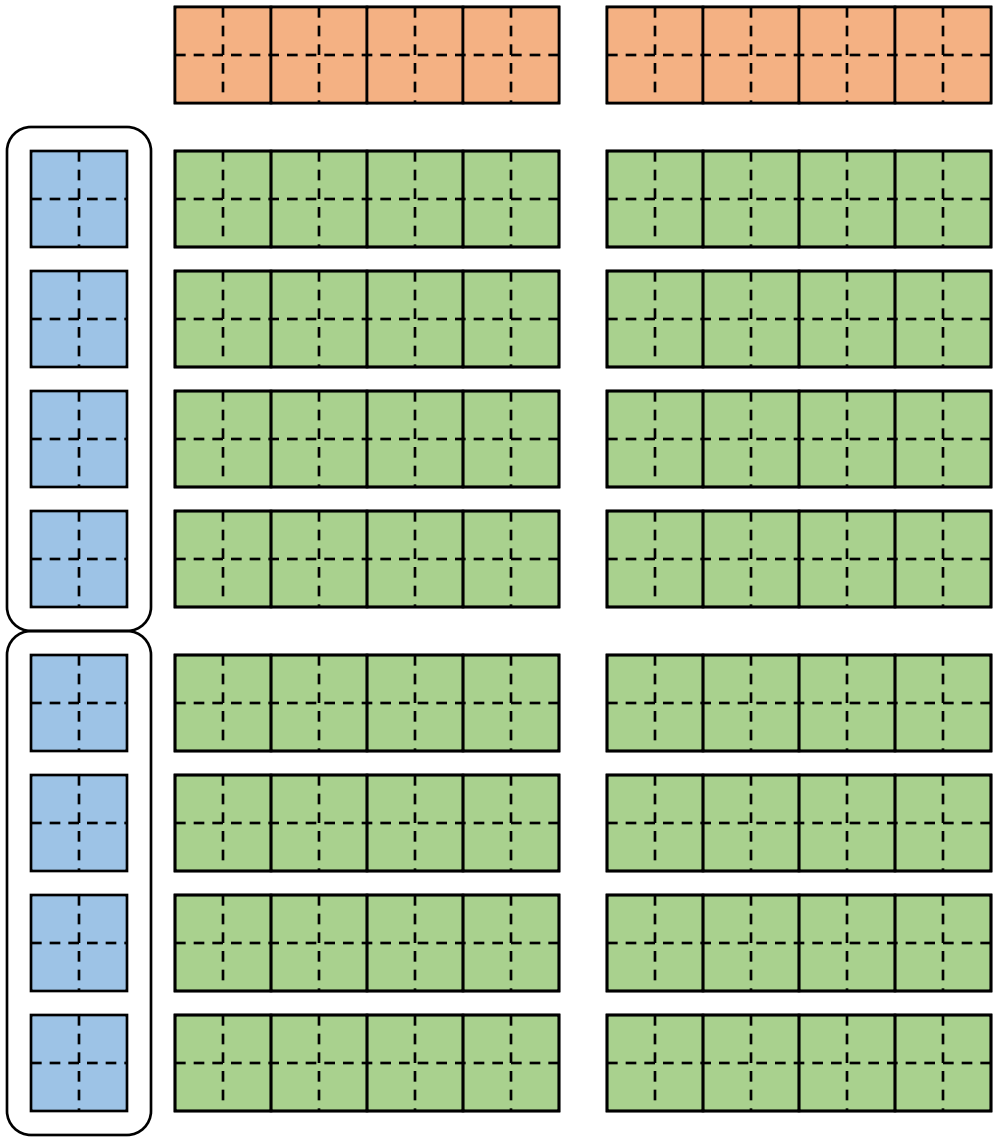
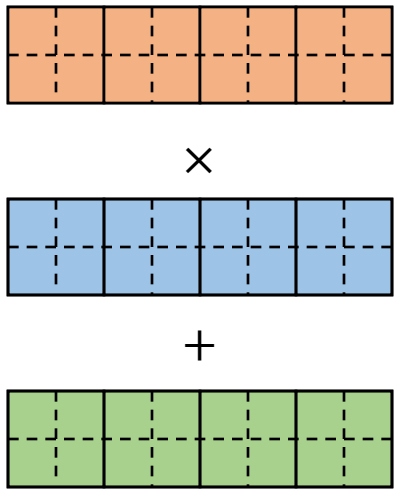
# Option C: Matrix as an element type



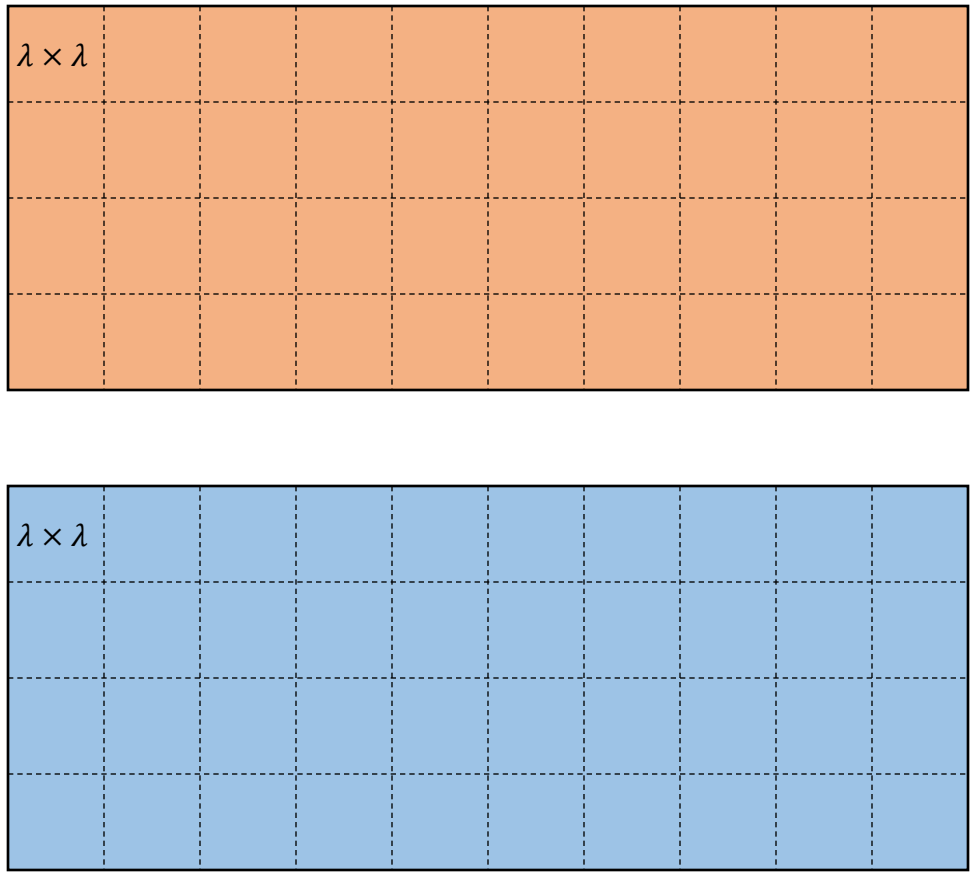
- We fix a $\lambda$ and define a new "vector element type" – a $\lambda \times \lambda$ matrix of words (e.g., $\lambda = 2$)
- A vector register of length $L$ holds ${}^{L}/_{\lambda^2}$ of these matrices (e.g., $L = 16, {}^{L}/_{\lambda^2} = 4$)
- "Some" number of registers hold an $8\lambda \times \lambda$ panel of $\boldsymbol{A}$ ($8\ \lambda \times \lambda$ matrices)
- 2 registers hold a $\lambda \times {}^{2L}/_{\lambda}$ panel of $\boldsymbol{B}$ (${}^{2L}/_{\lambda^2}\ \lambda \times \lambda$ matrices)
- 16 registers hold an $8\lambda \times {}^{2L}/_{\lambda}$ panel of $C$ ($16L$ words)
- We compute $\boldsymbol{C}_{8\lambda \times {}^{2L}/_{\lambda}} \leftarrow \boldsymbol{A}_{8\lambda \times \lambda} \times \boldsymbol{B}_{\lambda \times {}^{2L}/_{\lambda}} + \boldsymbol{C}_{8\lambda \times {}^{2L}/_{\lambda}}$
- Total of $16L\lambda$ multiply-adds
- Minimum time = $\lambda\Delta$ cycles
- Maximum computation rate $R = \frac{16L}{\Delta} = 4L$ madds/cycle
- This is the upper bound with 16 registers for $\boldsymbol{C}$
- Total of $2L + 8\lambda^2$ words loaded (${}^{2L+8\lambda^2}/_{\lambda\Delta}$ words/cycle)
- $\eta = \frac{16L\lambda}{2L+8\lambda^2} = [\frac{8\lambda}{5}, 8\lambda)$ madds/word
- This works $\forall L \geq \lambda^2$ words
- Single- and double-precision are compatible $\forall L \geq 2\lambda^2$ words

# Option C: Compute instructions



- A vector of matrix multiplies
  - $\boldsymbol{C}_{\lambda\times\lambda} \leftarrow \pm\boldsymbol{A}_{\lambda\times\lambda} \times \boldsymbol{B}_{\lambda\times\lambda} \pm \boldsymbol{C}_{\lambda\times\lambda}$ ($^{L}/_{\lambda^2}$ times)
  - Computations: $\lambda L$ madds/instruction
  - Latency: $\lambda\Delta$
  - Must dispatch/issue $^{4}/_{\lambda}$ computational instructions/operations every cycle to achieve maximum computation rate ($4L$ madds/cycle)

# Option C: Software impact




- For Option C to work, both $\boldsymbol{A}$ and $\boldsymbol{B}$ must be packed into $\lambda \times \lambda$ blocks – $\boldsymbol{A}$ in column-major, $\boldsymbol{B}$ in row-major
- Both the compute and packing kernels must be modified to support matrix operations
- $\boldsymbol{C}$ panel must also be reformatted for load/store