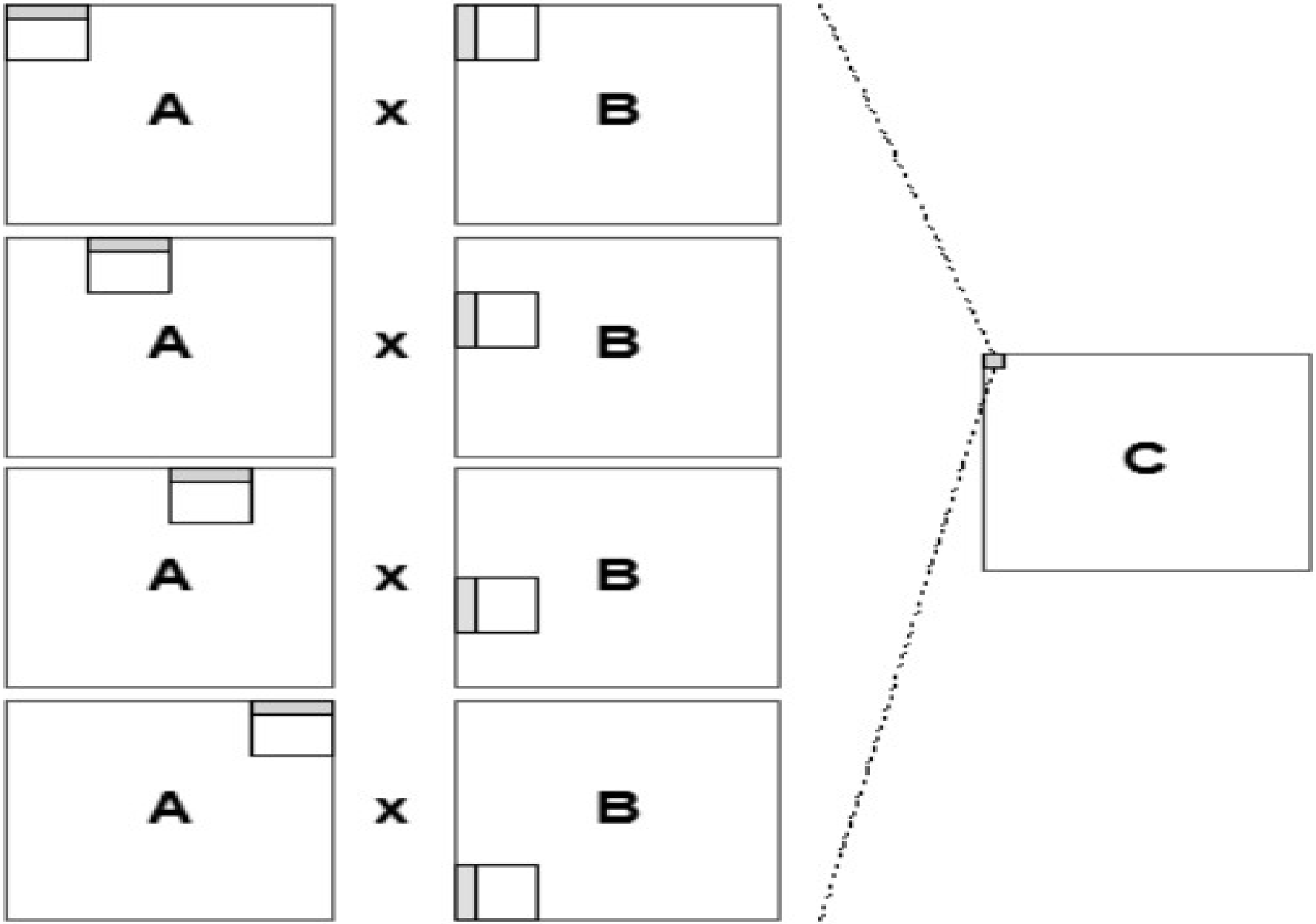
Data Distribution



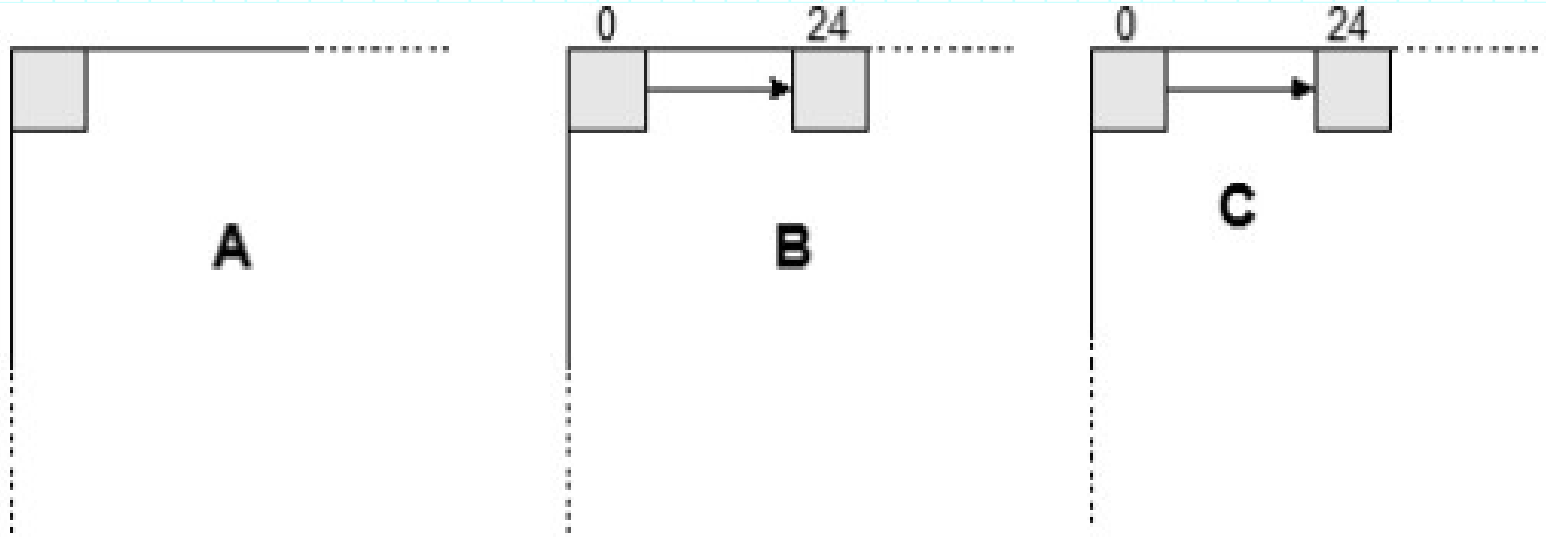
FPGA Floating Point Acceleration



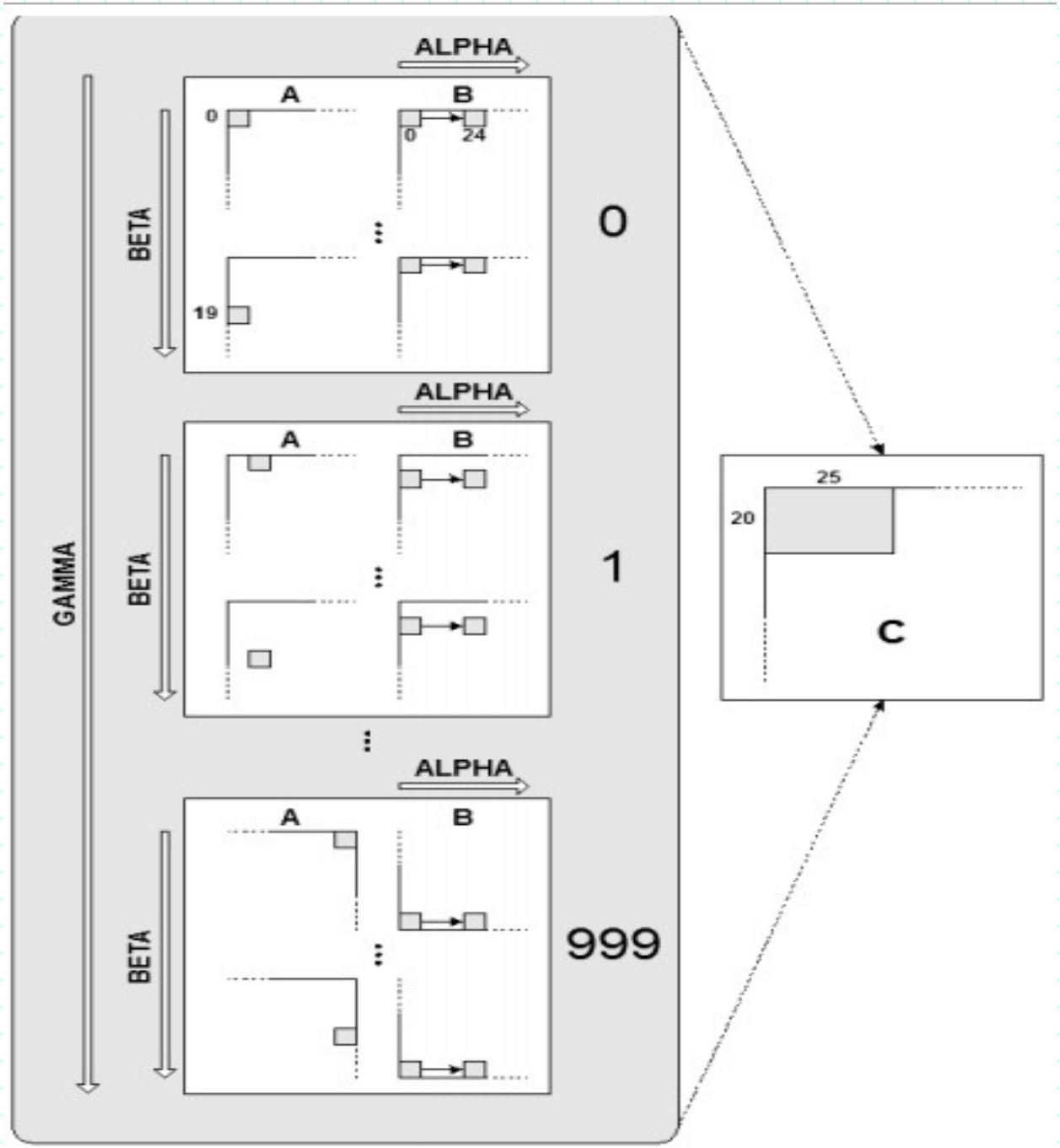
Blocking



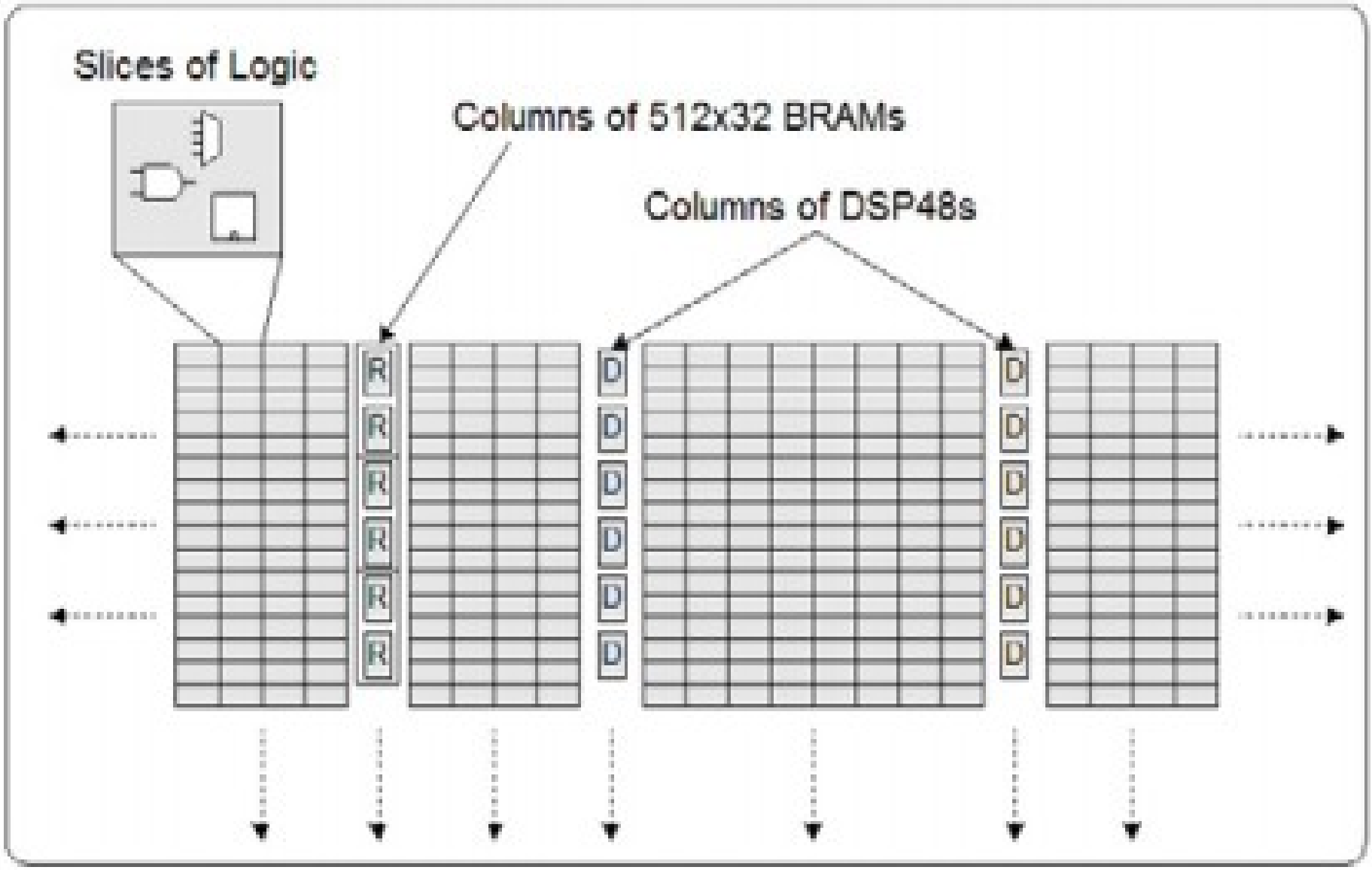
Blocking



Blocking


ALPHA
A
B
0
0
24
BETA
19
0
ALPHA
A
B
GAMMA
BETA
1
ALPHA
A
B
BETA
999
25
20
C

Layout


Slices of Logic
Columns of 512x32 BRAMs
Columns of DSP48s
R
D

Conclusions

- Does performance live up to theoretical max?
- What can be done to improve performance?