Exploring Matrix Multiplication Techniques and Enhancements Using RISC-V Custom Extension

Chun-Nan Ke
Sr. Tech. Manager
Andes Technology

RISC-V® Summit

Agenda

- Qualitative Comparison of Matrix Instructions
- Challenges of RVV on Matrix Operations
- SW Scalability/Portability Support
- Efficient Tiling using VRF Load/Stores
- Preliminary Results
- Conclusion

Qualitative Comparison for Matrix Instructions

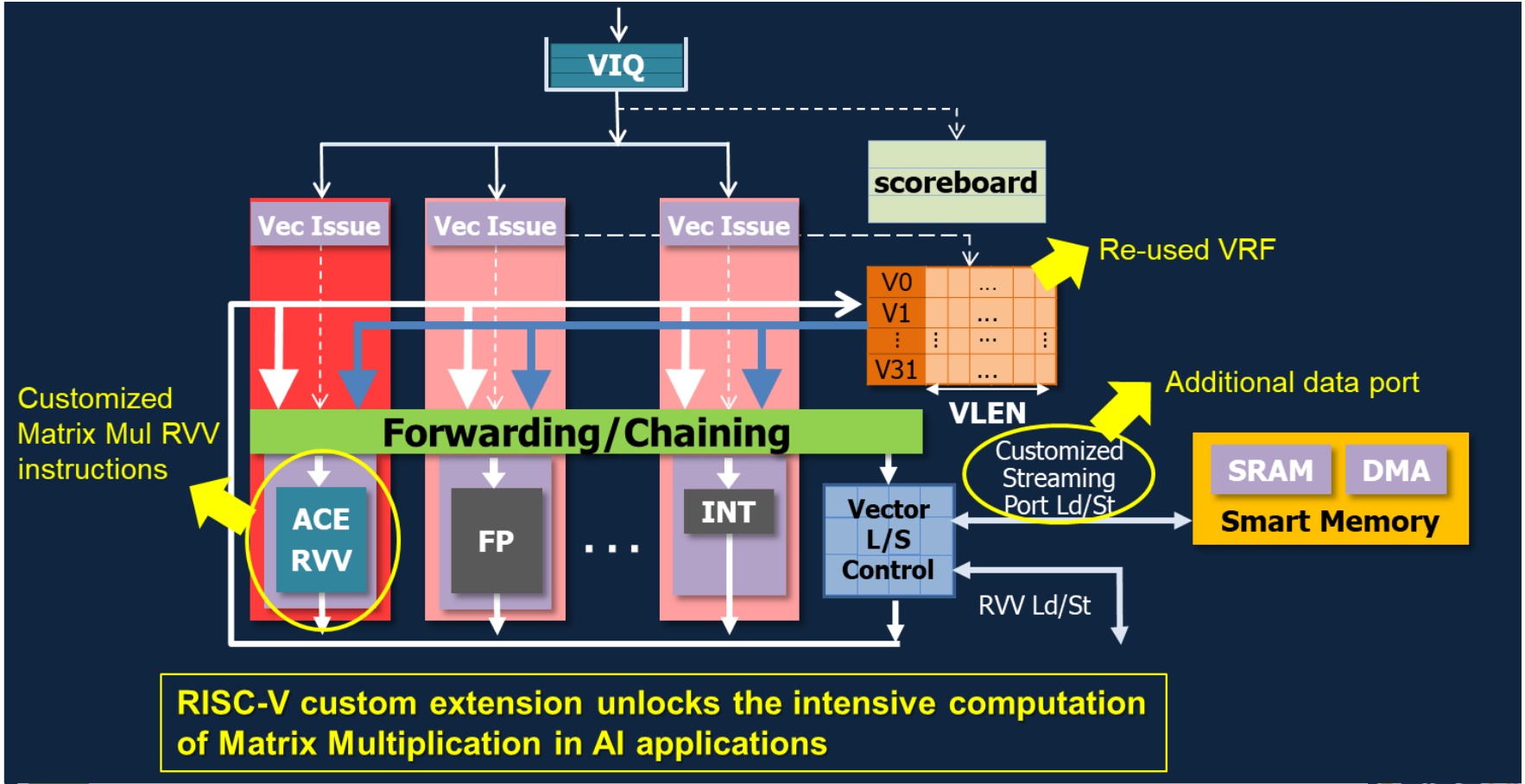
Proposal	Performance	Power	Area	Program Model
	• FU Utilization-rate • Memory Bandwidth • Diverse Metrics	• Context Switch Overhead • Data Exchange Efficiency	• RF Storage • R/W port	• Scalability • VLEN Agnostic
Integrated VRF‡ Today's focus	✓ Efficient IO scheme ✓ High Utilization-rate ✓ High Compute-Mem Ratio	✓ Low Dissipation	✓ Low	✓ Yes
Hybrid△ Watching	✓ Seamless Scalable ✓ Outer-Product Support ✓ Transpose/Column Access	• Medium	• Medium	✓ Yes
Attached Facility† Watching	✓ Better Physical Implementation ✓ Data Compression support	• Extra Dissipation	• High	✓ Yes

‡:Reuse Vector Register: Andes AMM, SiFive Intelligence (not disclosed)
△:Additional Matrix Register: ARM SME
†:Independent Matrix Register: T-head, Streaming Computing

Challenges of RVV on Matrix Operations

- Computational Capacity Challenges
  - Constrained computing power based on vector-product/reduced-sum
- I/O Efficiency of Tiling VRF
  - load/store instruction struggles for forming matrix tiles
  - permute/slide instruction overheads for reshaping
- Data Reuse/Locality
  - Inner product exhibits poor memory bandwidth requirements
- SW Scalability
  - Tiling/widening present porting difficulties cross diverse VLENs
- Boundary/Tail Handling
  - Legacy vl csr is inadequate for managing matrix/tensor boundaries

Andes Custom Extension (ACE) on RVV



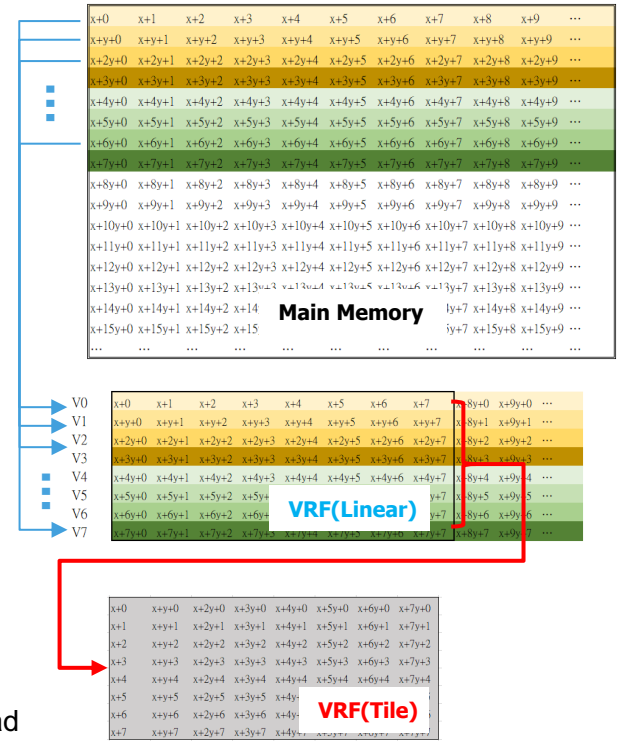
I/O Efficiency of Tiling VRF

- Leverage Standard RVV instructions:
  - load/store:
    - Linear load/store → vl/se<eew>.v
    - Constant stride load/store → vl/s**s**e<eew>.v
    - Segment load → vl/sseg<nf>e<eew>.v
  - permutation:
    - Vector slide(up/down) → vslideup/down.v

RVV Overheads of Forming Matrix Tile(1)

- Pseudo Code demonstration for linear-load + Slide to form an 8x8 INT8 Matrix Tile
  - Take Example of VLEN 512, Feature Map/Weight format INT8

```
;linear vload to 8 VRFs
vle8.v v0,(base),vl
vle8.v v1,(base+K),vl
vle8.v v2,(base+K*2),vl
vle8.v v3,(base+K*3),vl
vle8.v v4,(base+K*4),vl
vle8.v v5,(base+K*5),vl
vle8.v v6,(base+K*6),vl
vle8.v v7,(base+K*7),vl
;vslideup/down to merge vt in multi-cycle permutations
vslideup.vx va,v0,0,vm ;va[i]=v0[i]
vslideup.vx va,v1,8,vm ;va[i+8]=v1[i]
vslideup.vx va,v2,16,vm ;va[i+16]=v2[i]
vslideup.vx va,v3,24,vm ;va[i+24]=v3[i]
vslideup.vx va,v4,32,vm ;va[i+32]=v4[i]
vslideup.vx va,v5,40,vm ;va[i+40]=v5[i]
vslideup.vx va,v6,48,vm ;va[i+48]=v6[i]
vslideup.vx va,v7,56,vm ;va[i+56]=v7[i]
```



(1) ***x***T↑(L2$hit) + 8****y***T↑slide (multi-cycle perm.) ~ **at least > (x+8y)T** for 1st va load

(2) 2nd va load: 8****y***T↑slide (multi-cycle perm.) ~ **8yT** for 2nd va permutation

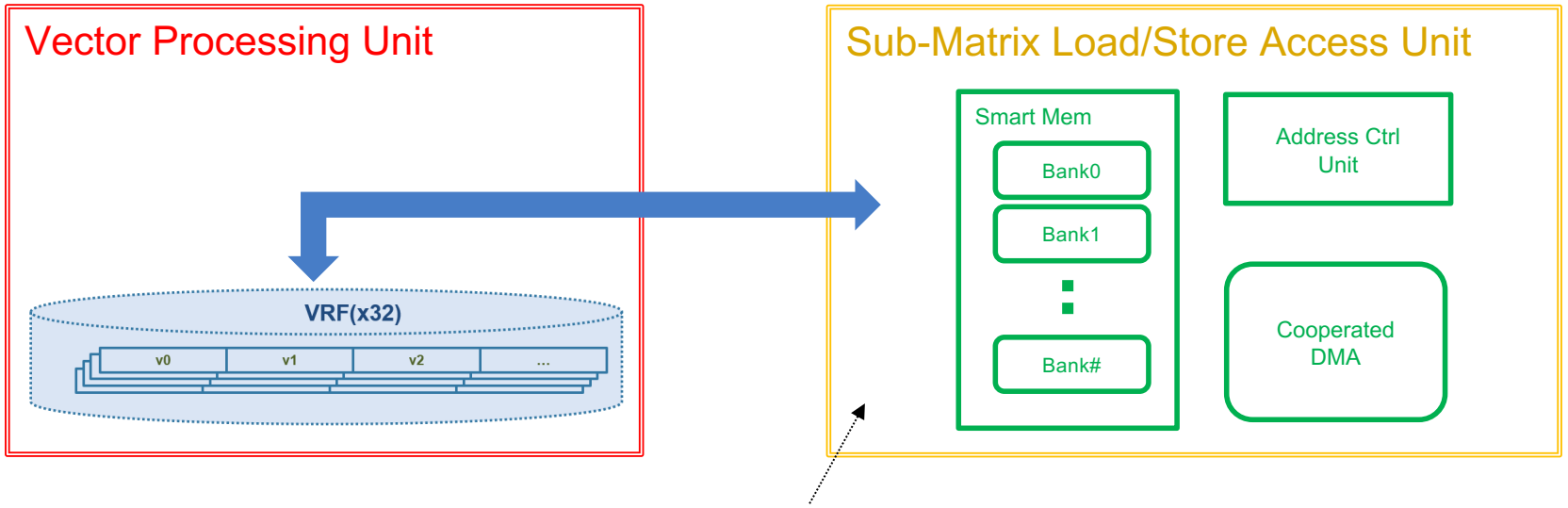
RVV Overheads of Forming Matrix Tile(2)

- Example latencies of forming sub-matrix tiles using RVV
  - Assume VLEN 512 and Input Feature Map is in INT8

Tiling Shape	1st vload	Seq. vload (permutation)	U-rate (GeMM 128x128)
2x32	>9*T	>6*T	
8x8 (Linear w/o Transpose)	>27*T	>24*T	< 22%

*Note: Private L2 cache hit latency takes 3 cycles
*Note: Permutation Instruction takes 3 cycles

Andes Custom Streaming Port (ASP) on RVV

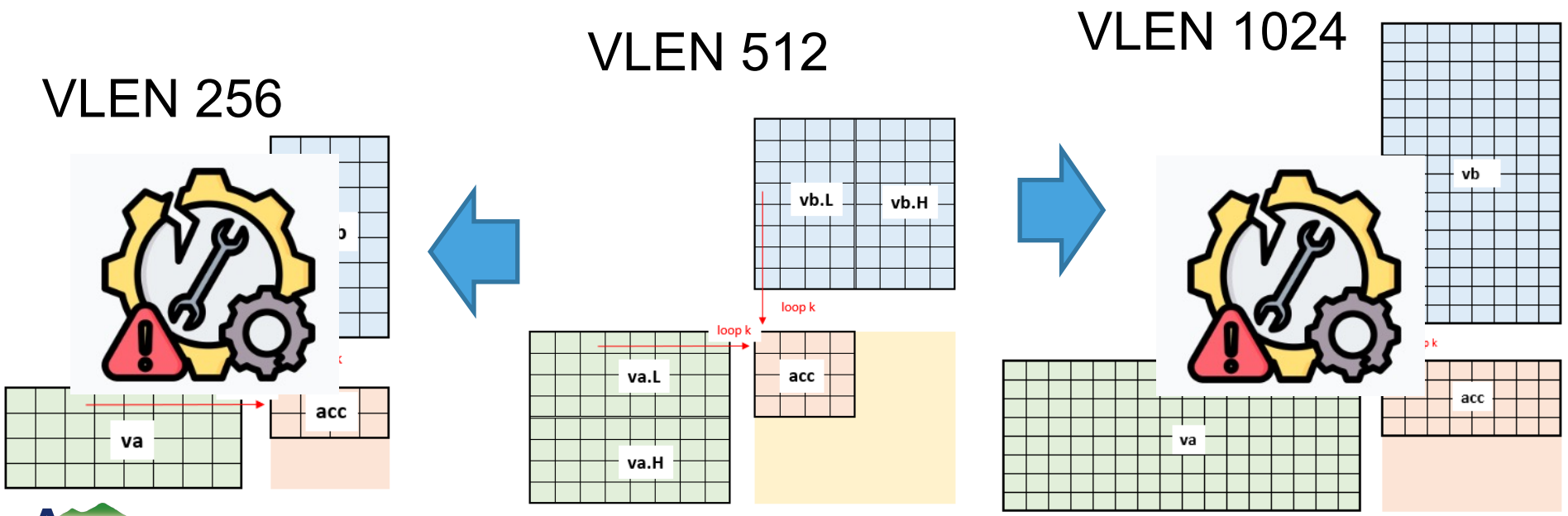


Efficient Tiling Load Store Access through Andes Streaming Port (ASP) to break the RVV load/store bottleneck:

- Pre-fetching
- Forming Tiles
- Sub-Matrix Re-shaping (Transpose)

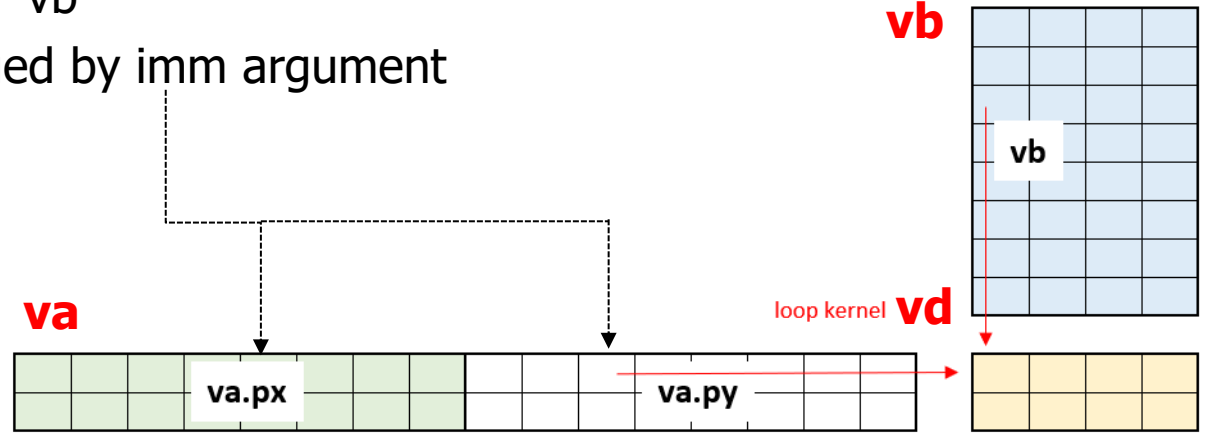
SW Scalability Challenges

- Tile Shape and Widening need reprograming when platform changes
  - E.g. int8 feature map/weight → accumulator widened to int32



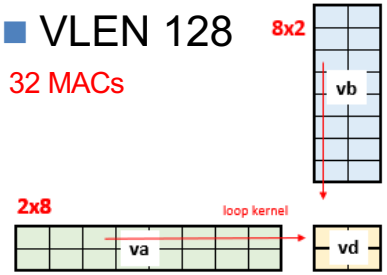
SW Scalability Innovations

- Achieve SW Portability by incorporating following innovations
  - Scalability management for vector registers
  - Diverse tile shape support using hybrid inner/outer products
  - Flexible matrix multiplication unit for versatile programs
- amm vd, vb, va, imm
  - vd = va.px/py * vb
  - portions managed by imm argument

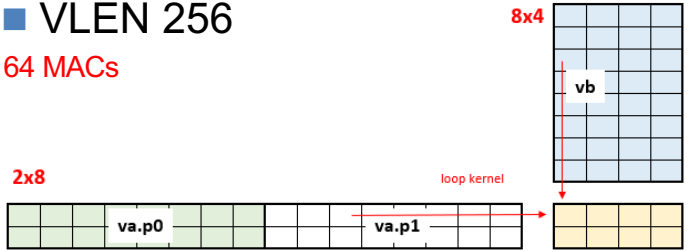


A Proposed Scalable Programming Model(1)

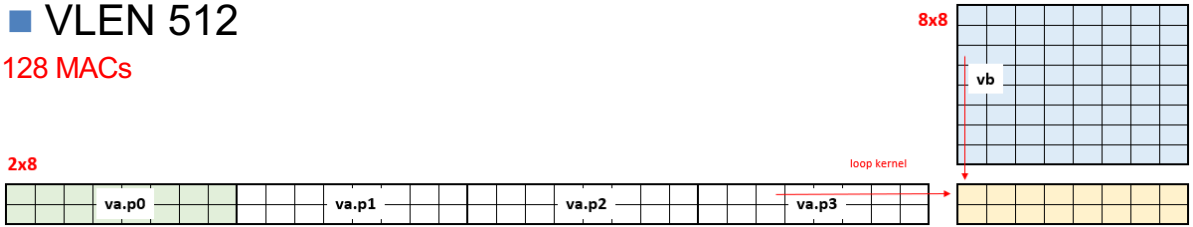
- VLEN 128
  32 MACs



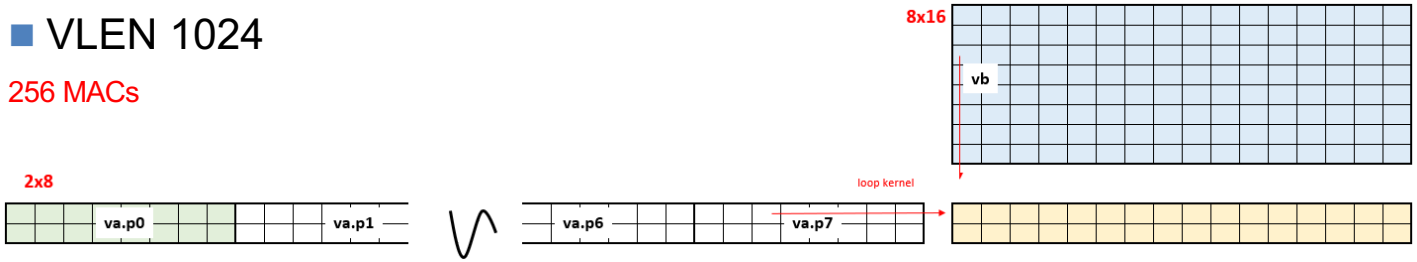
- VLEN 256
  64 MACs



- VLEN 512
  128 MACs



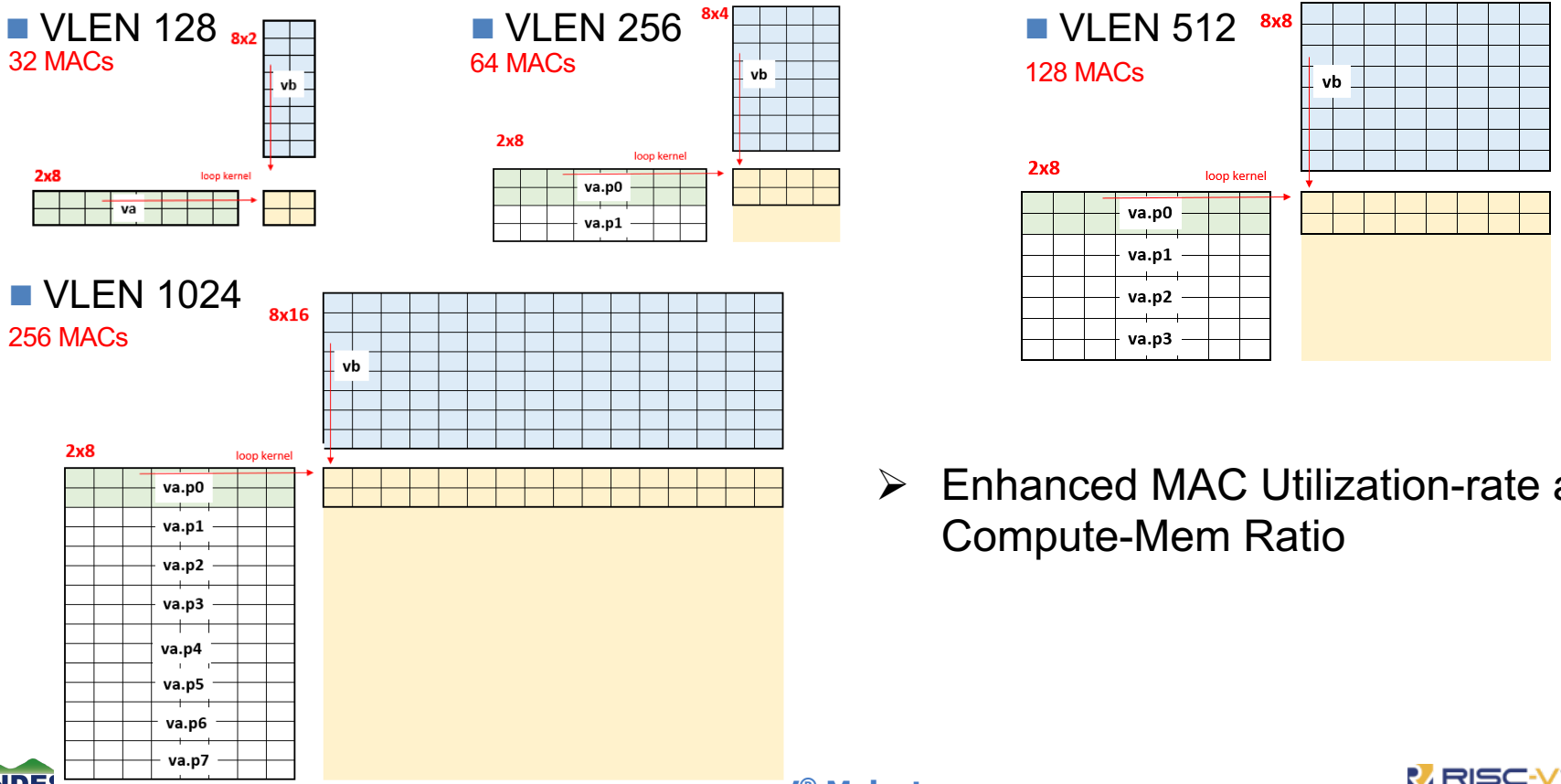
- VLEN 1024
  256 MACs



A Proposed Scalable Programming Model(2)

```
;E.g. VLEN 512, INT8 IFM/Weight Tiling
segs = VLEN >> exp(seg_size)
loopm:
  loopn:
  vand vc,0x0
    loopk:
      asp_vload va,[mem]
      loop_portion:
        asp_vload vb,[mem]
        amm vc, vb, va, portion
      loop loop_portion
    loop loopk
  vquan vc
  loop loopn
loop loopm
```

A Proposed Optimal Programming Model (1)



- Enhanced MAC Utilization-rate and Compute-Mem Ratio

A Proposed Optimal Programming Model (2)

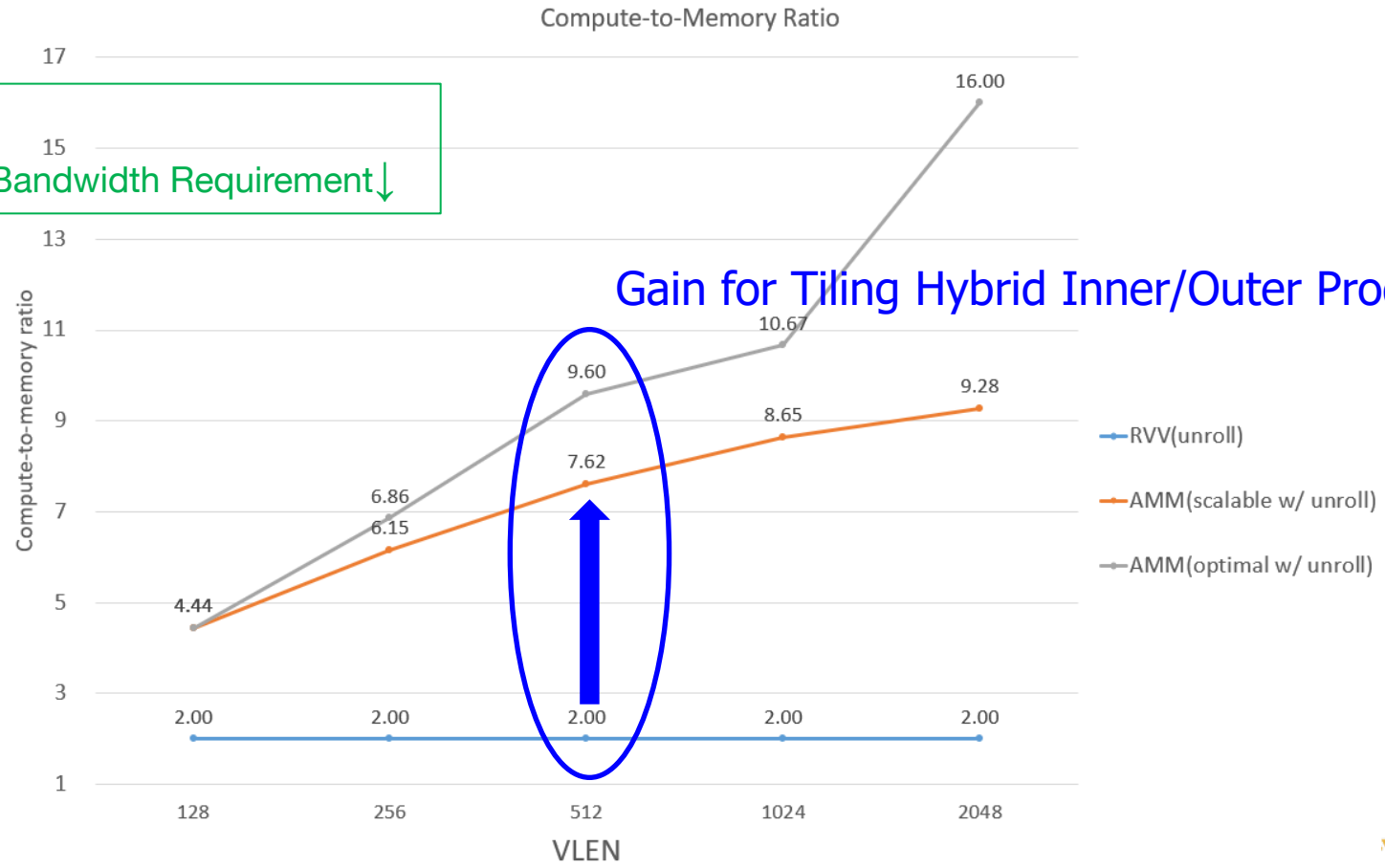
```
;E.g. VLEN 512, INT8 IFM/Weight Tiling
loopm:
  loopn:
  vand vd0-vd3,0x0
    loopk:
      asp_vload va,[mem]
      asp_vload vb,[mem]
      amm   vd0,vb,va,p0;va[portion 0]
      amm   vd1,vb,va,p1;va[portion 1]
      amm   vd2,vb,va,p2;va[portion 2]
      amm   vd3,vb,va,p3;va[portion 3]
    loop loopk
    vquan vd0-vd3
  loop loopn
loop loopm
```

+ Compute bound : MAC utilization-rate ↑
+ Compute-mem : Low memory bandwidth requirement ↓

"Baseline operations" to be capsulated as a Macro/Builtins
- VLEN(stride/portion size)
- VSRC1/2 w/ Memory Addr
- VDST

Preliminary Results of Reduced Memory Bandwidth

Compute-mem↑
Memory Access Bandwidth Requirement↓



Gain for Tiling Hybrid Inner/Outer Product

Preliminary Results for GeMM

- Scenario: GeMM 128x128 * 128x128

Architecture (VLEN/DLEN/AMM 512/512/512)	Speed-up	U-rate(%)
Std. RVV (libvec)	1x	~15%
AMM (optimal w/o unroll)	~3.3x	~39%
AMM (optimal w/ unroll)	**~6.9x**	**~82%**

Conclusion

- Assessing Std. RVV on Matrix Operation, Including:
  - SW Scalability/Portability cross Diverse Platforms
  - Memory Bandwidth Requirements
  - Load/Store Performance Bottlenecks
  - Challenges on Matrix boundary handling
- Integrated-Vector Register Files with Customized Matrix Extension ISA is Briefly Summarized
- Preliminary Results Demonstrate Significant Performance Improvement
- A Reference Arch. and Preliminary Results for RISC-V M-ext. TG.

ANDES
TECHNOLOGY

Follow Andes, Find Latest Trends

扫一扫，关注晶心