Vector IRAM Memory Performance for Image Access Patterns

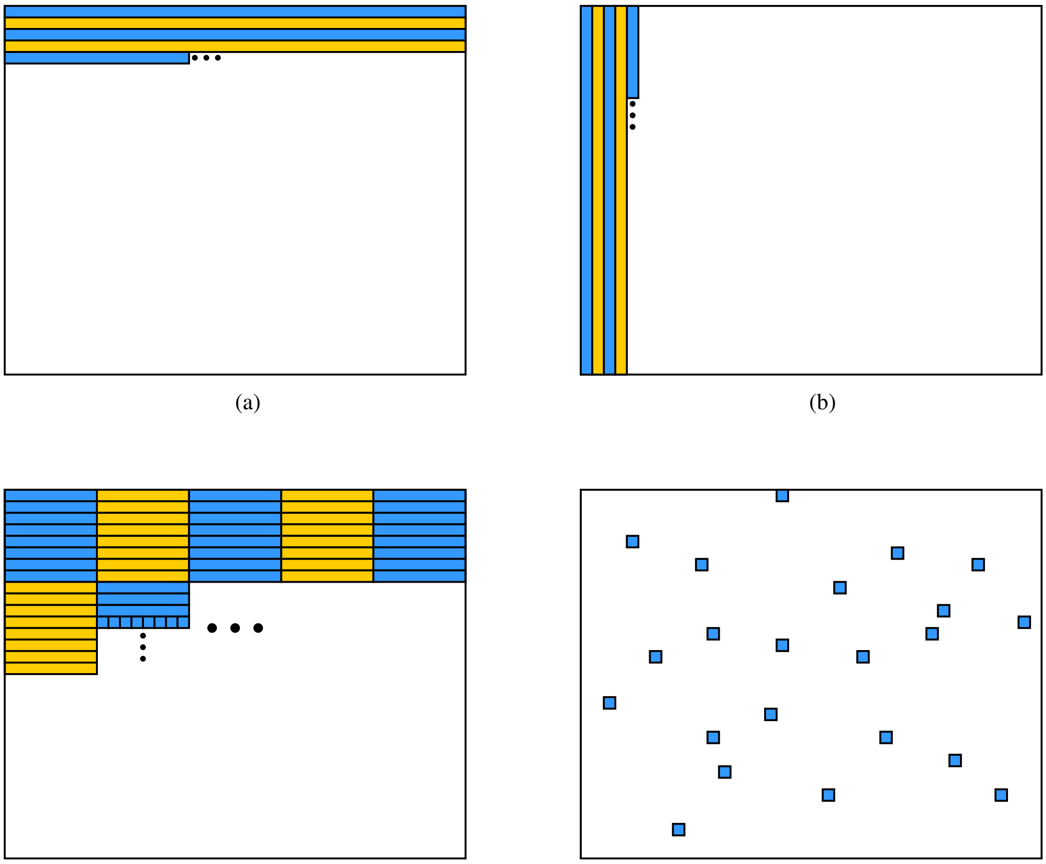
Richard Fromm

rfromm@cs.berkeley.edu

`http://www.cs.berkeley.edu/~rfromm/masters.html`

Study VIRAM memory system performance

- Image access patterns representative of multimedia applications
  - (a) Horizontal
  - (b) Vertical
  - (c) 8 $\times$8 blocked
  - (d) Random



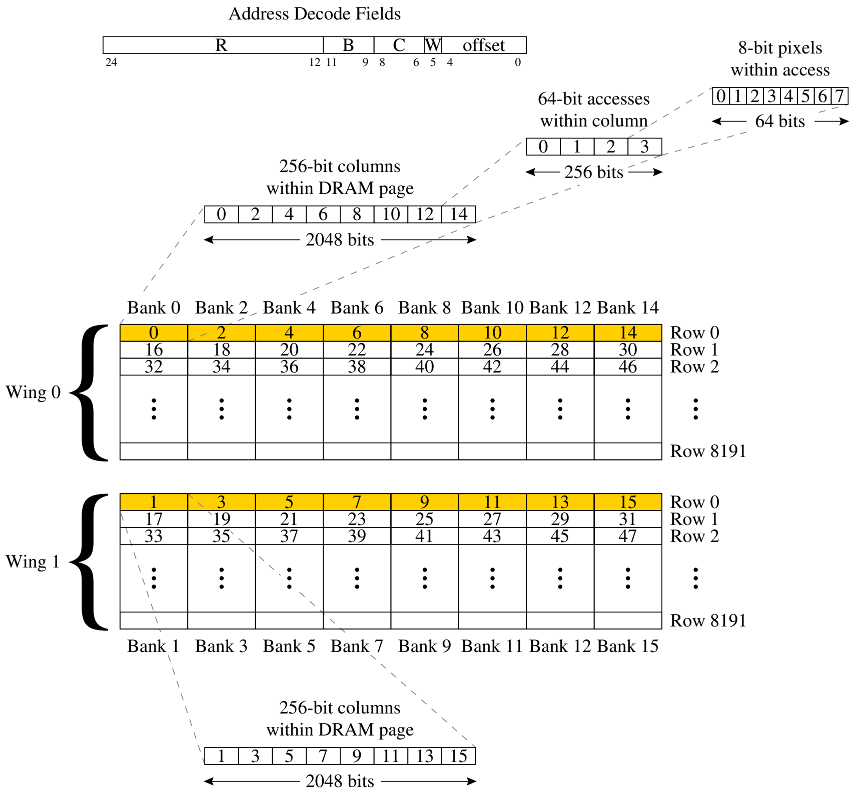
(a) (b) (c) (d)

- Image Sizes

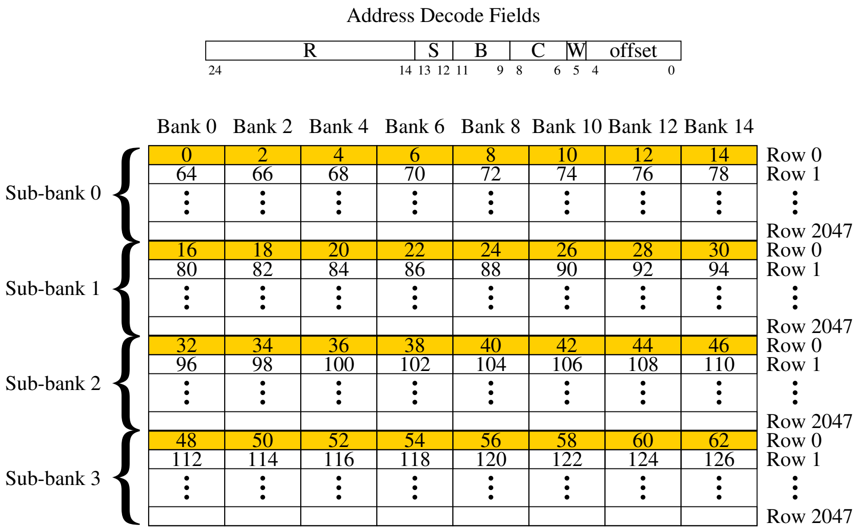
128x96	512x384	800x600	1600x1200
176x144	544x480	832x624	1800x1440
352x240	640x480	1024x768	1920x1080
352x288	704x480	1152x864	1920x1200
352x480	720x400	1280x720	
480x480	720x480	1280x1024	

VIRAM Memory System Organization

- Default configuration, 32 MB, RSBCW layout



- Wing divided into 4 sub-banks

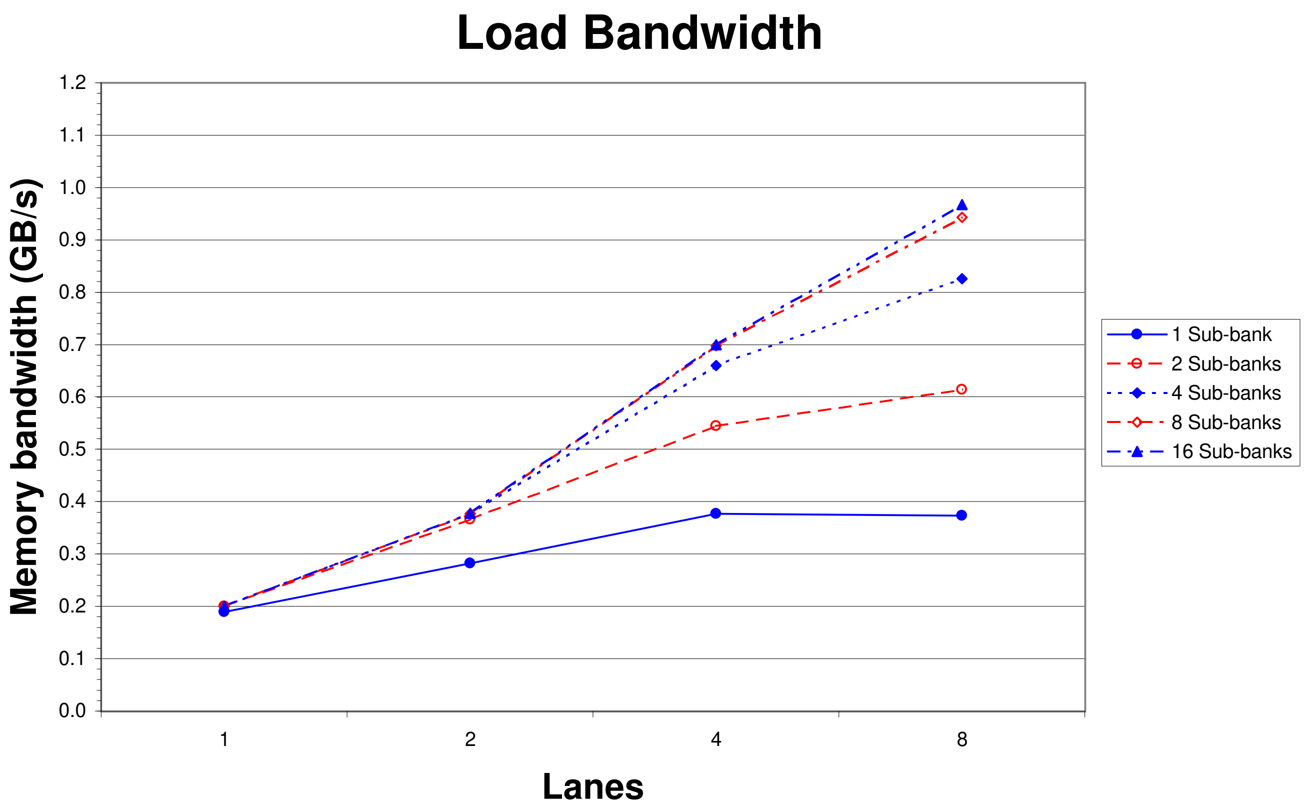


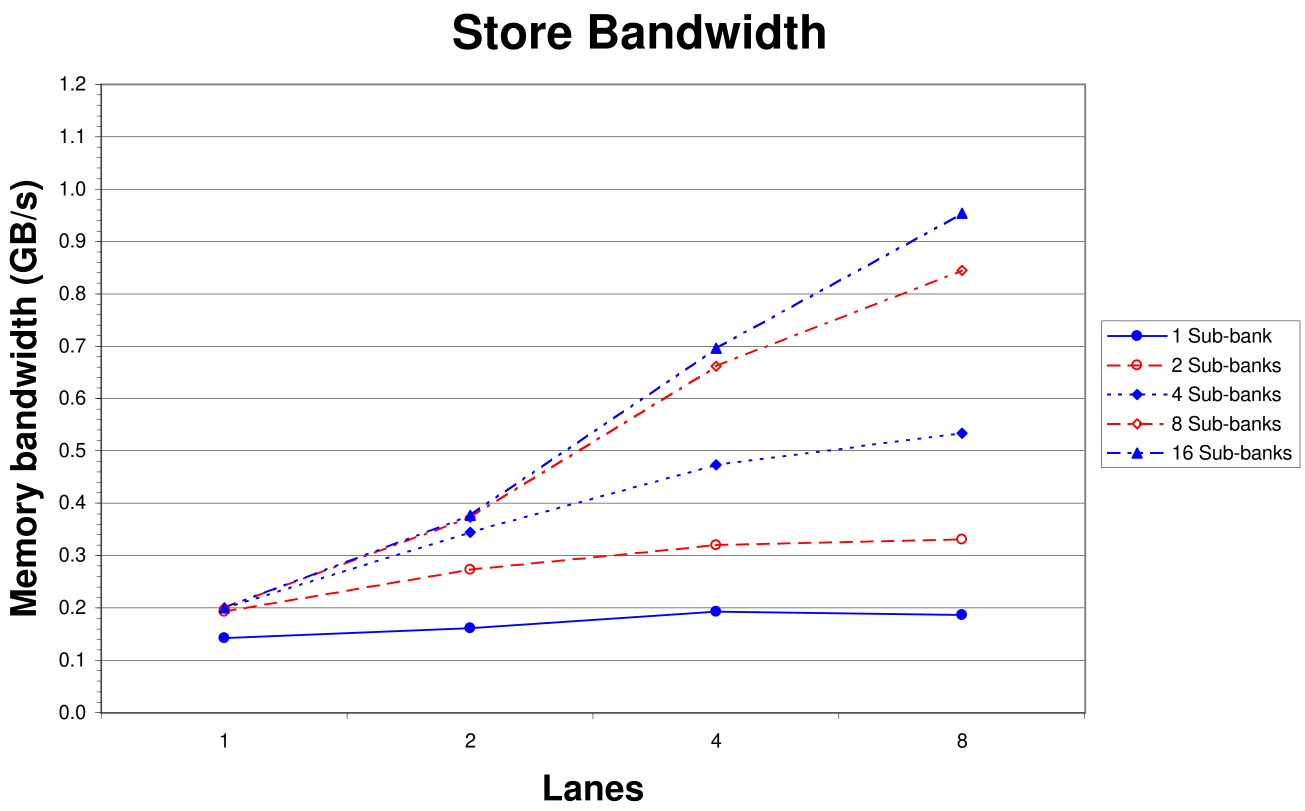
Summary of Results

Access Pattern	Bandwidth in GB/s (Percentage of Peak)				
		Load		Store	
	Peak	Mean	Std dev	Mean	Std dev
Horizontal	6.4	6.4 (100%)	0.01 (0%)	6.4 (100%)	0.01 (0%)
Vertical	0.8	0.38 (47%)	0.16 (19%)	0.19 (24%)	0.09 (11%)
8 × 8 Blocked	6.4	1.4 (21%)	0.09 (1%)	1.1 (17%)	0.31 (5%)
Random	0.8	0.20 (25%)	0.01 (2%)	0.10 (12%)	0.00 (1%)

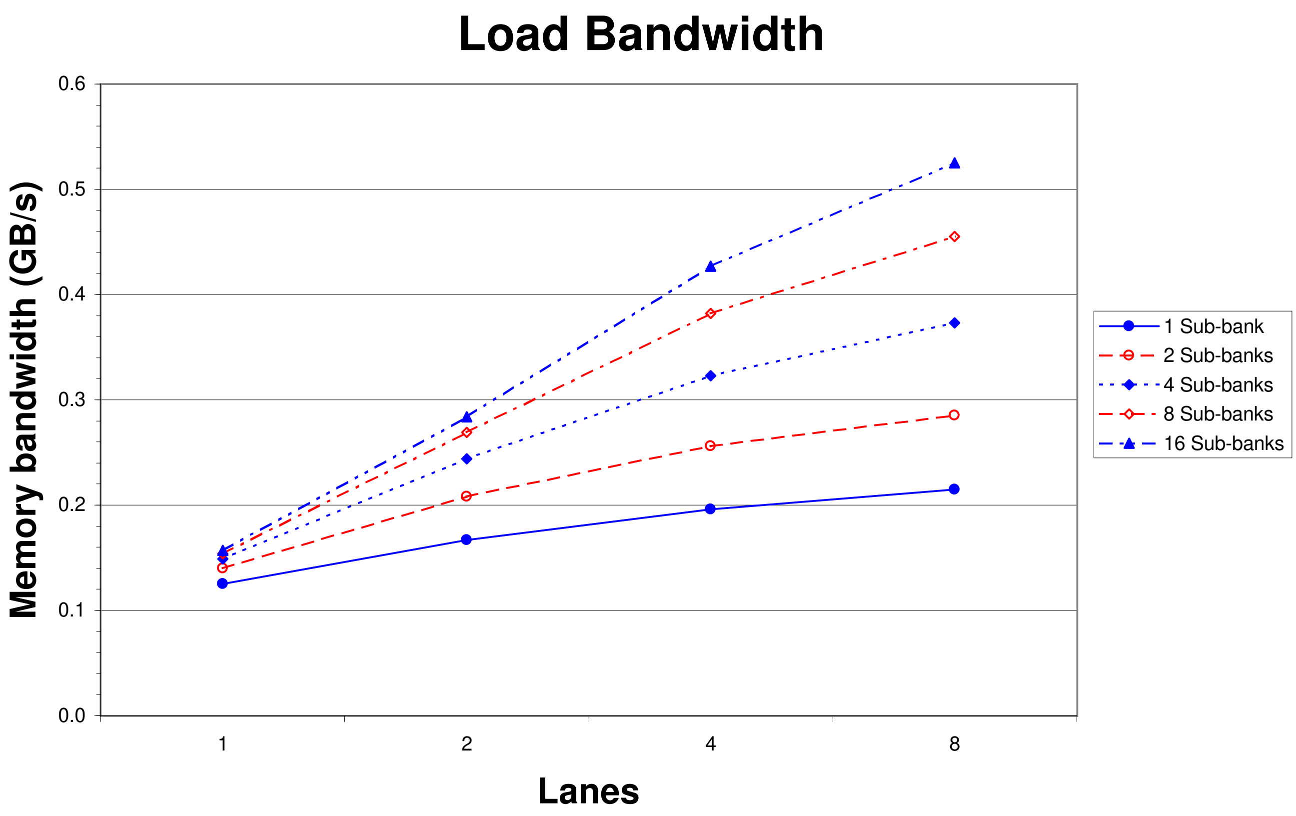
- Easy to get peak performance out of large unit strides
- Harder to get peak performance otherwise
- Averages can be deceiving
  - Can be wide variance (esp. strided)
- Factors limiting performance
  - Bank Conflicts
    - One request per bank per cycle
  - Sub-bank Conflicts
    - Sub-bank busy time
      - 4 cycles (loads)
      - 9 cycles (stores)
  - Short Vectors
  - Insufficient issue bandwidth
  - Simplified pipeline control
  - Data alignment

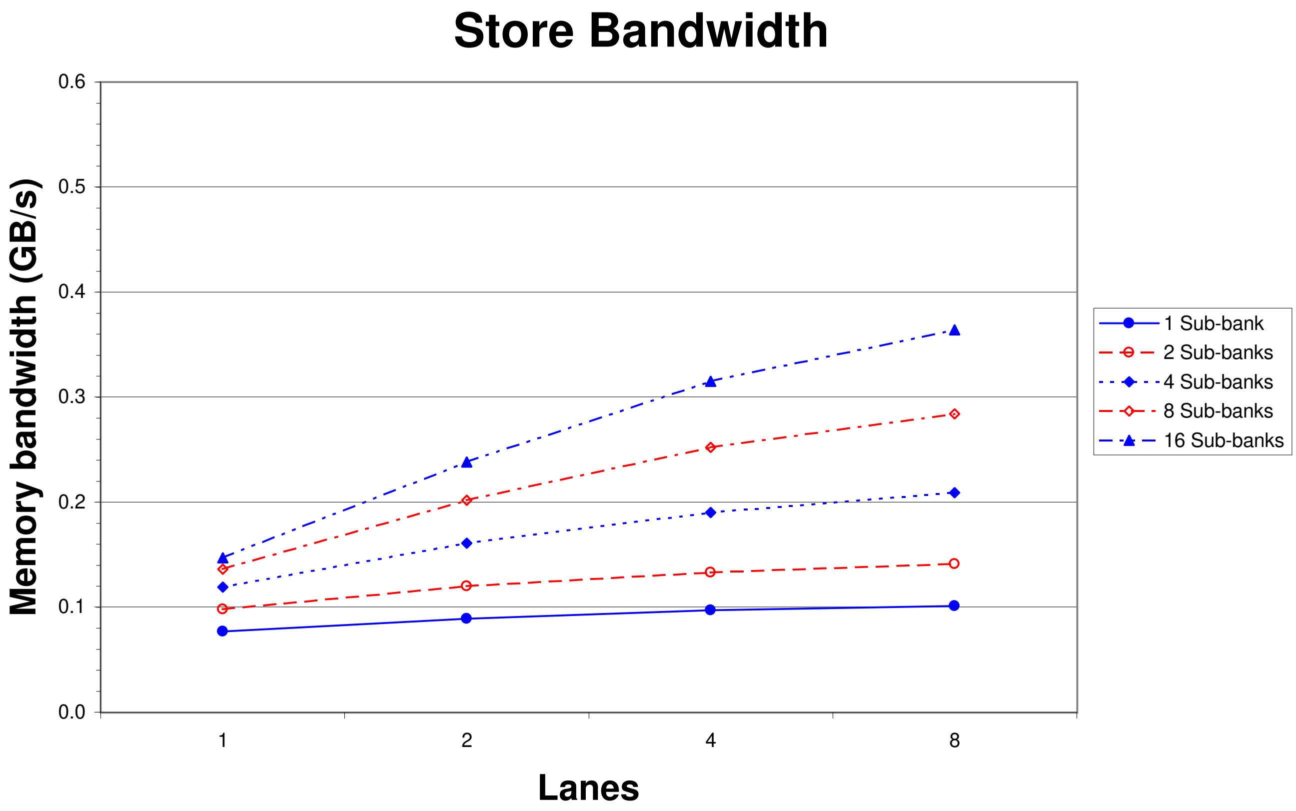
Effect of Sub-banks on Vertical Pattern





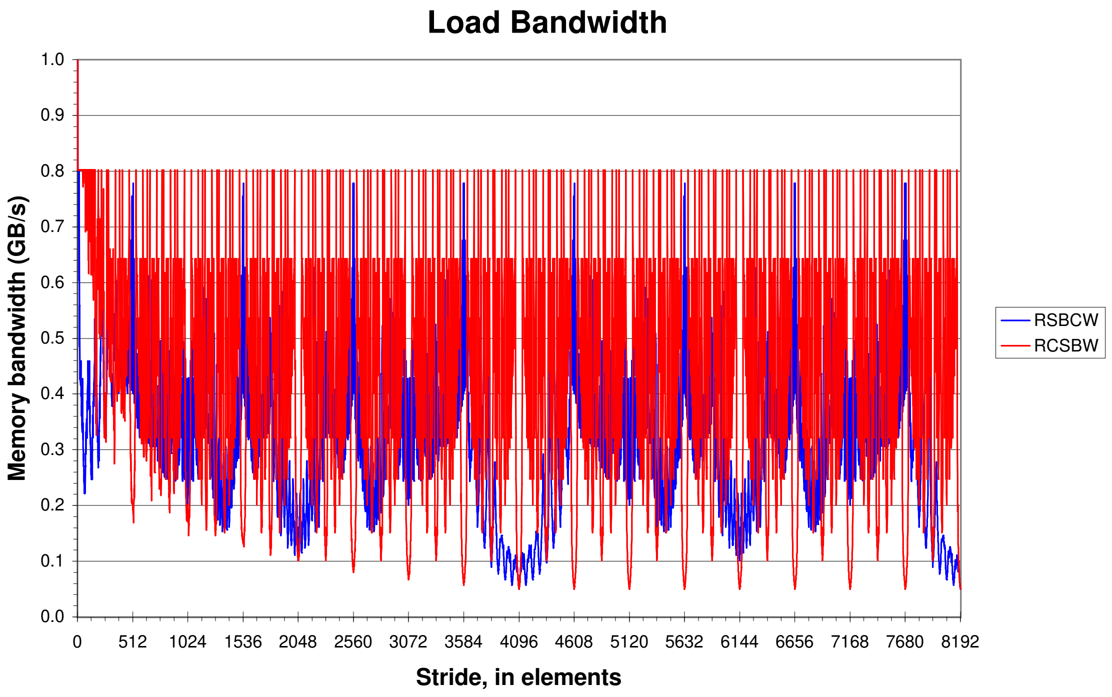
Effect of Sub-banks on Randomized Pattern

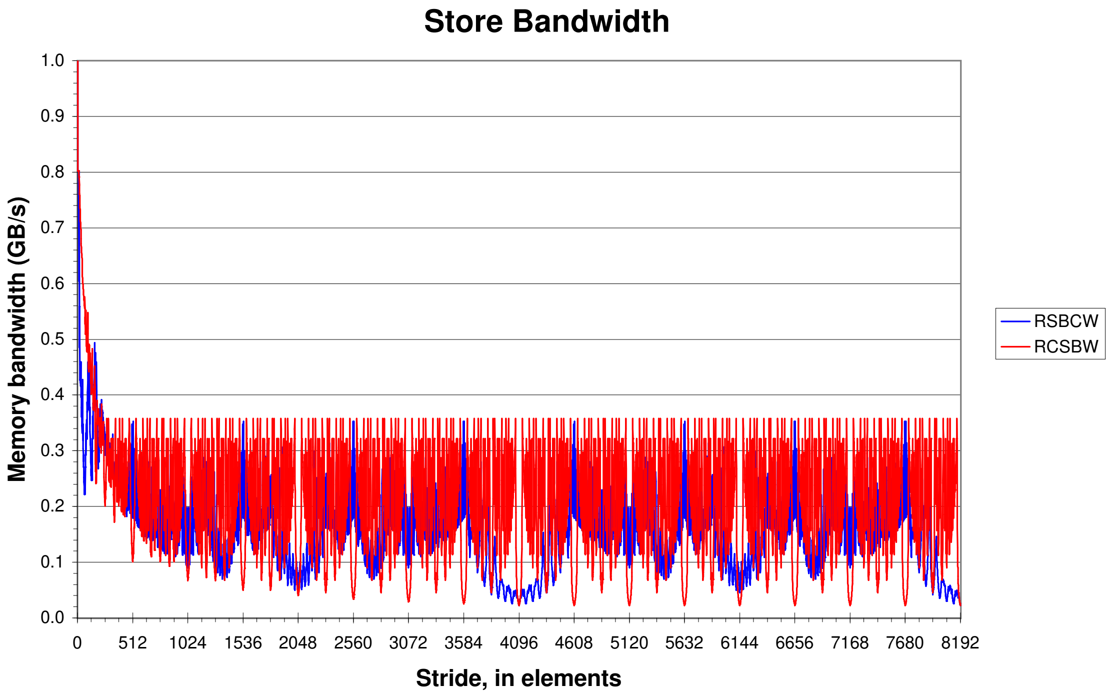




Alternative Data Layout

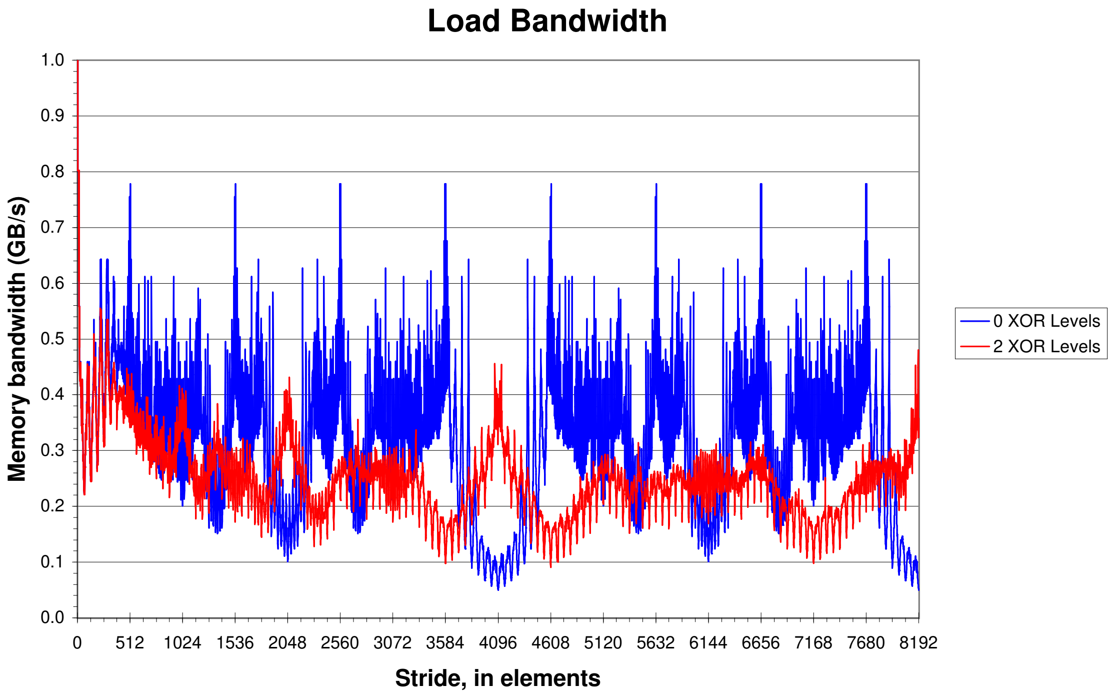
- (+) Modestly improves mean vertical access bandwidth
- (−) Increases variance and decreases bandwidth of some cases

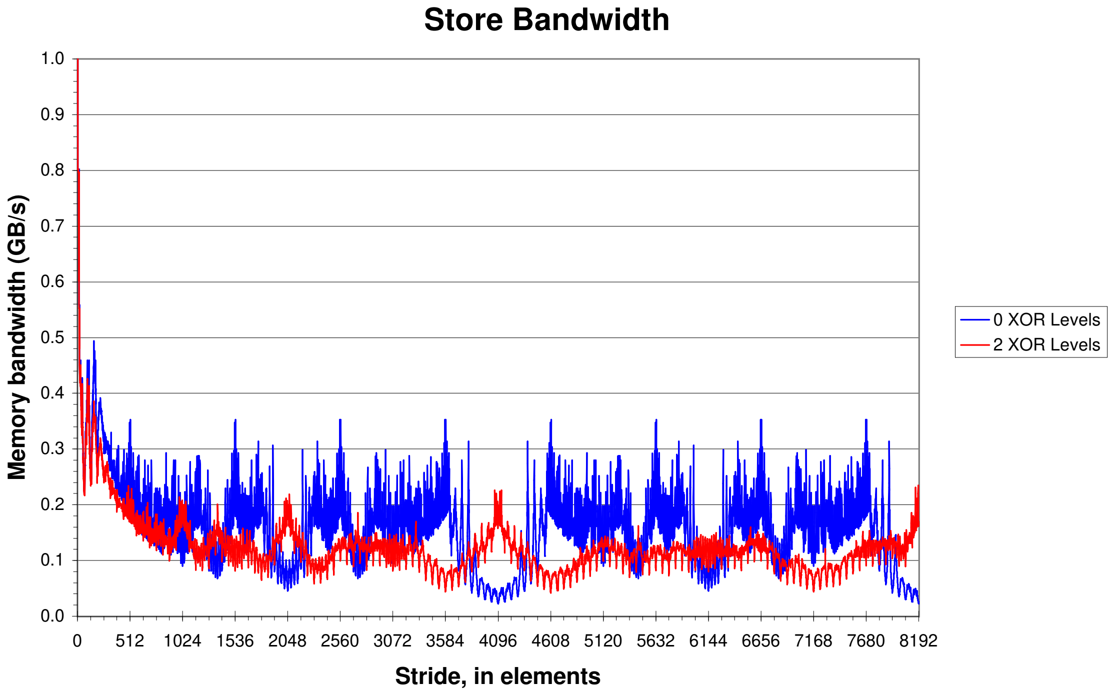




Simple (XOR) Address Hashing Scheme

- (−) Decreases mean vertical access bandwidth
- (+) Decreases variance and increases bandwidth of some cases





Conclusions

- The bad news
  - Answer to question "What is the performance of the VIRAM memory system" is "It depends"
  - Can be difficult to reason about data placement and relation to bank and sub-bank conflicts
  - Lack of sub-banks are severe limitation on performance
  - Extra address generation resources would be useless without memory system to support them (not shown)
  - Multiple memory units can interfere with each other
- The good news
  - Even when less than peak, still impressive bandwidth
    - Best performance of *any* Intel x86 based PC for memory to memory copy STREAM benchmark is 304.0 MB/s
      - This is a 400 MHz machine, twice VIRAM-1
  - Real applications are not only loads and stores
    - With sufficient computation to memory ratio and proper scheduling, hope that bandwidth is sufficient and application can keep all VFUs busy
  - Only one memory unit (recent change) makes reasoning easier, eliminates interference between memory units
  - Only one memory unit allows for decoupled stores (and maybe loads)