# Qualitative Comparison

Andes Technology

# Outline

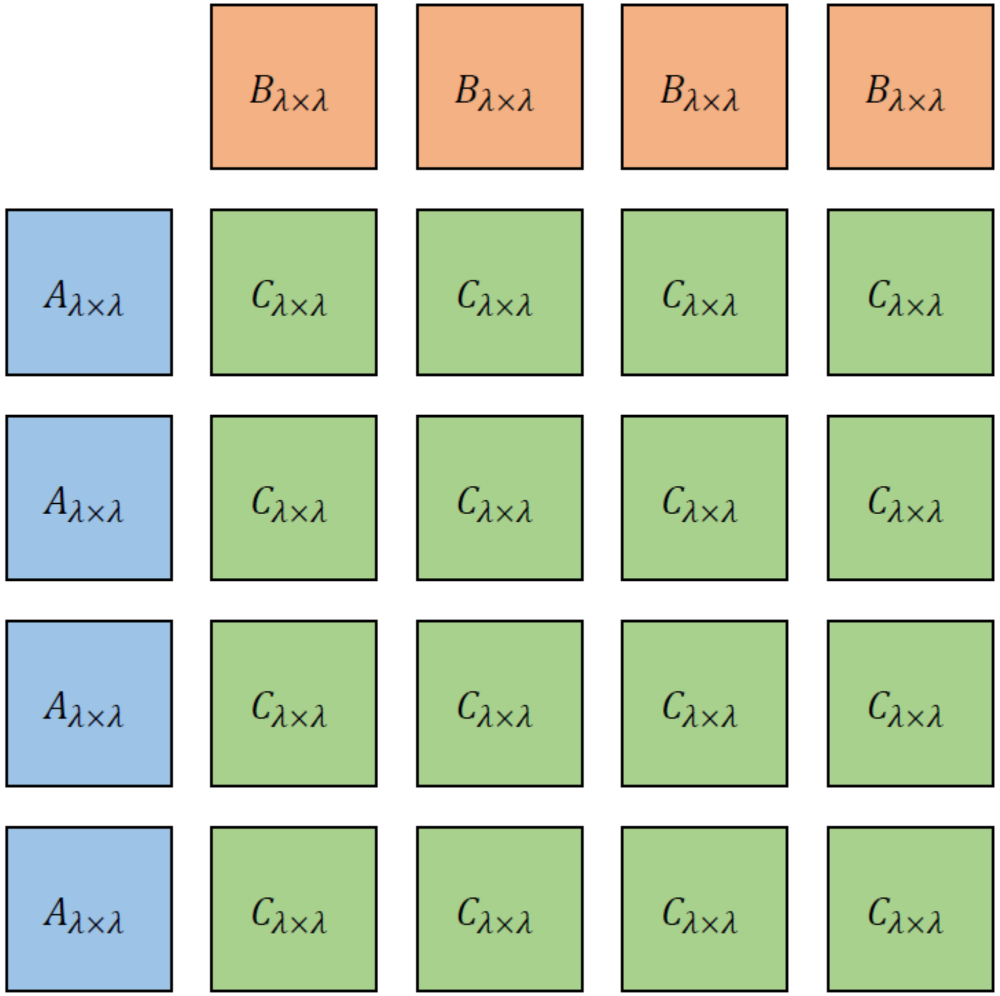
- Preface  Option A~E recap
- Fixes I/II/III Discussions
  - Analysis Program shared on GitHub
- Qualitative Comparison Based on
  - Restricted ACC Register Allocation
  - Theoretical Optimal Performance
- Charter Criteria (9+1) Comparison

# Preface

- Based on 2023/10/16 Meeting Presentation https://github.com/riscv-admin/vector/tree/main/minutes/2023/2023-10-16
- Terminology: (reused from V-SIG Conventions)
  - $$C_{m \times n} = A_{m \times K} \times B_{K \times n}$$
  - **L** : Let the architected space consist of 32 vector registers of size **L** words(32-bits)

# Option A Recap

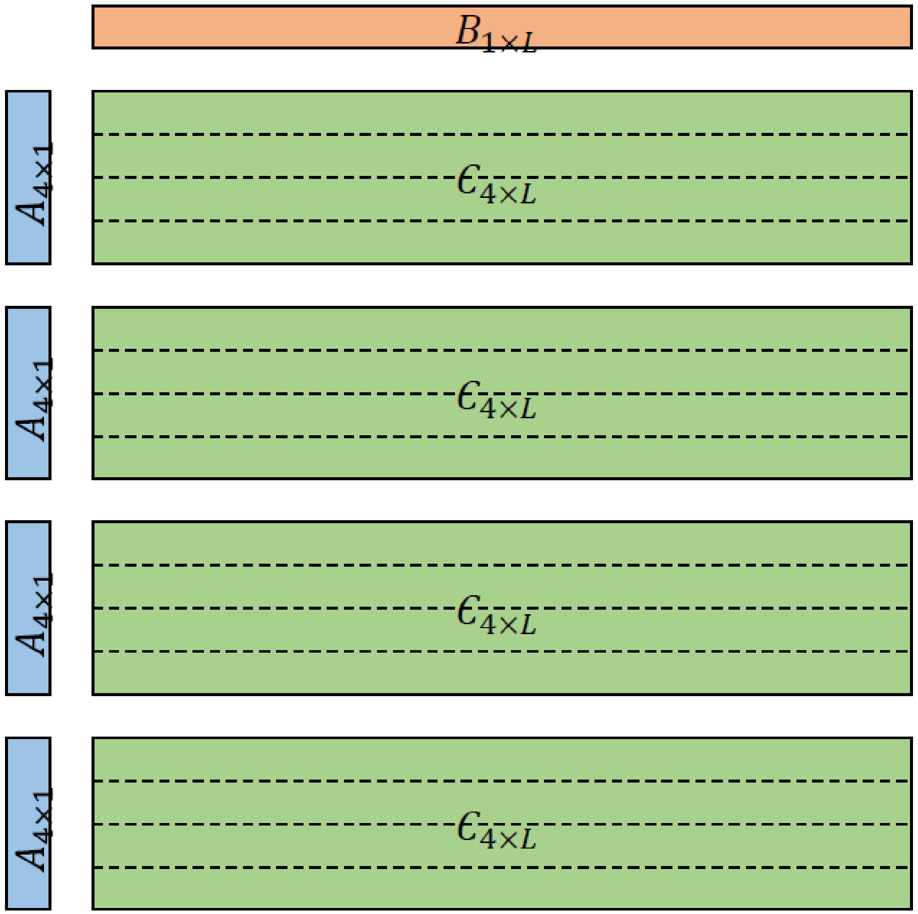
## Option A: 1 Matrix per vector register




- An $L$-word vector register holds an $\lambda \times \lambda$ matrix, $\lambda = \sqrt{L}$
- 16 registers hold a $4\lambda \times 4\lambda$ panel of $\boldsymbol{C}$
- 4 registers hold a $4\lambda \times \lambda$ panel of $A$
- 4 registers hold a $\lambda \times 4\lambda$ panel of $B$
- We compute $\boldsymbol{C}_{4\lambda\times4\lambda} \leftarrow \boldsymbol{A}_{4\lambda\times\lambda} \times \boldsymbol{B}_{\lambda\times4\lambda} + \boldsymbol{C}_{4\lambda\times4\lambda}$
- Total of $4\lambda \times 4\lambda \times \lambda = 16\lambda^3$ multiply-adds
- Minimum time = $\lambda\Delta$ cycles
- Maximum computation rate $R = \frac{16\lambda^3}{\lambda 4} = 4\lambda^2 = 4L$
- This is the upper bound with 16 registers for $\boldsymbol{C}$
- Total of $8\lambda^2$ elements loaded ($^{8\lambda}/_{\Delta}$ words/cycle)
- Computational intensity $\eta = \frac{16\lambda^3}{8\lambda^2} = 2\lambda$ madds/word
- This works for $L = 4, 16, 64, \ldots$ words
- Single- and double-precision are incompatible

# Option B Recap

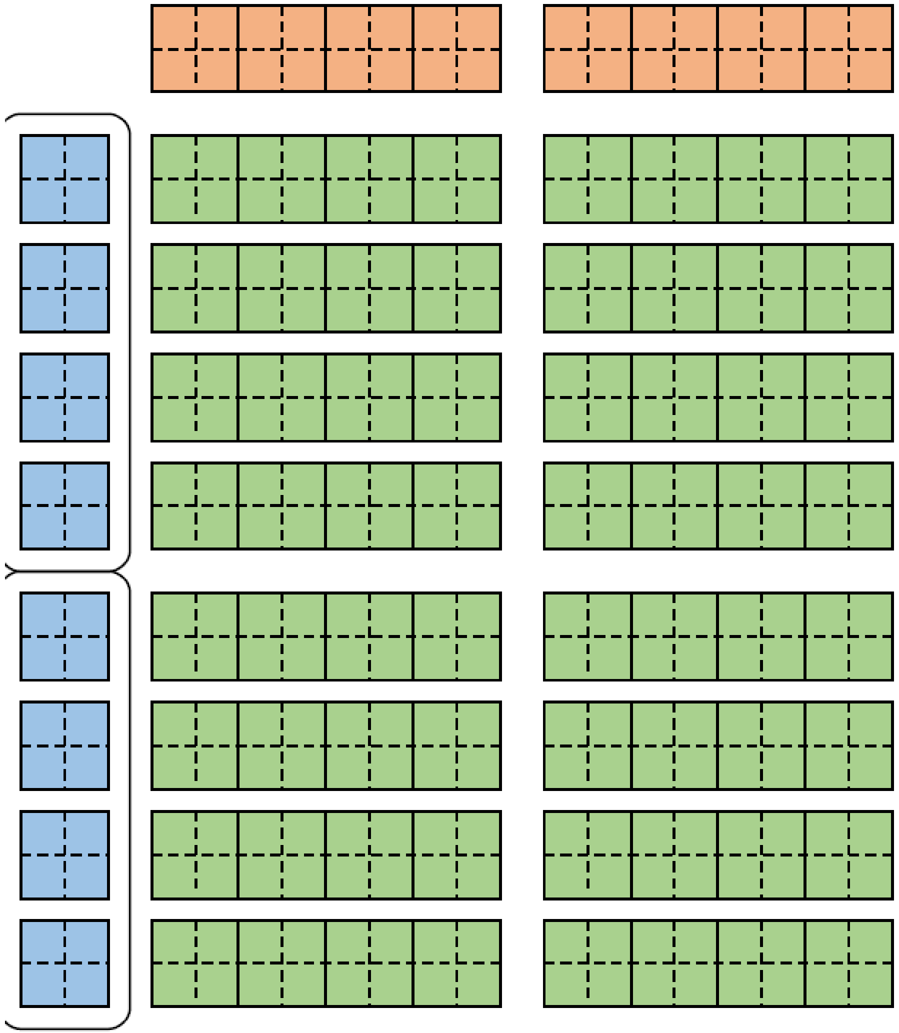
## Option B: 1 Matrix in 4 vector registers




- 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 = $\Delta$ cycles
- Maximum computation rate $R = \frac{16L}{\Delta} = 4L$ madds/cycle
- This is the upper bound with 16 registers for $\boldsymbol{C}$
- Total of $L + 16$ 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 C Recap

## Option C: Matrix as an element type

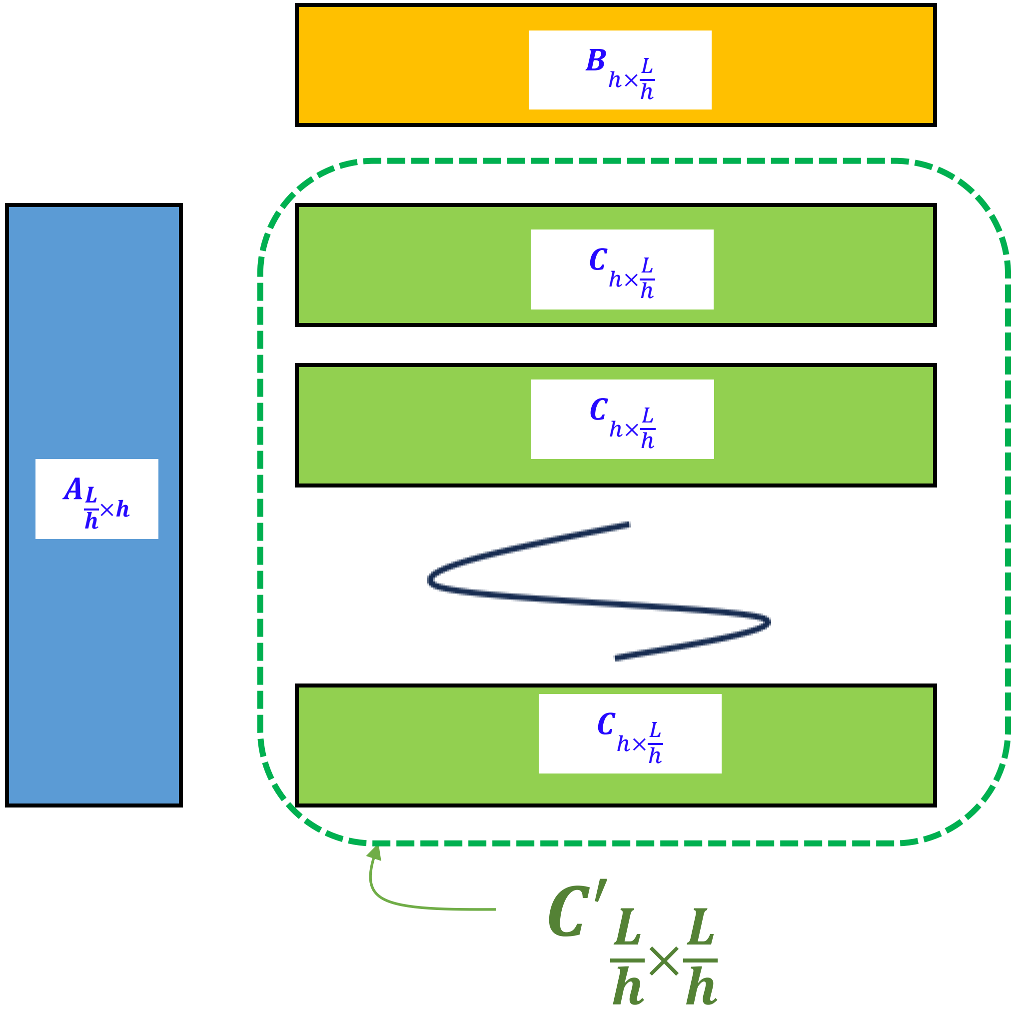


- We fix a $\lambda$ and define a new "vector element type" – a $\lambda \times \lambda$ matrix of words (e.g., $\lambda = 2$)
- A vector register of length $L$ holds $^{L}/_{\lambda^2}$ of these matrices (e.g., $L = 16, ^{L}/_{\lambda^2} = 4$)
- "Some" number of registers hold an $8\lambda \times \lambda$ 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 D Recap

- Introduce 2 stream Source Matrix A and B
- Leverage the same design Matrix C as Option A

# Option E: Architecture State Usages (1/)




① $C_{h\times\frac{L}{h}}$ Single VRF choose h for achieving optimal Architecture State Usages , where h=$\sqrt{L_{min}}$ keeping C efficiently utilized

② $C'_{\frac{L}{h}\times\frac{L}{h}}$ ACC VRFs keep m = n as ACC square for achieving Optimal Computation Intensity

③ Near Optimal Compute Rate :$\widetilde{M}\widetilde{N}\left(\frac{L^2}{4*L_{min}}\right)^{\dagger}$

④ Near Optimal Compute Intensity:$\frac{\widetilde{M}\widetilde{N}\left(\frac{L}{\sqrt{L_{min}}}\right)^2*\sqrt{L_{min}}}{(\widetilde{M}+\widetilde{N})L} = \frac{\widetilde{M}\widetilde{N}L}{\sqrt{L_{min}}(\widetilde{M}+\widetilde{N})}^{\dagger}$

†: where $\widetilde{M},\widetilde{N}$ are reasonable implementation factor (see Appendix for details)

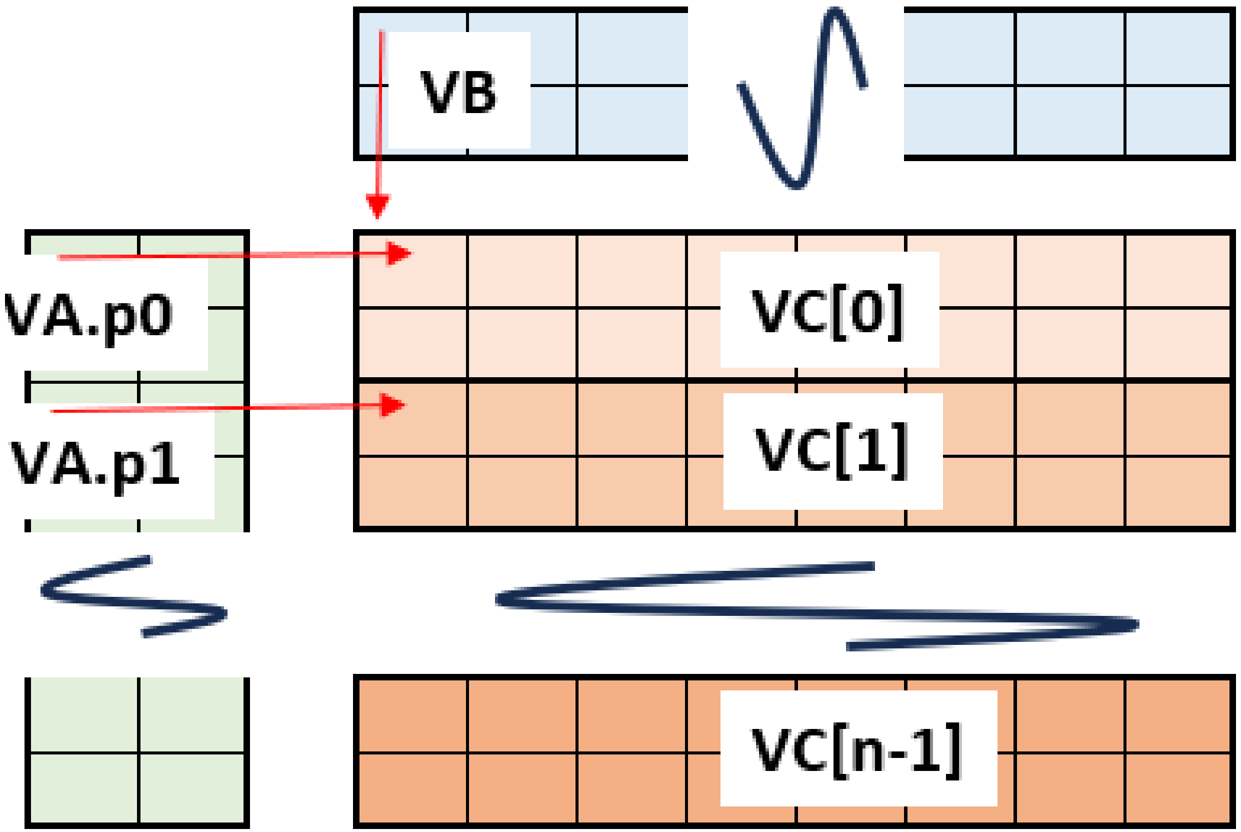
# Option E: Architecture State Usages (2/)

| h | Option | Benefits | Note |
| --- | --- | --- | --- |
| 1 | Close to Option B | | (1) High architecture costs (VRF Register R/W ports) to achieve comparable computation capacity. |
| $\sqrt{L_{min}}$† | Option E | (1) $C_{h\times\frac{L}{h}}$: h is optimal for high VRF Utilization efficiency<br>(2) $C'_{\frac{L}{h}\times\frac{L}{h}}$ : keep square for high compute intensity | |
| $\sqrt{L}$ | Close to Option A | | (1) non-efficient utilization for VRFs when L is not square of integer. E.g. VLEN 256,1024…. |

†: $L_{min}$ is a determined constant as minimal VLEN (assume VLEN 128, L=4 Words), $L_{min}$ = 4 in this presentation.

# Option E: Architecture State Usages (3/)

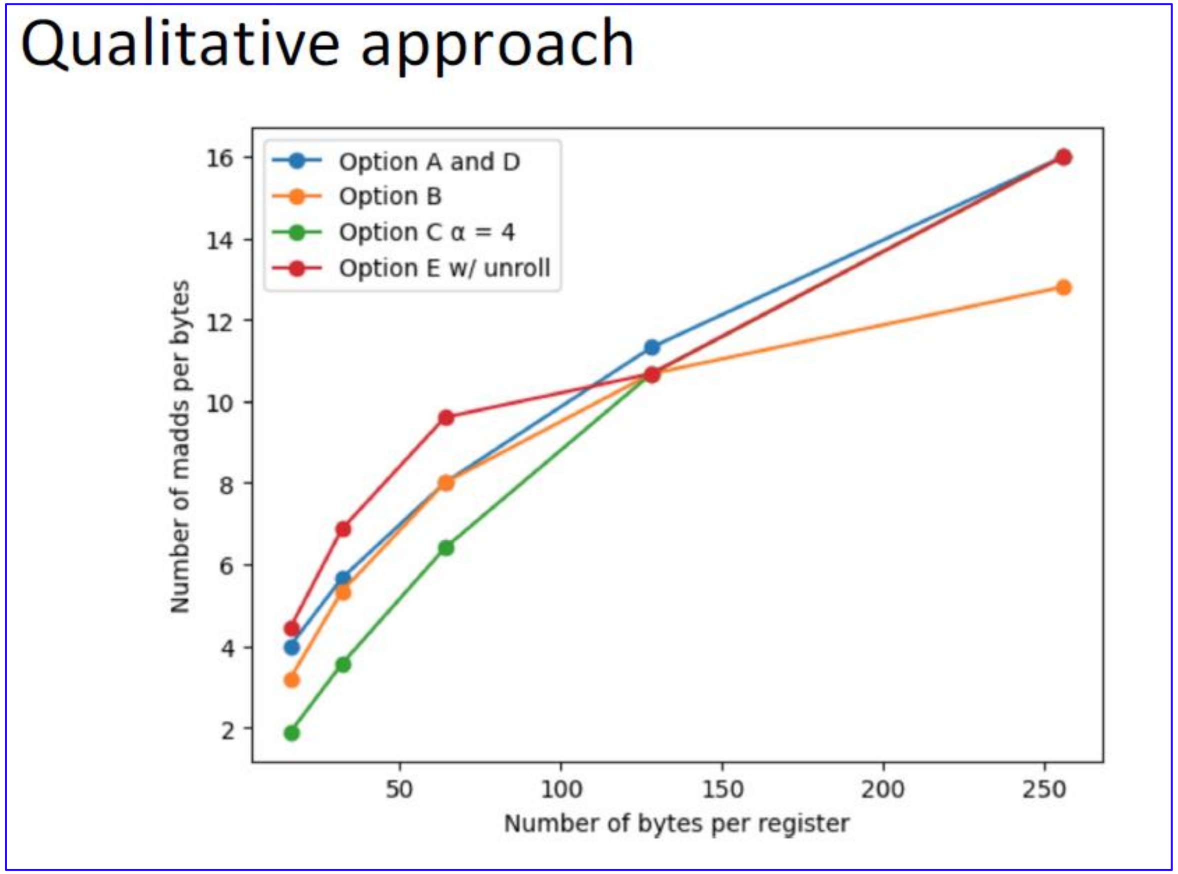
- Scalability Management for Vector Registers
- Efficient Outer Products are Formed with Portion Support for Source Matrix
- amm vd, vb, va, portion†
  - vd = va.px * vb
  - portion can be specified by arguments




†: Illustration of L=16, n= 4 cases of VLEN=512

# 2/12 IME Presentation Data ReCap

- https://drive.google.com/file/d/1RC2RZwX8IJctRgNOlq78xojyPoqiF_q5/view



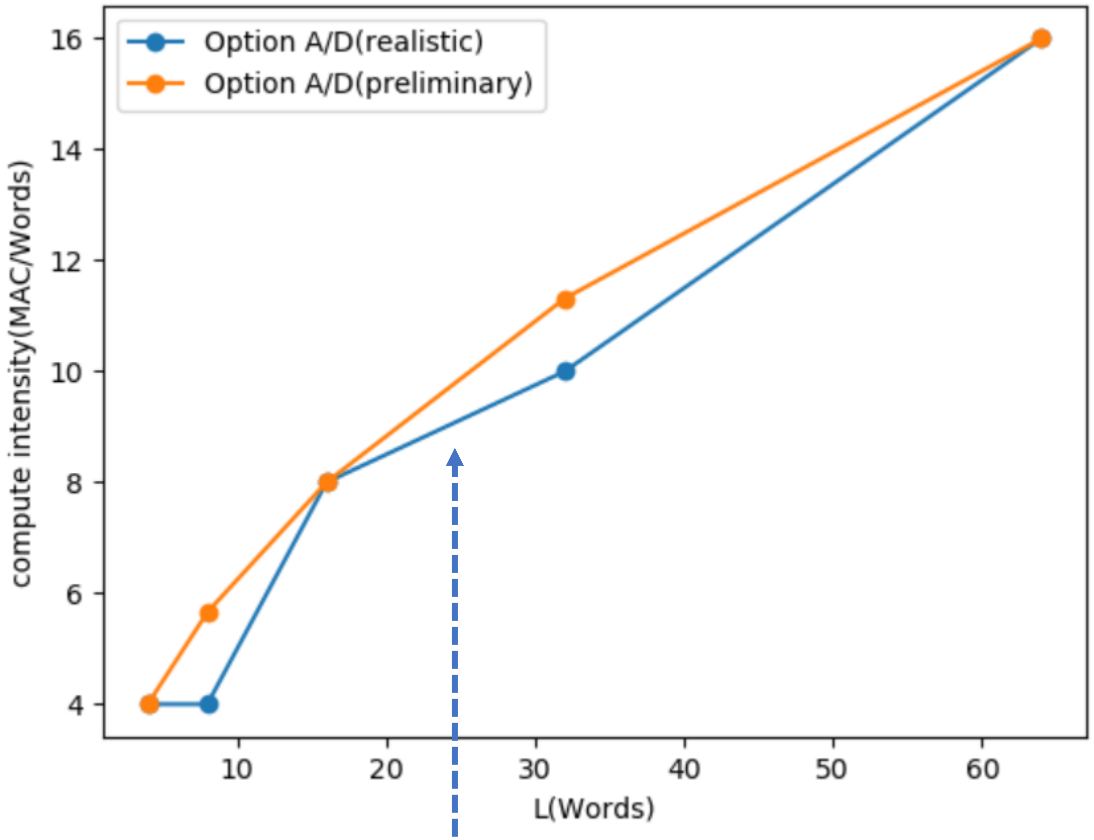

1. Fix I for Option A+D: “realistic” Compute Intensity
2. Fix II for Option C: “realistic” Alpha cross all VLENs
3. Fix III for Option C: Compute Intensity Lemma cross all VLENs
4. Qualitative Comparison Based on
    1) Restricted ACC Register Allocation
    2) Theoretical Optimal Performance

# Analysis Program

- Analysis Program is available on GitHub
  - https://github.com/CN-Ke/IME_Evaluation/tree/main

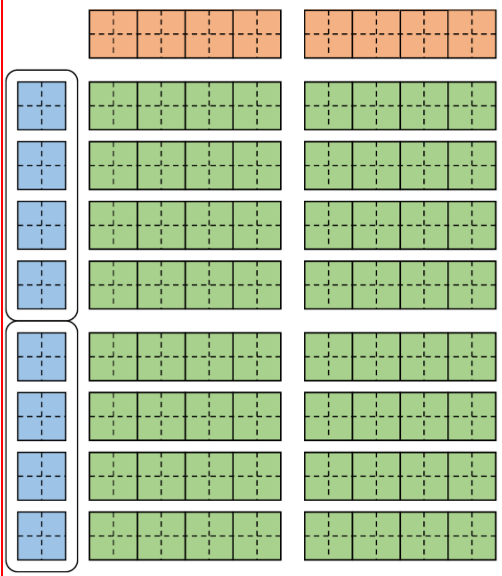
# Fix I for Option A/D Architecture State Utilization Rate

| VLEN | L (Words) | *Option A+D:* $c'_{\lambda x \lambda}$ | | Note |
|---|---|---|---|---|
| | | Preliminary λ | Realistic λ | |
| 128 | 4 | 2 | 2 | |
| 256 | 8 | 2.83 | 2 | (1) Arch. State Utilization Efficiency ~50%<br>(2) data reuse/compute intensity |
| 512 | 16 | 4 | 4 | |
| 1024 | 32 | 5.66 | 5 | (1) Arch. State Utilization Efficiency ~ 78%<br>(2) Poor data reuse/compute intensity |
| 2048 | 64 | 8 | 8 | |

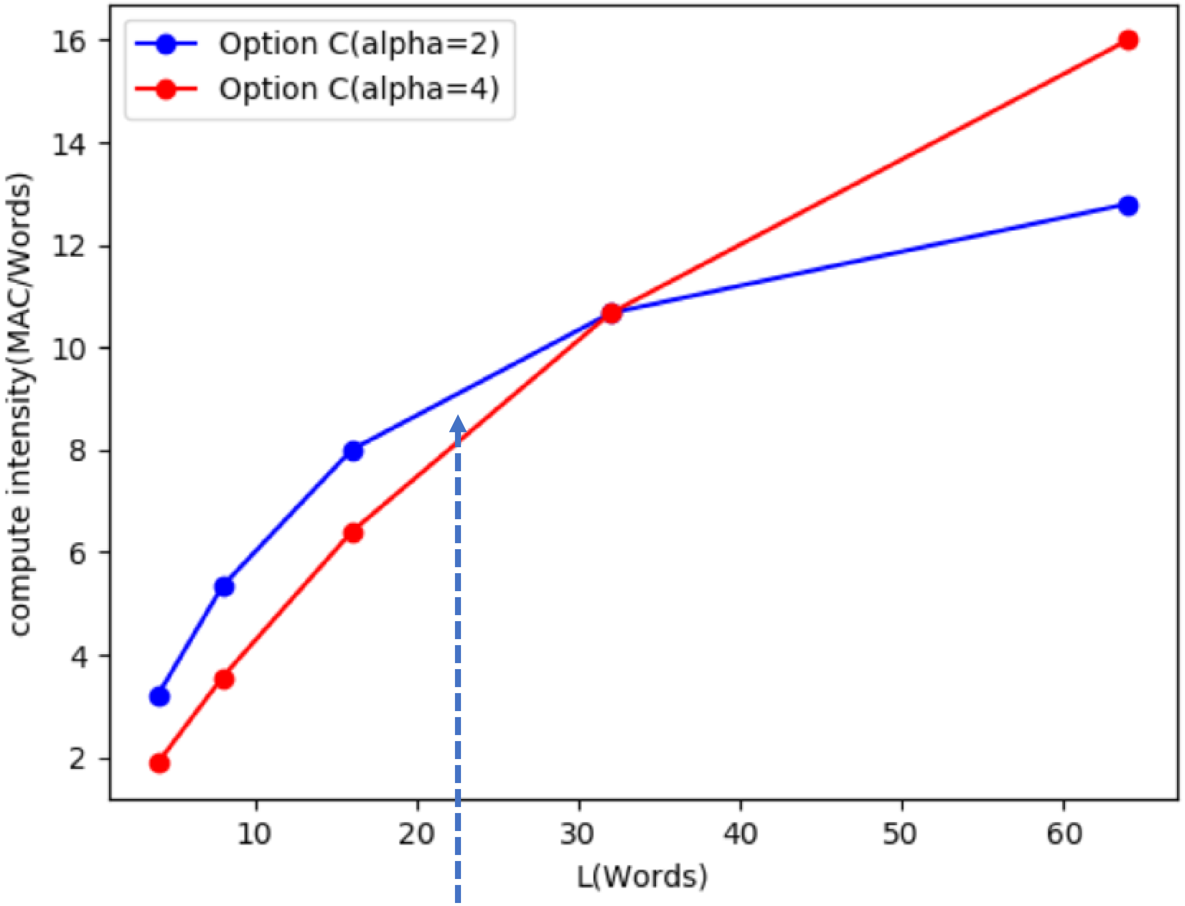



True Compute Intensity Considering Realistic λ (Analysis Program Updated)

# Fix II for Option C "realistic" Alpha cross all VLENs

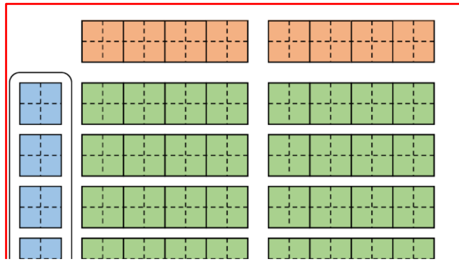


- We fix a $\lambda$ and define a new "vector element type" – a $\lambda \times \lambda$ matrix of words (e.g., $\lambda = 2$)
- A vector register of length $L$ holds ${}^{L}/_{\lambda^2}$ of these matrices (e.g., $L = 16, {}^{L}/_{\lambda^2} = 4$)
- "Some" number of registers hold an $8\lambda \times \lambda$ 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

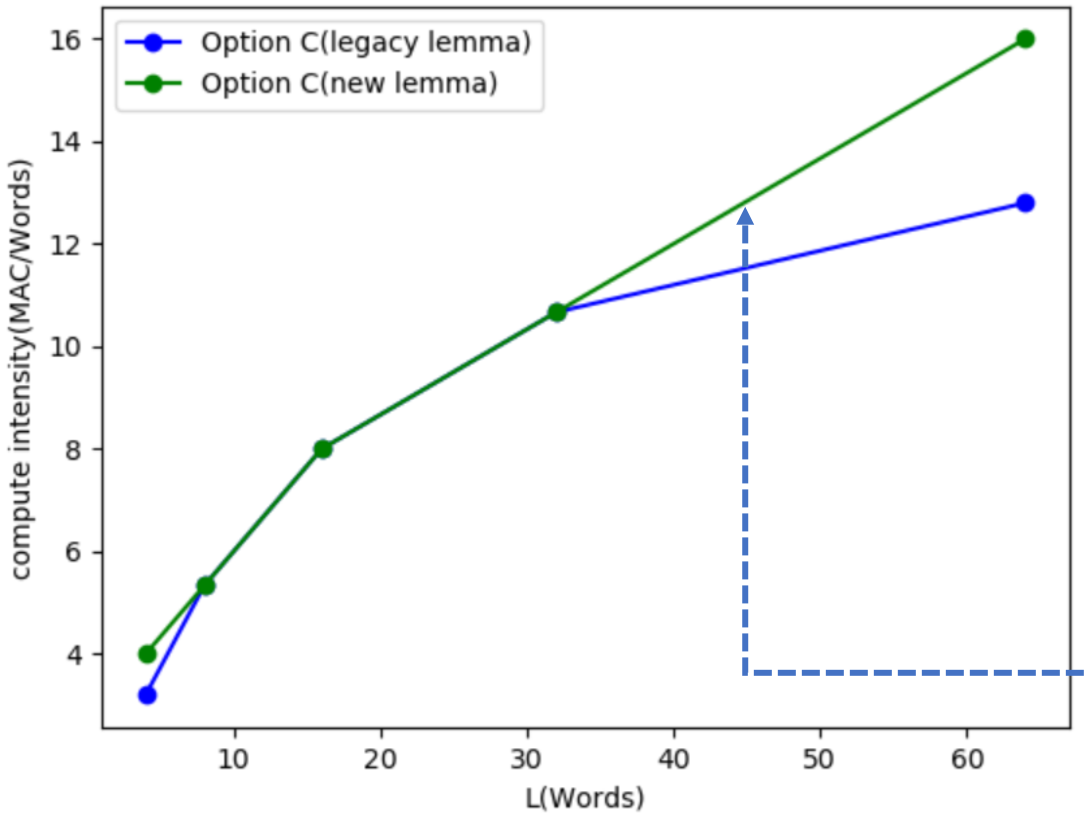



True Compute Intensity Considering Realistic Alpha
(If follow the original lemma shown above)

# Fix III for Option C Lemma cross all VLENs



- 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




| VLEN | L | lambda | L/(lambda$^2$) | SRC-A (VRF#) | A lambda ($\widetilde{M}$) | SRC-B (VRF#) ($\widetilde{N}$) | B lambda | ACC-C (VRF#) | Ultized (VRF#) | compute Intensity |
|---|---|---|---|---|---|---|---|---|---|---|
| 128 | 4 | 2 | 1 | 4 | 4 | 4 | 4 | 16 | 24 | 4.00 |
| 256 | 8 | 2 | 2 | 4 | 8 | 2 | 4 | 16 | 22 | 5.33 |
| 512 | 16 | 2 | 4 | 2 | 8 | 2 | 8 | 16 | 20 | 8.00 |
| 1024 | 32 | 2 | 8 | 1 | 8 | 2 | 16 | 16 | 19 | 10.67 |
| 2048 | 64 | 2 | 16 | 1 | 16 | 1 | 16 | 16 | 18 | 16.00 |

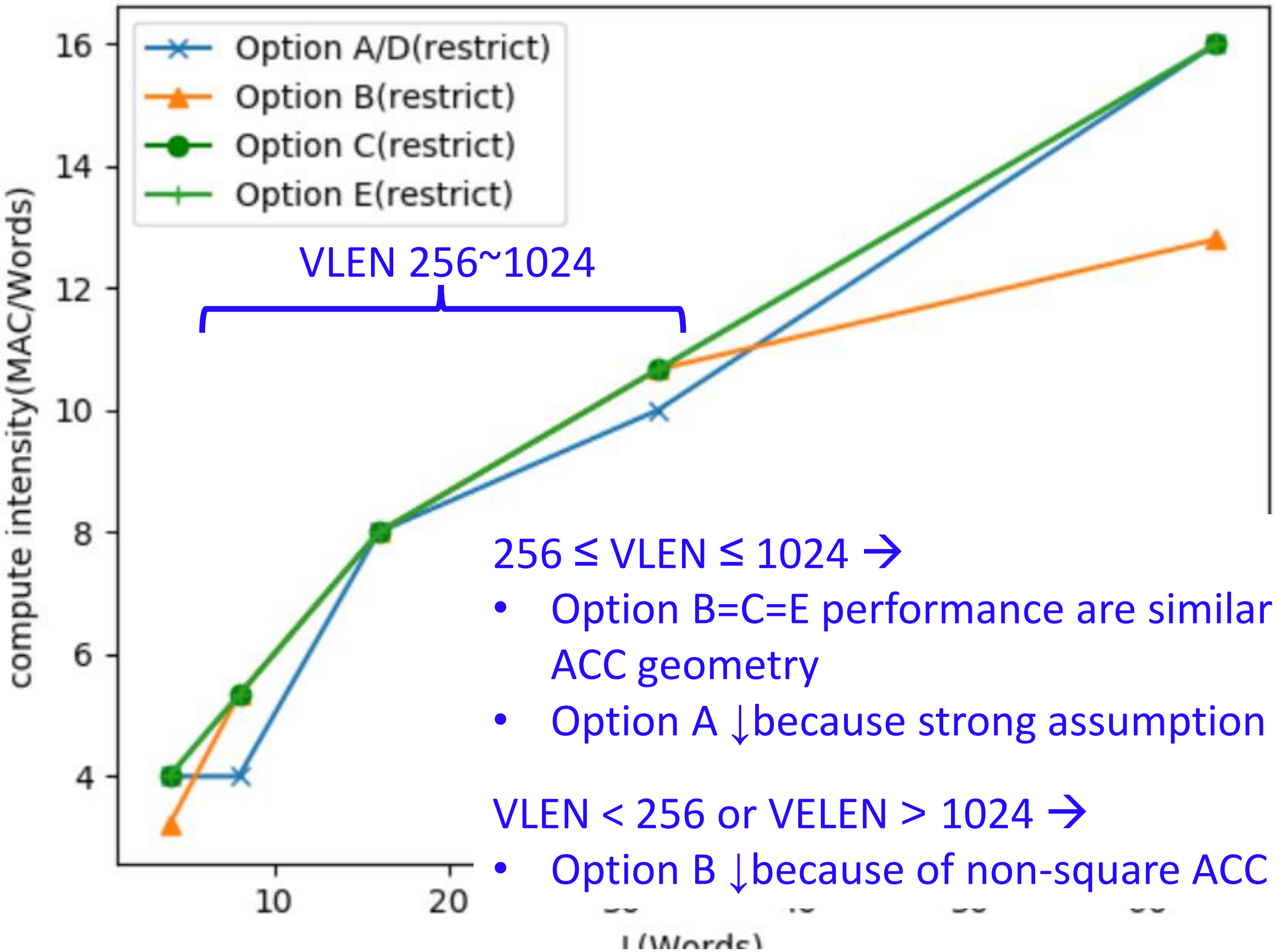
Total $\widetilde{M} * \lambda^2 + \widetilde{N} * L$ Words loaded

Computation Intensity $= \dfrac{\widetilde{M}*\lambda*\widetilde{N}*\left(\frac{L}{\lambda^2}\right)*\lambda^2}{\widetilde{M}*\lambda^2+\widetilde{N}*L} = \dfrac{\widetilde{M}*\widetilde{N}*L*\lambda}{\widetilde{M}*\lambda^2+\widetilde{N}*L}$

Fixed Compute Intensity for new lemma

# Updated Qualitative Comparison

- Based on Restricted ACC VRF by pre-determined half VRFs (16)




256 ≤ VLEN ≤ 1024 →

- Option B=C=E performance are similar because approximated square ACC geometry
- Option A ↓because strong assumption of $L=\lambda^2$

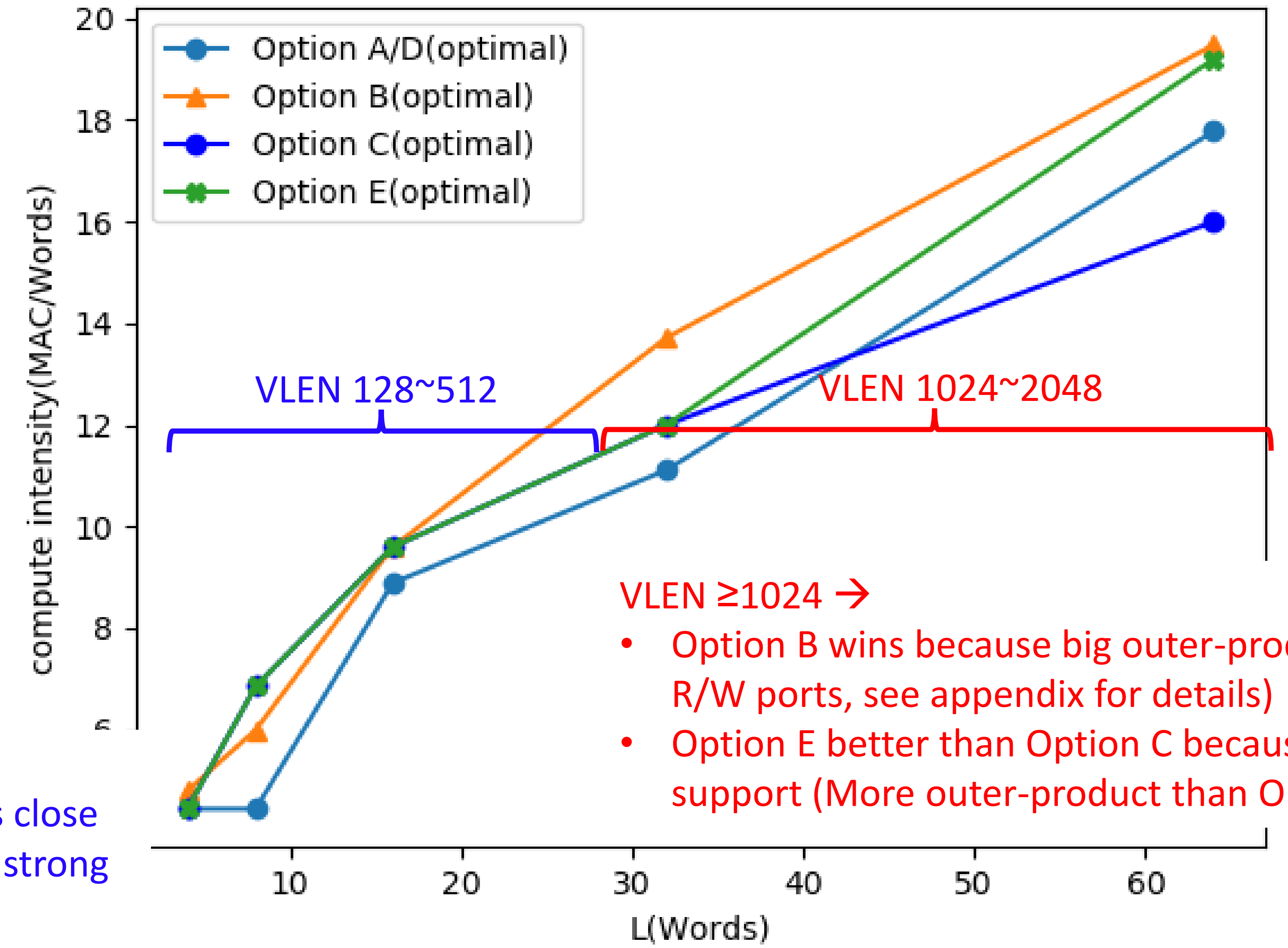
VLEN < 256 or VELEN > 1024 →

- Option B ↓because of non-square ACC gemetry

# Qualitative Comparison for Optimal Performance

- Analysis Based on Reasonable Realizations According to
  - ☐Realistic Alpha/Lambda/h Architecture Parameters
  - ☐General VLENs supported by RVV
  - ☐Feasible VRF Resource $\widetilde{M}, \widetilde{N}$ Unrolling/Grouping

# Qualitative Comparison: Theoretical Performance




VLEN ≥1024 →

- Option B wins because big outer-product (cost of uArch VRF R/W ports, see appendix for details)
- Option E better than Option C because flexibility of portion support (More outer-product than Option C)

VLEN ≤ 512 →

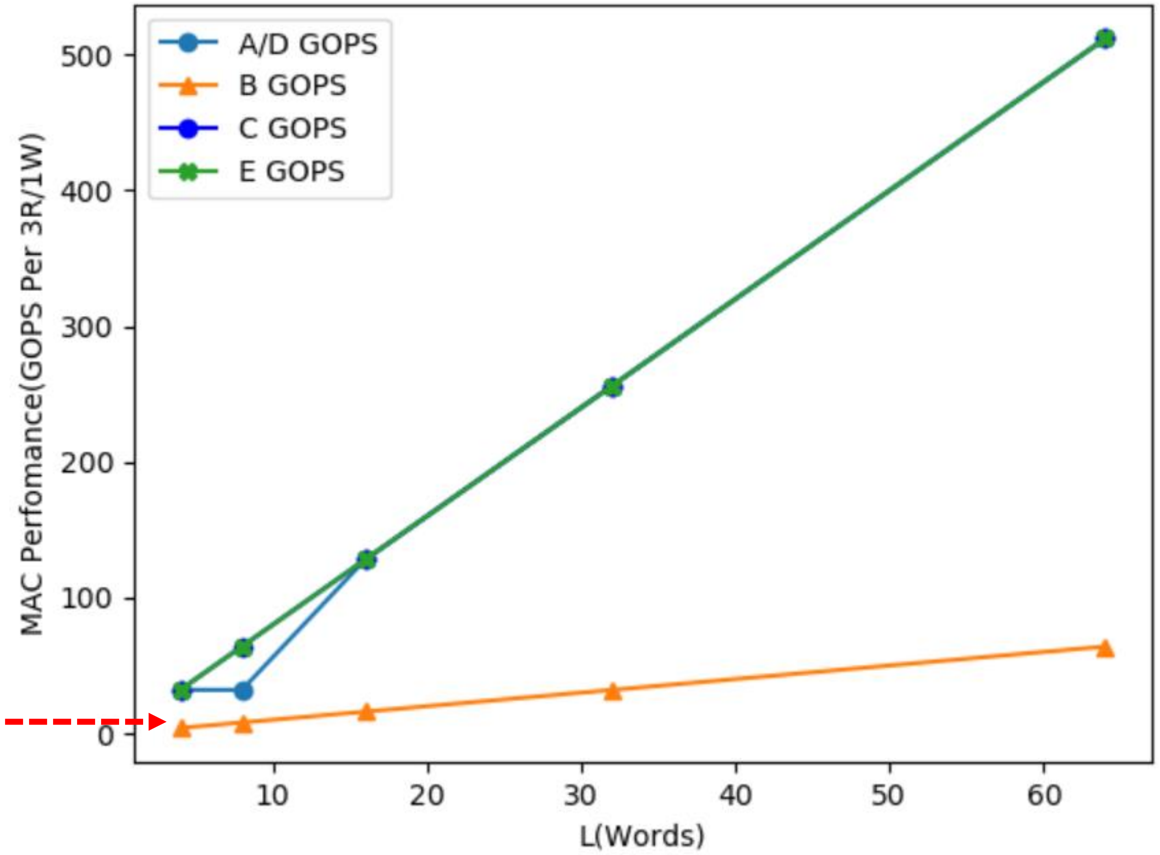
- Option C=E and B is close
- Option A ↓because strong assumption of $L=\lambda^2$

# Metrics for MAC Performance/uArch Cost

- Consider General Scenario for Int8 Convolutions Performance for
  - ① baseline uArch resources(restriction) : 3R/1W VRF Port
  - ② reasonable Int8 MAC bound ~ 8*L †
- All options' GOPS Performance can be derived as

| Option | A/D | B | C | E |
|---|---|---|---|---|
| MAC/Cycle | $4*\lambda*\lambda^2$ $=4*\lambda^3$ | L | $4\cdot\lambda\cdot\lambda\cdot\lambda\cdot(\frac{L}{\lambda^2})$ $=4*\lambda*L$ | $4\cdot h\cdot h\cdot(\frac{L}{h})$ $=4*h*L$ |

Option B Performance Explicitly restricted by uArch Resources (E.g. RF Ports)




†: 8L from current technology feasible realization (Eg. General Core/VPU at 1Ghz)

# Charter Criteria Discussion

- Based on proposing Matrix-TG → 9 explicitly guide-lines
- Suggest to add Metrics for Performance/uArch Cost

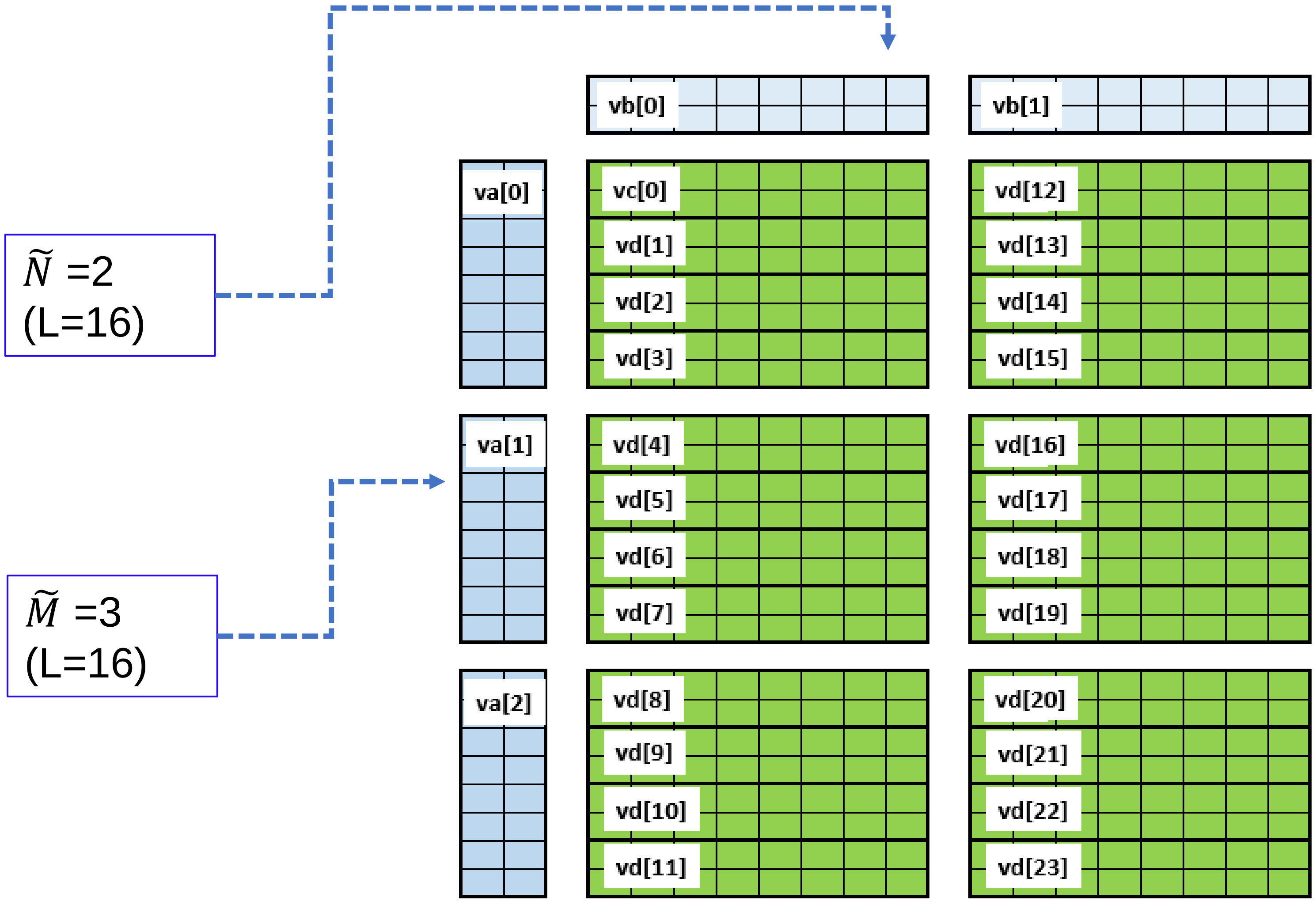
| No. | Guides | Option A | Option B | Option C | Option D | Option E |
|---|---|---|---|---|---|---|
| 1 | **VLEN agnostic** at binary level | Not Disclosed while L!=$\lambda^2$ | To Be Discussed (seems support?) | | Not Disclosed while L!=$\lambda^2$ | (1) Source Compliant<br>(2) Binary Compliant for vmul/fused ISA |
| 2 | **Deterministic Result** (FMAC rounding/ordering) | Shall support | Shall support | Shall support | Shall support | Support if finalized |
| 3 | **Re-producible result** with plain scalar/vector | Shall support | Shall support | Shall support | Shall support | Support (BIT TRUE test) |
| 4 | **Near peak (~90%) performance** for GEMM kernels is possible | | | | | GEMM kernels Near Peak U-rate |
| 5 | **Higher (~2X) performance** than vector | | | | | Over (>3X) enhancement than RVV |
| 6 | Maximization **computation intensity** for GEMM kernels | | | | | GEMM kernels Near Peak U-rate |
| 7 | Minimization additional **architecture state** | None new state | None new state | None new state | Not Support (New Streaming buffer for A/B) | None new state (not considering ZOB) |
| 8 | **Live-migration** with larger vector registers | Not Support for different lambda | To Be Discussed (seems support?) | Not Support for different vector element type | Not Support for different lambda | Under Working (AMM 2.0) |
| 9 | Proper Support for **packing/reformat data** | May Need Additional handling while L!=$\lambda^2$ | Support (No additional interleaving/shuffle required) | Support (No additional interleaving/shuffle required) | May Need Additional handling while L!=$\lambda^2$ | Support (No additional interleaving/shuffle required) |
| 10 | **Metrics for Performance/uArch cost (Suggest to consider)** | Feasible uArch cost(VRF R/W) for Specific MAC Performance | High uArch cost(VRF R/W) for MAC Performance† | feasible uArch cost(VRF R/W) for MAC Performance | Feasible uArch cost(VRF R/W) for MAC Performance | Feasible uArch cost(VRF R/W) for MAC Performance |

†: refer details at

# Appendix

# Option E: Theoretical Performance Revisit

- $\widetilde{M}, \widetilde{N}$ Denotes Reasonable Unrolling Parameters

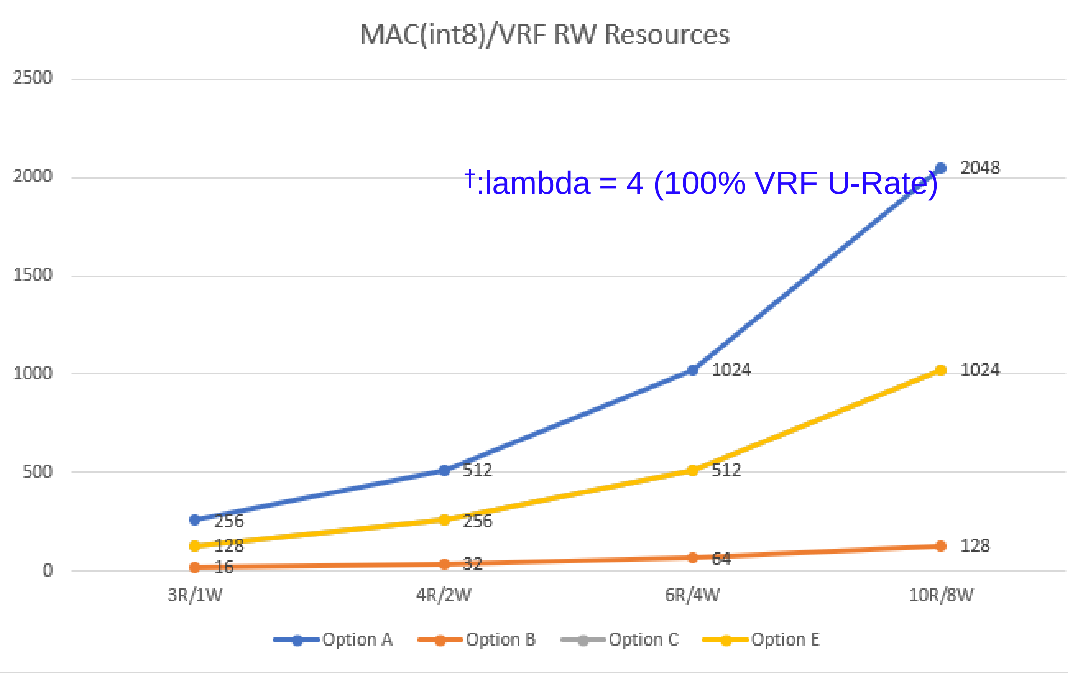


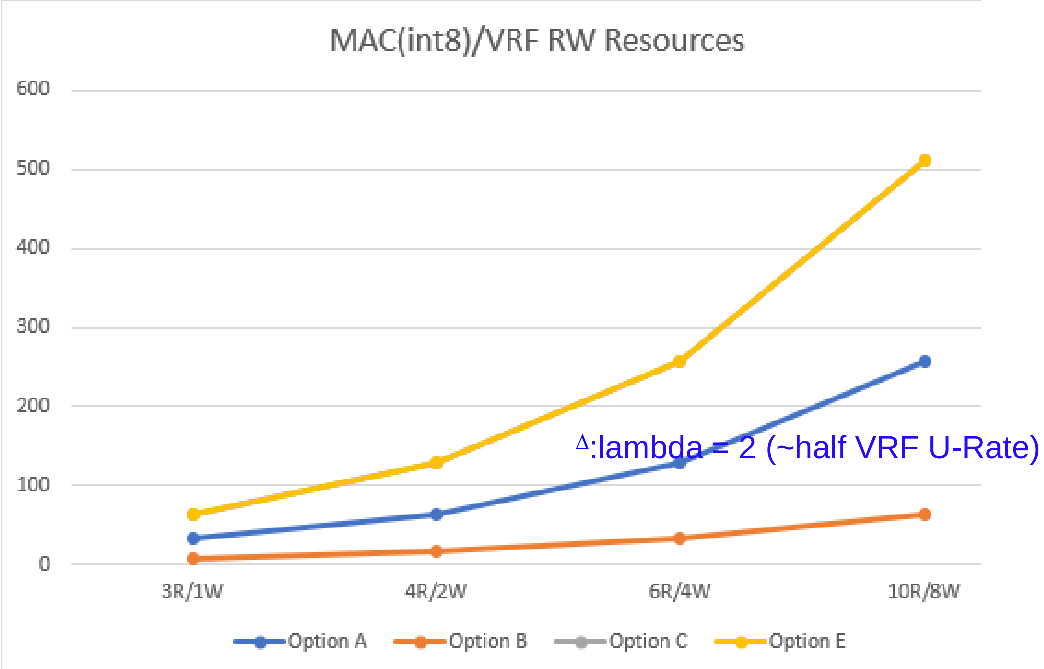
# Metrics for MAC Performance vs. uArch Cost

- Single Instruction Required VRF R/W Resources (Cycle) for specific MAC Performance

| VRF RW Ports | Option A † | Option B | Option C | Option D | Option E | Note |
|---|---|---|---|---|---|---|
| 3R/1W | 256 | 16 | 128 | additional Streaming buffer R/W Port Required | 128 | References VLEN 512,int8,128MAC/cycle |
| 4R/2W | 512 | 32 | 256 | | 256 | |
| 6R/4W | 1024 | 64 | 512 | | 512 | |
| 10R/8W | 2048 | 128 | 1024 | | 1024 | |

| VRF RW Ports | Option A Δ | Option B | Option C | Option D | Option E | Note |
|---|---|---|---|---|---|---|
| 3R/1W | 32 | 8 | 64 | additional Streaming buffer R/W Port Required | 64 | References VLEN 256,int8,64MAC/cycle |
| 4R/2W | 64 | 16 | 128 | | 128 | |
| 6R/4W | 128 | 32 | 256 | | 256 | |
| 10R/8W | 256 | 64 | 512 | | 512 | |



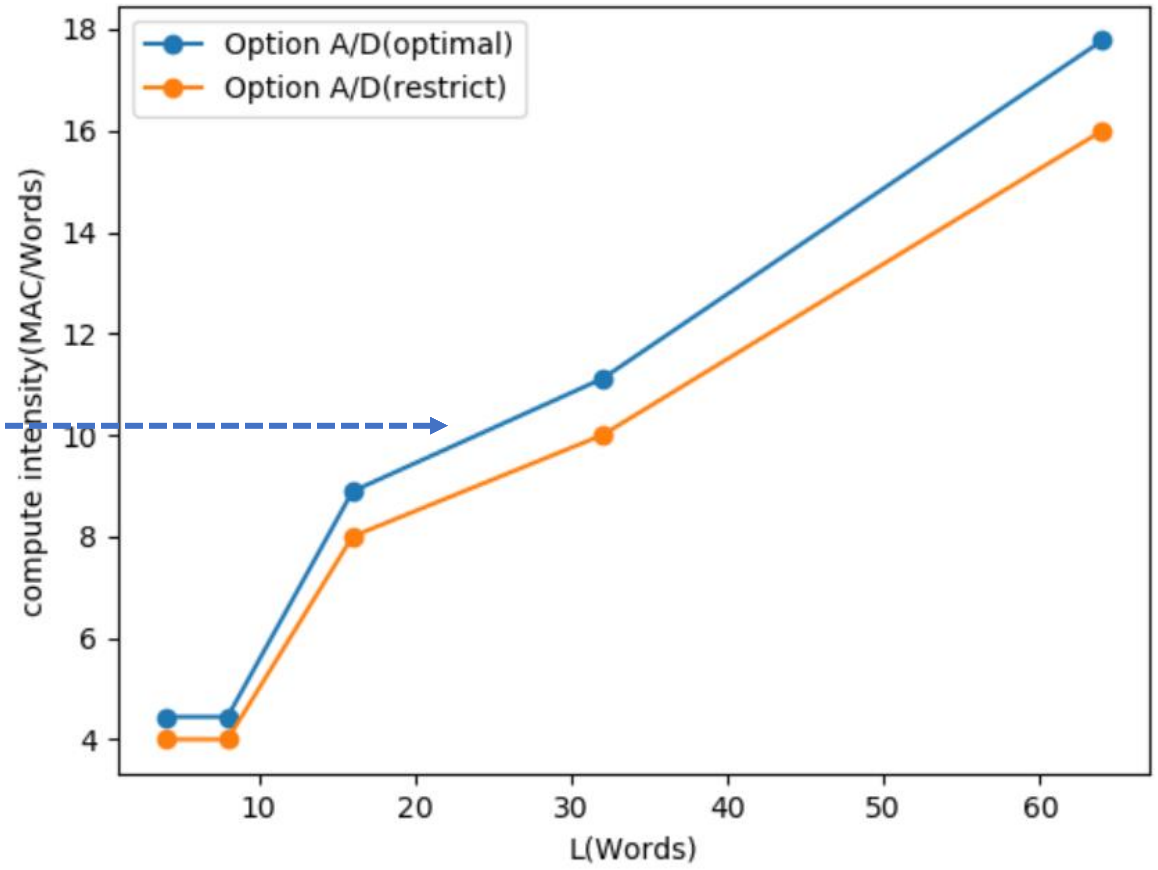



# Option A: Optimal Performance Revisit

| SRC-A (VRF#) $\widetilde{M}$ | SRC-B (VRF#) $\widetilde{N}$ | ACC-C (VRF#) | Ultized (VRF#) | lamda† | compute Intensity |
|---|---|---|---|---|---|
| 1 | 15 | 15 | 31 | 4 | 3.75 |
| 2 | 10 | 20 | 32 | 4 | 6.67 |
| 3 | 7 | 21 | 31 | 4 | 8.40 |
| 4 | 5 | 20 | 29 | 4 | 8.89 |
| 5 | 4 | 20 | 29 | 4 | 8.89 |
| 6 | 3 | 18 | 27 | 4 | 8.00 |
| 7 | 3 | 21 | 31 | 4 | 8.40 |
| 8 | 2 | 16 | 26 | 4 | 6.40 |
| 9 | 2 | 18 | 29 | 4 | 6.55 |
| 10 | 2 | 20 | 32 | 4 | 6.67 |
| 11 | 1 | 11 | 23 | 4 | 3.67 |
| 12 | 1 | 12 | 25 | 4 | 3.69 |
| 13 | 1 | 13 | 27 | 4 | 3.71 |
| 14 | 1 | 14 | 29 | 4 | 3.73 |
| 15 | 1 | 15 | 31 | 4 | 3.75 |

†: Illustration by lambda = 4 where VLEN = 512

- Total $(\widetilde{M} + \widetilde{N}) * \lambda^2$ Words loaded
- Computation Intensity $= \frac{(\widetilde{M} * \widetilde{N}) * \lambda^3}{(\widetilde{M} + \widetilde{N}) * \lambda^2} = \frac{(\widetilde{M} * \widetilde{N}) * \lambda}{(\widetilde{M} + \widetilde{N})}$



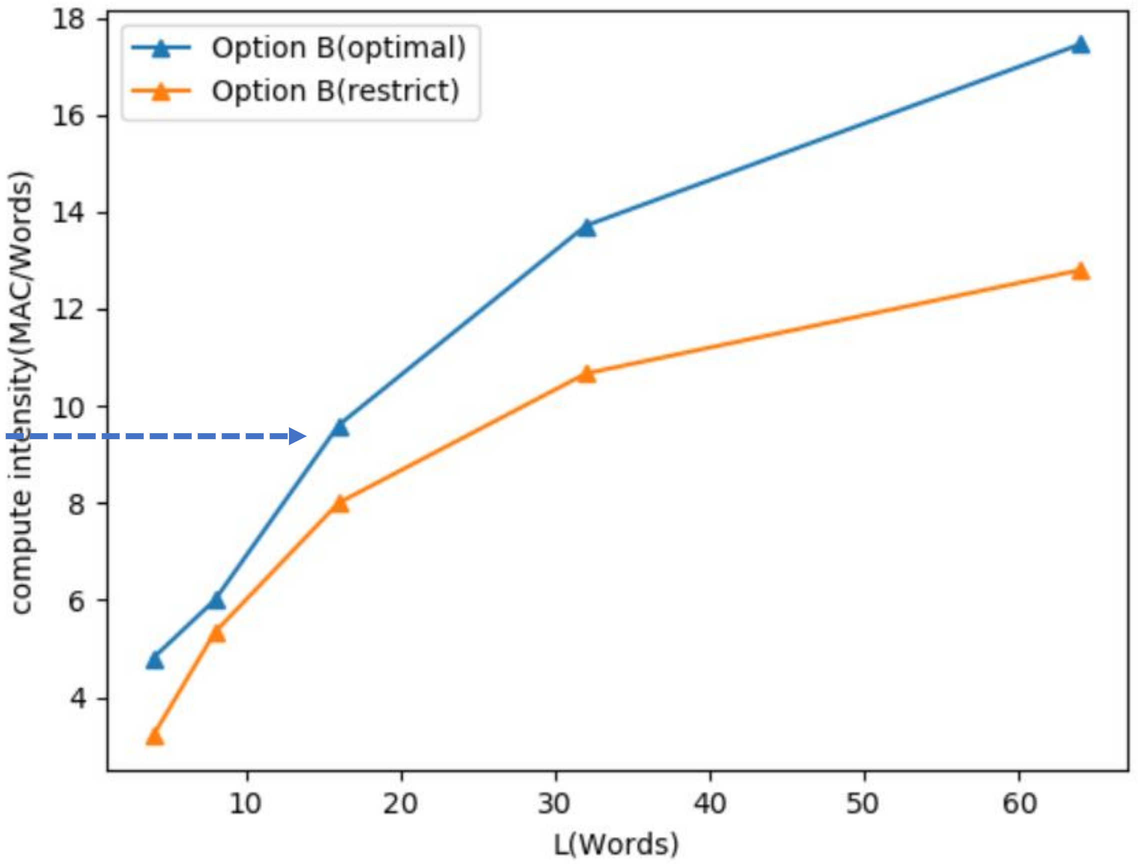

Theoretical Performance Could be Achieved

# Option B: Optimal Performance Revisit

| VLEN | L | SRC-A (VRF#) | $\widetilde{A_{4x1}}$<br>$A_{4x1}$ (Words) | $\widetilde{N}$<br>SRC-B (VRF#) | ACC-C (VRF#) | Ultized (VRF#) | compute Intensity |
|---|---|---|---|---|---|---|---|
| 128 | 4 | 3 | 12 | 2 | 24 | 29 | 4.80 |
| 256 | 8 | 3 | 24 | 1 | 24 | 28 | 6.00 |
| 512 | 16 | 1.5 | 24 | 1 | 24 | 26.5 | 9.60 |
| 1024 | 32 | 0.75 | 24 | 1 | 24 | 25.75 | 13.71 |
| 2048 | 64 | 0.375 | 24 | 1 | 24 | 25.375 | 17.45 |

- Total $\widetilde{A_{4x1}} + \widetilde{N} * L$ Words loaded
- Computation Intensity $= \frac{(\widetilde{A_{4x1}} * \widetilde{N}) * L}{\widetilde{A_{4x1}} + \widetilde{N} * L}$

Theoretical Performance Could be Achieved

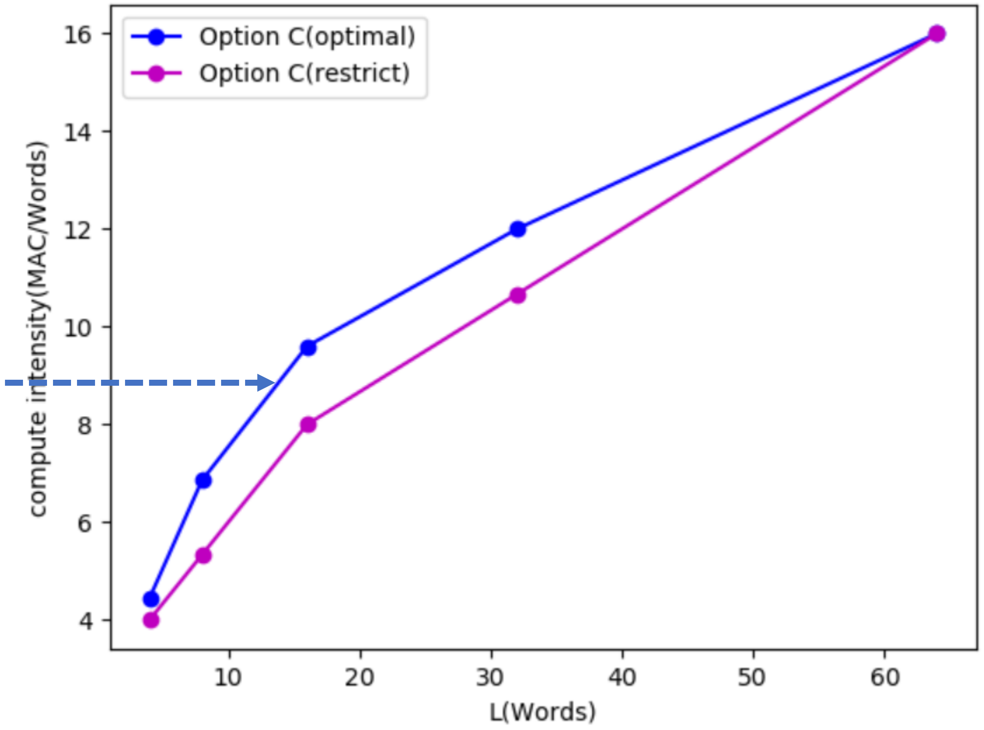



- Refer Metrics for $\frac{MAC\ Performace}{uArch\ cost}$

# Option C: Theoretical Performance Revisit

| | | | | | $\widetilde{M}$ | $\widetilde{N}$ | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **VLEN** | **L** | **lambda** | **L/(lambda$^2$)** | **SRC-A (VRF#)** | **A lambda** | **SRC-B (VRF#)** | **B lambda** | **ACC-C (VRF#)** | **Ultized (VRF#)** | **compute Intensity** |
| 128 | 4 | 2 | 1 | 5 | 5 | 4 | 4 | 20 | 29 | 4.44 |
| 256 | 8 | 2 | 2 | 3 | 6 | 4 | 8 | 24 | 31 | 6.86 |
| 512 | 16 | 2 | 4 | 2 | 8 | 3 | 12 | 24 | 29 | 9.60 |
| 1024 | 32 | 2 | 8 | 1 | 8 | 3 | 24 | 24 | 28 | 12.00 |
| 2048 | 64 | 2 | 16 | 1 | 16 | 1 | 16 | 16 | 18 | 16.00 |

- Total $\widetilde{M} * \lambda^2 + \widetilde{N} * L$ Words loaded
- Computation Intensity $= \frac{\widetilde{M}*\lambda*\widetilde{N}*\left(\frac{L}{\lambda^2}\right)*\lambda^2}{\widetilde{M}*\lambda^2+\widetilde{N}*L} = \frac{\widetilde{M}*\widetilde{N}*L*\lambda}{\widetilde{M}*\lambda^2+\widetilde{N}*L}$




Theoretical Performance Could be Achieved

# Option E: Theoretical Performance Revisit

- Total $(\widetilde{M} + \widetilde{N}) * L$ Words loaded
- Computation Intensity = $\frac{\widetilde{M}\widetilde{N}\left(\frac{L}{\sqrt{L_{min}}}\right)^2 * \sqrt{L_{min}}}{(\widetilde{M}+\widetilde{N})L} = \frac{\widetilde{M}\widetilde{N}L}{\sqrt{L_{min}}(\widetilde{M}+\widetilde{N})}$ †

| VLEN | L | $\sqrt{L_{min}}$ | portion | SRC-A (VRF#) $\widetilde{M}$ | SRC-A* portion | SRC-B (VRF#) $\widetilde{N}$ | ACC-C (VRF#) | Ultized (VRF#) | compute Intensity |
|---|---|---|---|---|---|---|---|---|---|
| 128 | 4 | 2 | 1 | 5 | 5 | 4 | 20 | 29 | 4.44 |
| 256 | 8 | 2 | 2 | 4 | 8 | 3 | 24 | 31 | 6.86 |
| 512 | 16 | 2 | 4 | 3 | 12 | 2 | 24 | 29 | 9.60 |
| 1024 | 32 | 2 | 8 | 3 | 24 | 1 | 24 | 28 | 12.00 |
| 2048 | 64 | 2 | 16 | 1.5 | 24 | 1 | 24 | 26.5 | 19.20 |

Theoretical Performance Could be Achieved
See Reference in Appendex