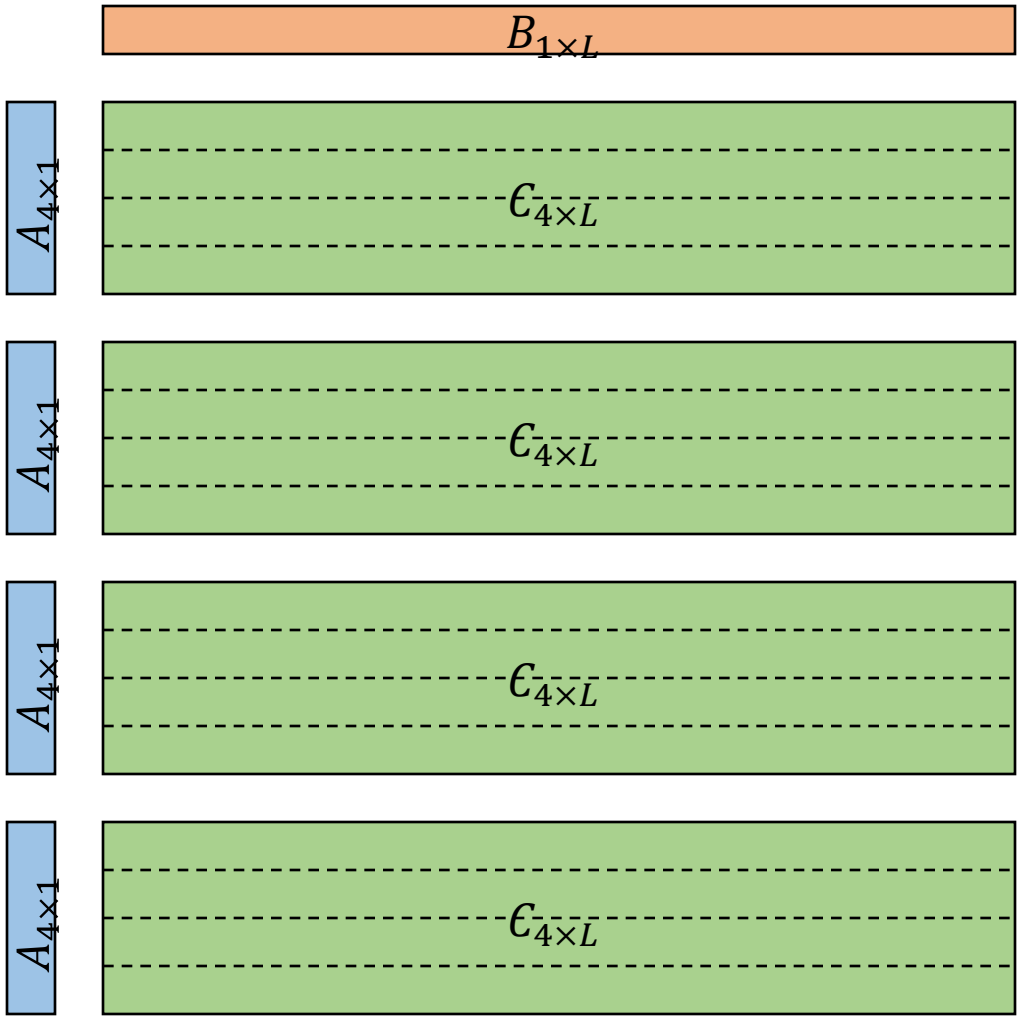
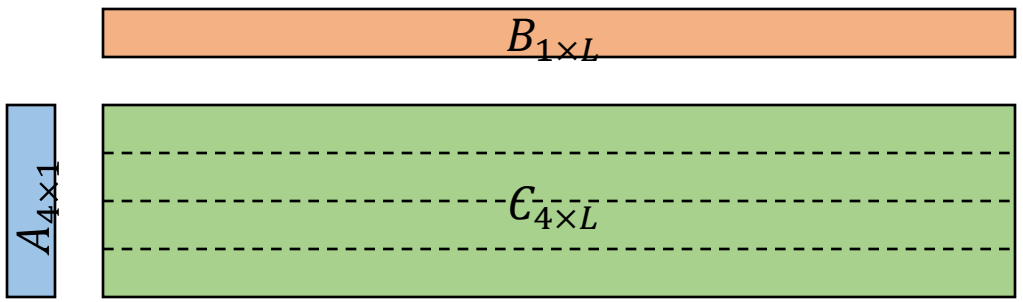
Option B: 1 Matrix in 4 vector registers



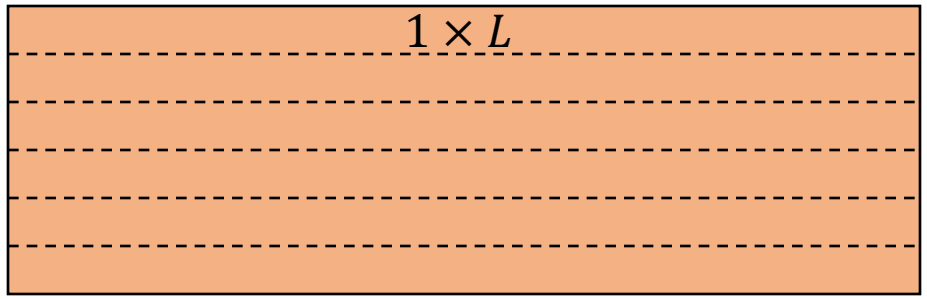
- An L-word vector register holds a row of $\boldsymbol{C}$
- 4 registers hold a $4 \times L$ panel of $\boldsymbol{C}$
- 16 registers hold a $16 \times L$ panel of $\boldsymbol{C}$
- An L-word vector register holds a row of $\boldsymbol{B}$
- "Some" combination of registers holds a column of $\boldsymbol{A}$
- We compute $\boldsymbol{C}_{16\times L} \leftarrow \boldsymbol{A}_{16\times 1} \times \boldsymbol{B}_{1\times L} + \boldsymbol{C}_{16\times L}$
- Total of $16L$ multiply-adds
- Minimum time = Δ cycles
- Maximum computation rate $R = \frac{16L}{\Delta} = 4L$ madds/cycle
- This is the upper bound with 16 registers for $\boldsymbol{C}$
- Total of $L + 16$ words loaded ($^{L+16}/_{\Delta}$ words/cycle)
- $\eta = \frac{16L}{L+16} = [\frac{16}{5}, 16)$ madds/word
- This works $\forall L \geq 4$ elements
- Single- and double-precision are compatible $\forall L \geq 8$

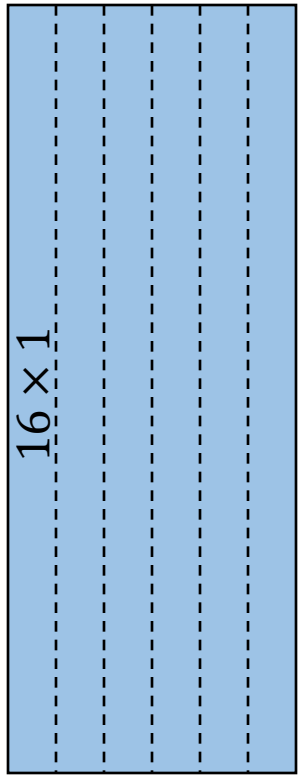
Option B: Compute instructions



- rank-1 update (outer product)
  - $C_{4\times L} \leftarrow \pm A_{4\times 1} \times B_{1\times L} \pm C_{4\times L}$
  - Computations: $4L$ madds/instruction
  - Latency: Δ
  - Must dispatch/issue 1 computational instruction/operation every cycle to achieve maximum computation rate ($4L$ madds/cycle)

Option B: Software impact





- Conventional BLAS formatting
  - $\boldsymbol{A}$ panel formatted as column-major $16 \times K$ matrix
  - $\boldsymbol{B}$ panel formatted as row-major $K \times L$ matrix
  - $\boldsymbol{C}$ panel does not have to be reformatted
- Easier pre/post-processing of rows/columns of matrices
  - Does not require reformatting data from/to vector format to/from matrix format
  - Although not that critical for matrix multiplication, insert/extract of rows from matrix registers from/to vector registers is useful in other algorithms (e.g., DFT)