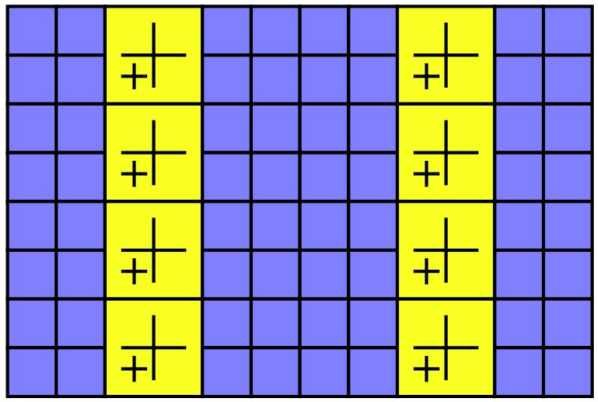
Impact of Memory Architecture on FPGA Energy Consumption



Edin Kadric, David Lakata, André DeHon

Penn
University of Pennsylvania

App/Arch Mismatch

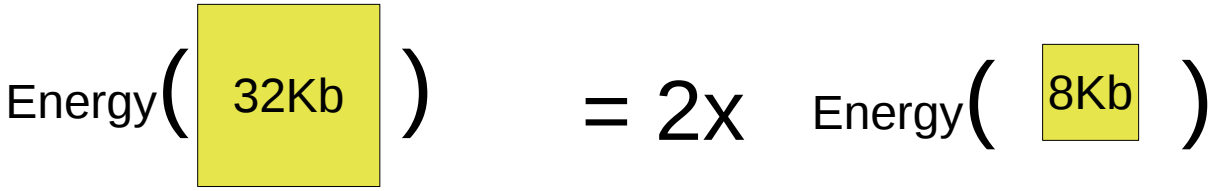
- FPGAs are one-size-fits-all architectures
  - → **mismatch** between App and Arch

App/Arch Mismatch

- FPGAs are one-size-fits-all architectures

  → **mismatch** between App and Arch

- Memory cost:



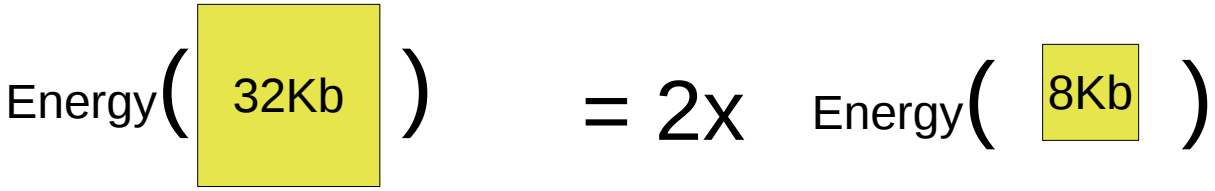
- Memory mismatch → excess energy

App/Arch Mismatch

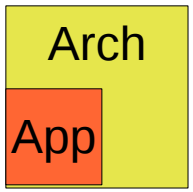
- FPGAs are one-size-fits-all architectures

  → **mismatch** between App and Arch

- Memory cost:



- Memory mismatch → excess energy
  ($M_{app} < M_{arch}$)



App/Arch Mismatch

- FPGAs are one-size-fits-all architectures

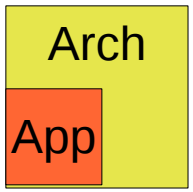
  → **mismatch** between App and Arch

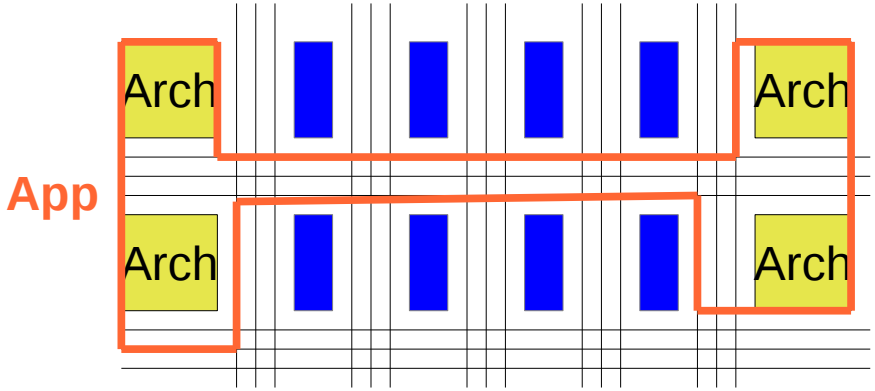
- Memory cost:

Energy(32Kb) = 2x Energy(8Kb)

- Memory mismatch → excess energy

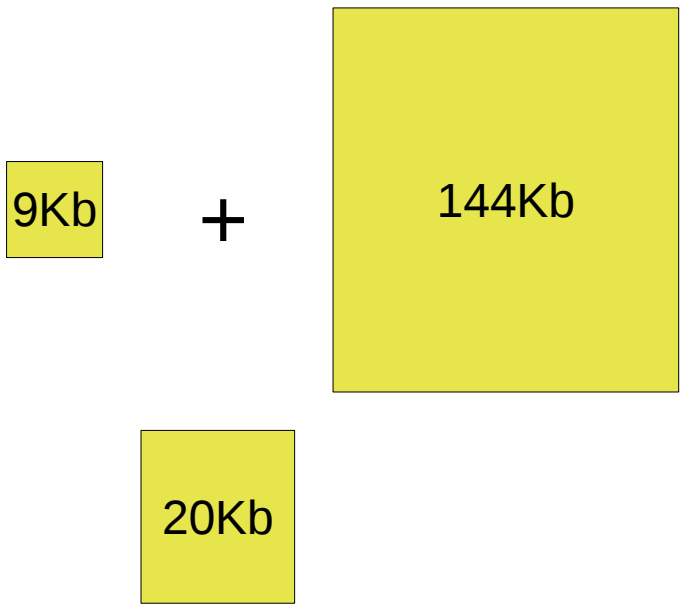
  ($M_{app} < M_{arch}$) ($M_{arch} < M_{app}$)





Mismatch Also Impacts Area

- “Architectural enhancements in Stratix V”
  Lewis *et al.* (FPGA 2013)
  - Moved from Stratix IV

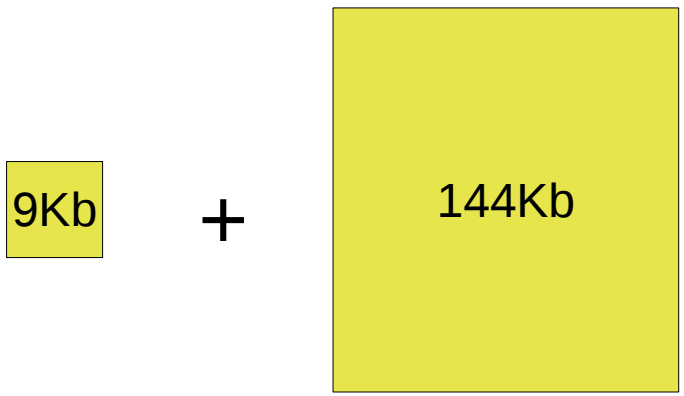


  - To Stratix V

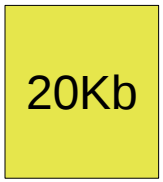
  - Goal: minimize area

Mismatch Also Impacts Area

- “Architectural enhancements in Stratix V”
  Lewis *et al.* (FPGA 2013)
  - Moved from Stratix IV



  - To Stratix V

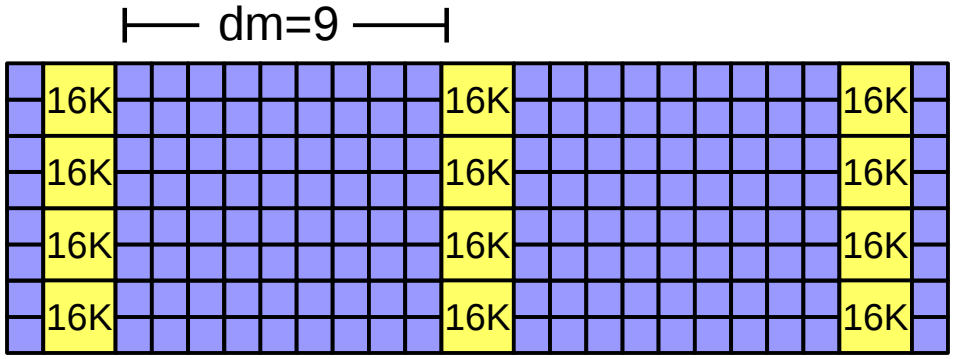


  - Goal: minimize area

- Question: How about energy?

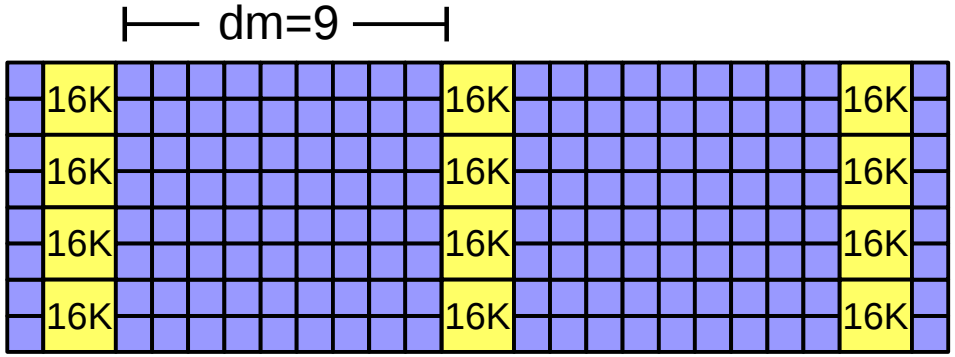
Mismatch Example

- Stratix V: 20Kb memories, ~10% of columns
- ~Stratix V: 16Kb memories placed every 10 columns



Mismatch Example

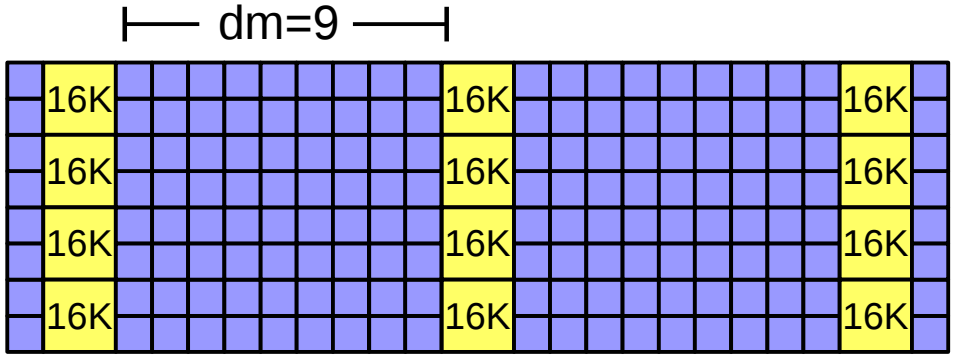
- Stratix V: 20Kb memories, ~10% of columns
- ~Stratix V: 16Kb memories placed every 10 columns



  - On *spree.v*: 147% energy overhead

Mismatch Example

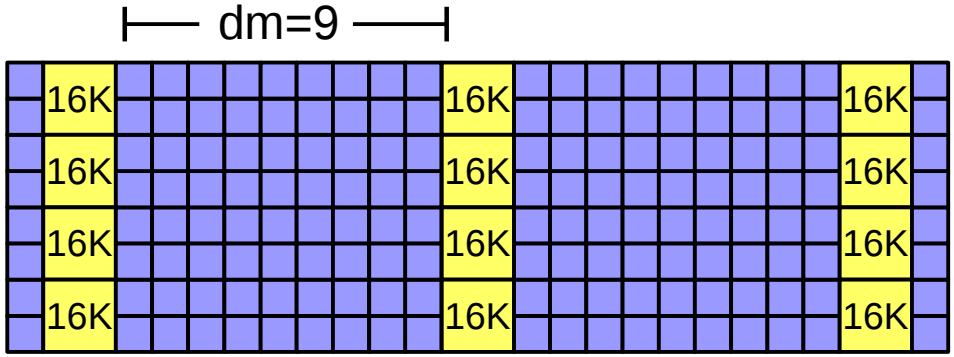
- Stratix V: 20Kb memories, ~10% of columns
- ~Stratix V: 16Kb memories placed every 10 columns



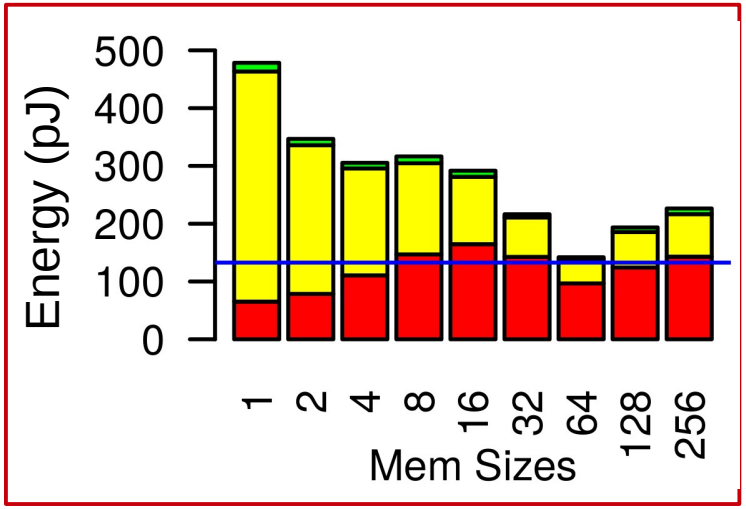
  - On *spree.v*: 147% energy overhead
- Alternative:
- 64Kb placed every 5 columns
  - 7% overhead
  - Energy reduced by 2.3x

Mismatch Example

- Stratix V: 20Kb memories, ~10% of columns
- ~Stratix V: 16Kb memories placed every 10 columns

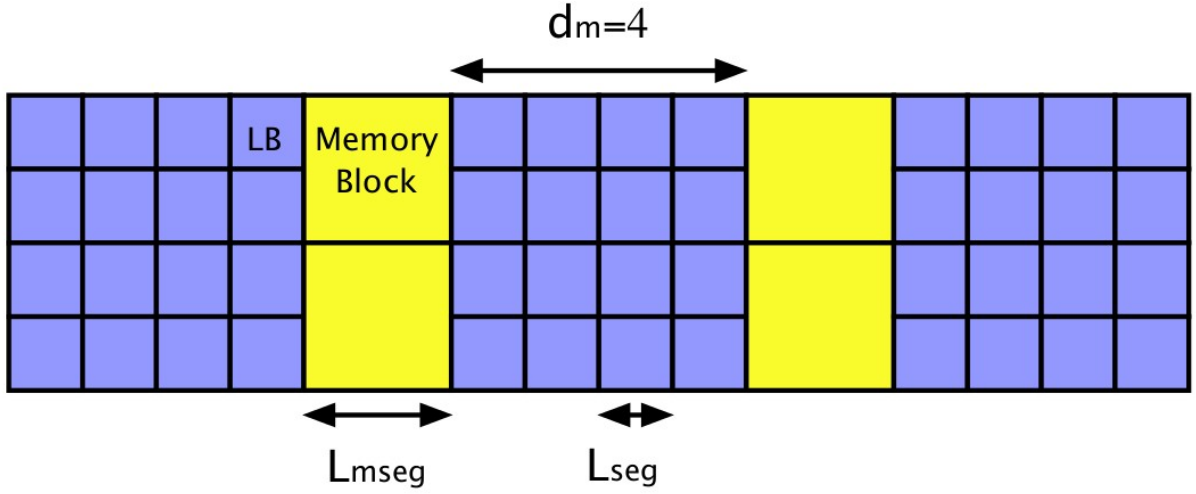


  - On *spree.v*: 147% energy overhead
- Alternative:
- 64Kb placed every 5 columns
  - 7% overhead
  - Energy reduced by 2.3x



To Reduce Energy (Mismatch):

- How large should the on-chip memories be?
- How frequently should they be placed?
- How should they be organized internally?
- How many different levels of memory?



Outline

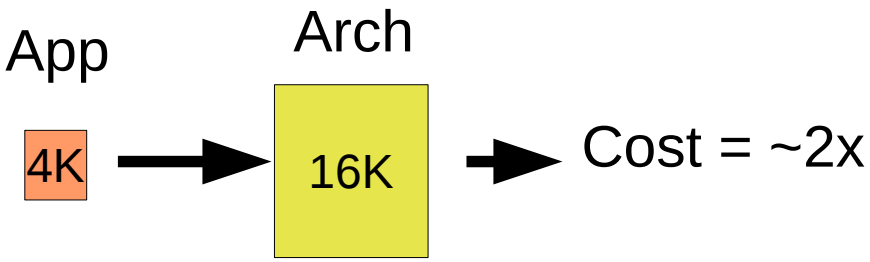
1) Motivation
2) Mismatch Reduction Techniques
3) Experimental Architecture Exploration

Outline

1) Motivation
2) Mismatch Reduction Techniques
    a) Multiple Memory Levels
    b) Continuous Hierarchy Memory (CHM)
3) Experimental Architecture Exploration

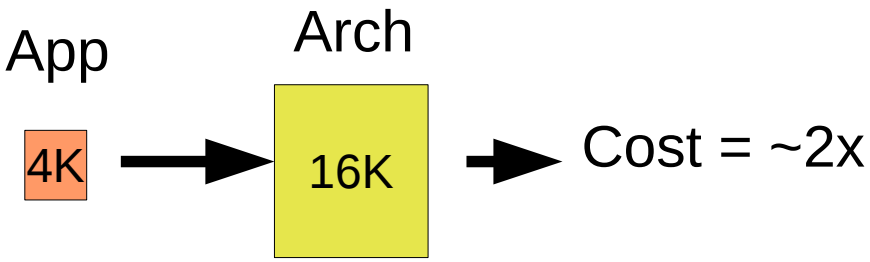
Multiple Memory Levels

- Provides more choice to the mapping tool
- Stratix V: M20K (~16K)

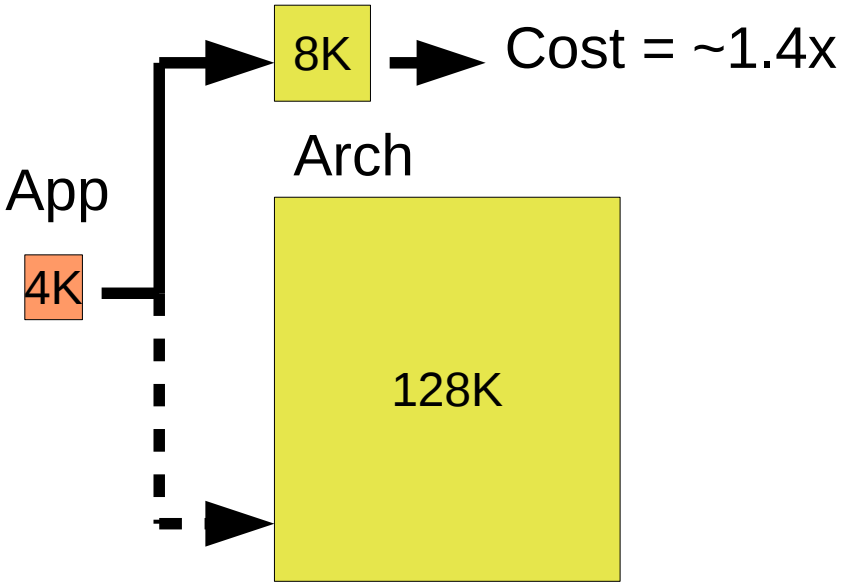


Multiple Memory Levels

- Provides more choice to the mapping tool
- Stratix V: M20K (~16K)

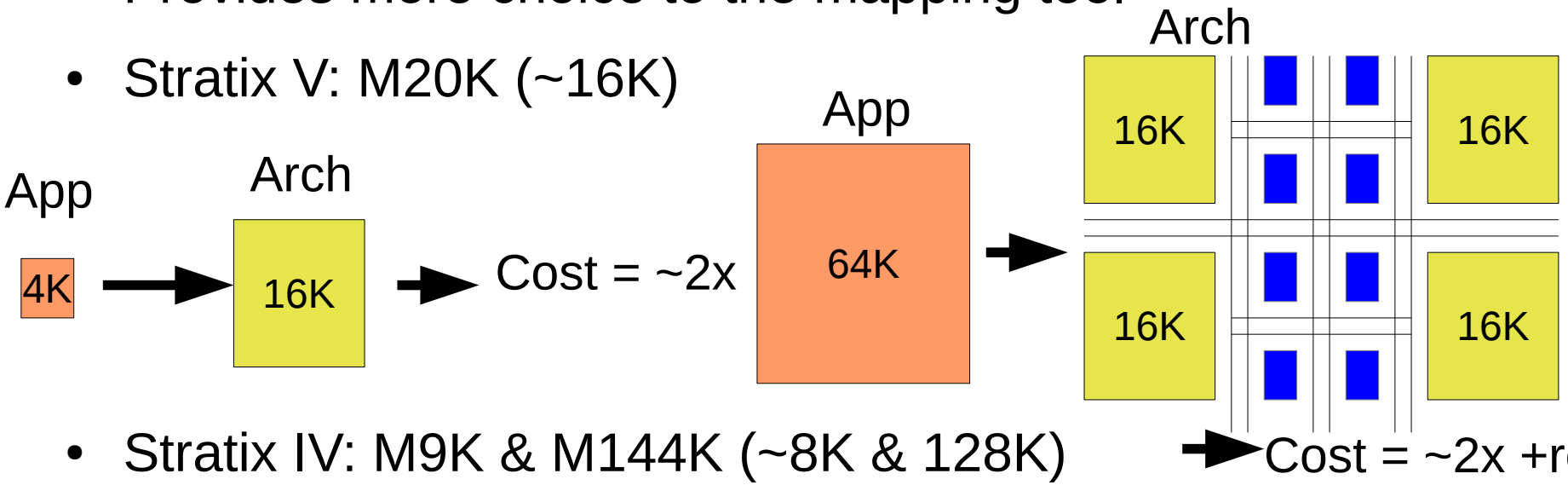


- Stratix IV: M9K & M144K (~8K & 128K)

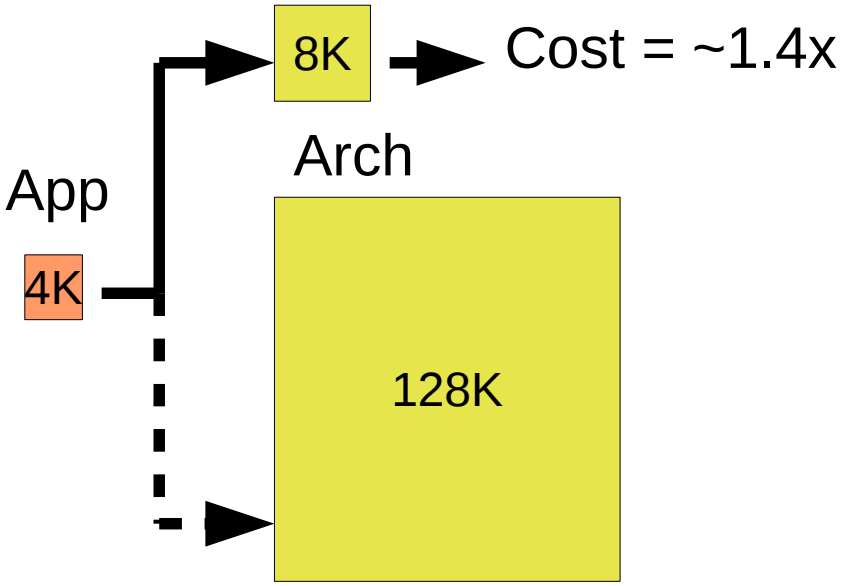


Multiple Memory Levels

- Provides more choice to the mapping tool
- Stratix V: M20K (~16K)

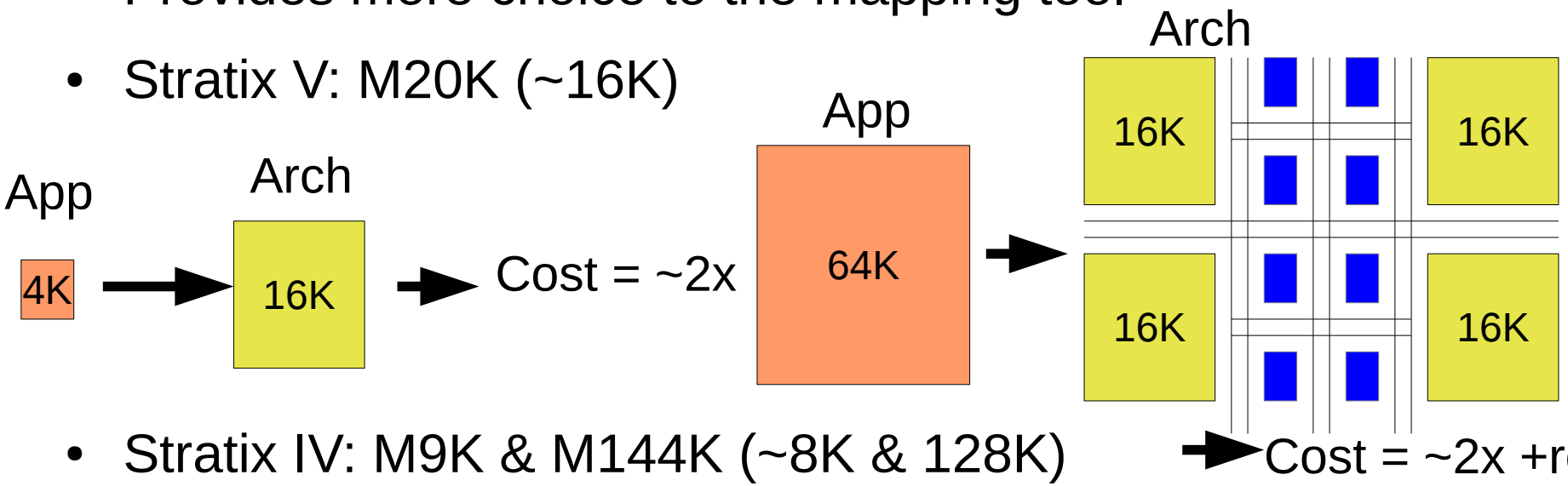


- Stratix IV: M9K & M144K (~8K & 128K)

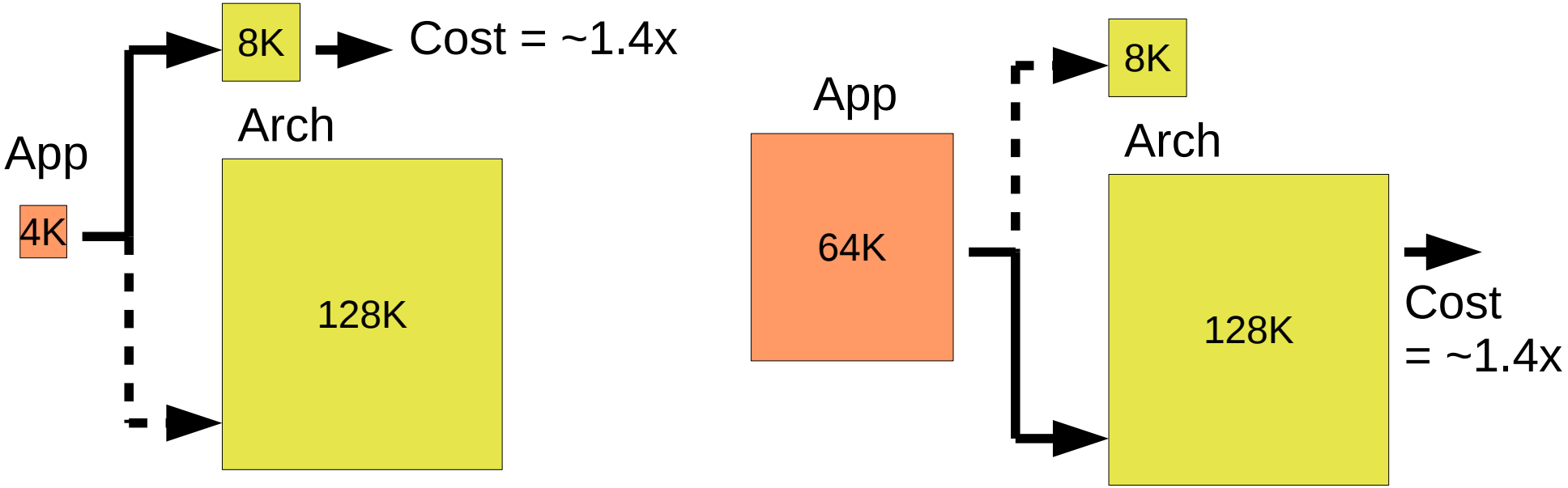


Multiple Memory Levels

- Provides more choice to the mapping tool
- Stratix V: M20K (~16K)

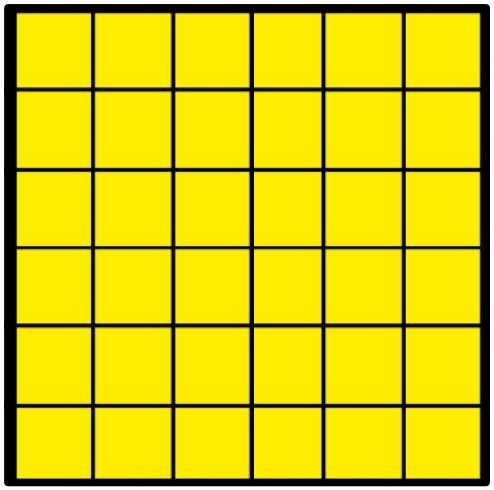


- Stratix IV: M9K & M144K (~8K & 128K)



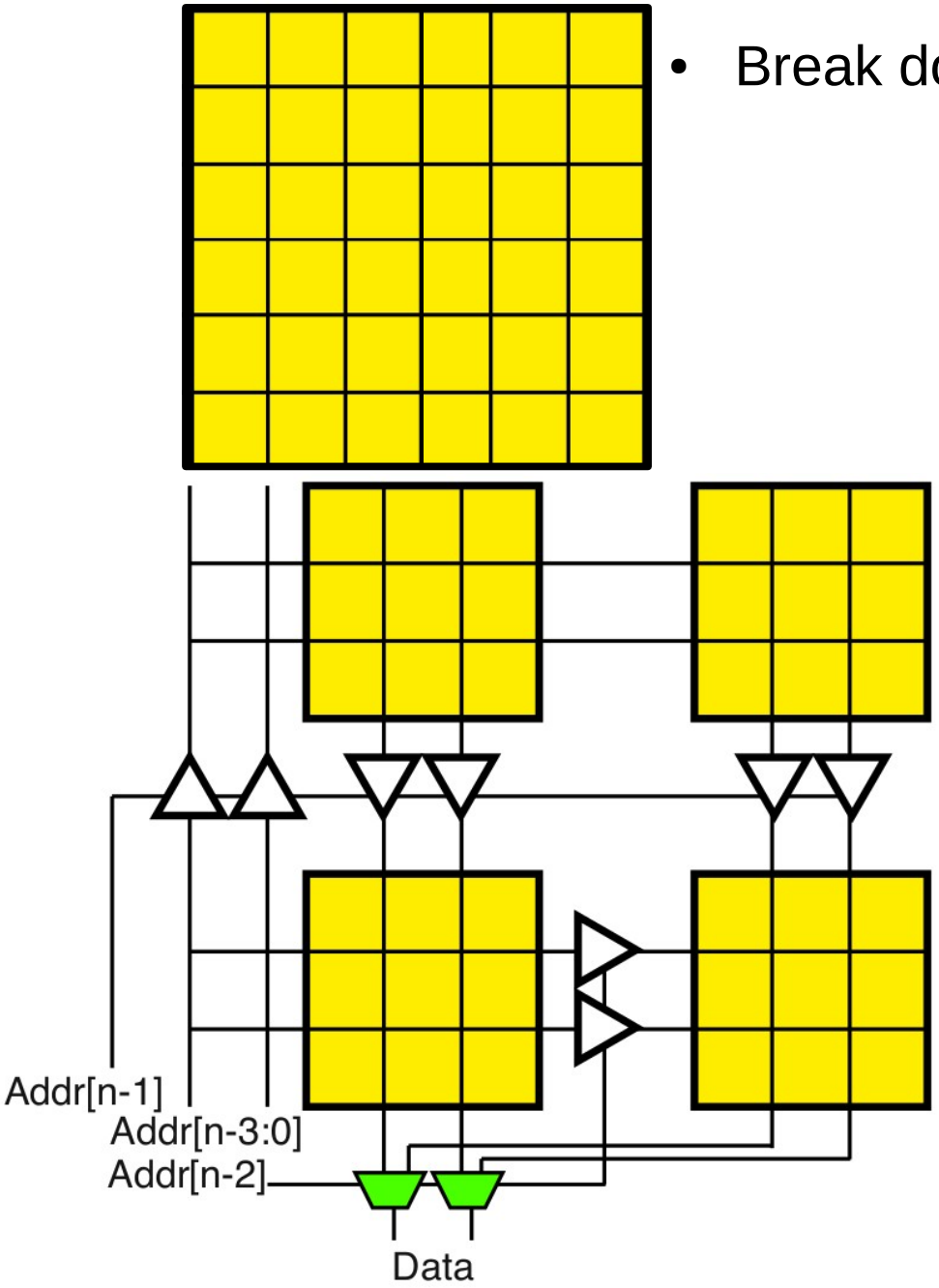
Continuous Hierarchy Memory

"Kung Fu data energy, minimizing communication energy in FPGA computations"
(Kadric *et al.* FCCM 2014)



Continuous Hierarchy Memory

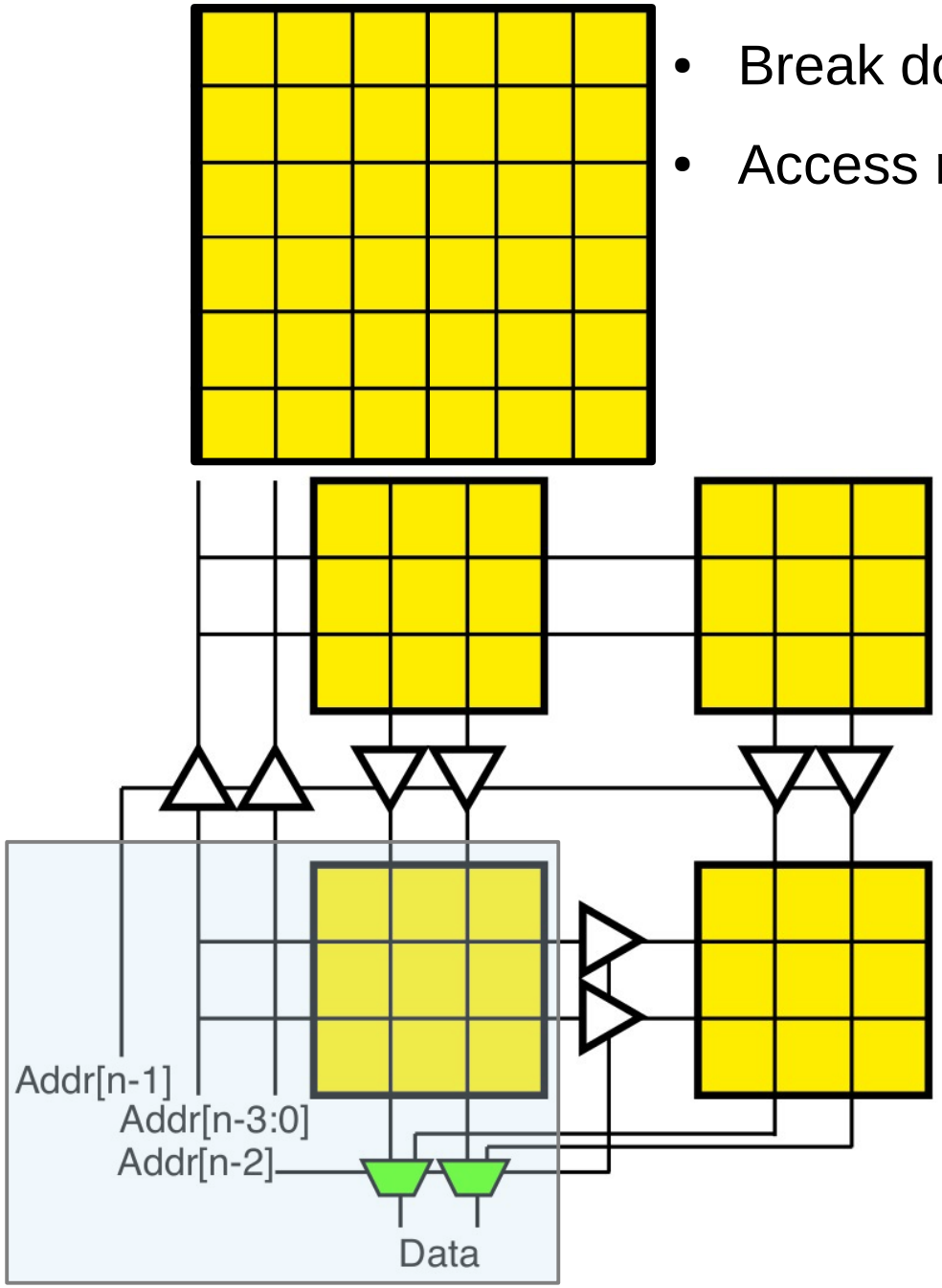
"Kung Fu data energy, minimizing communication energy in FPGA computations"
(Kadric *et al.* FCCM 2014)



- Break down into smaller banks

Continuous Hierarchy Memory

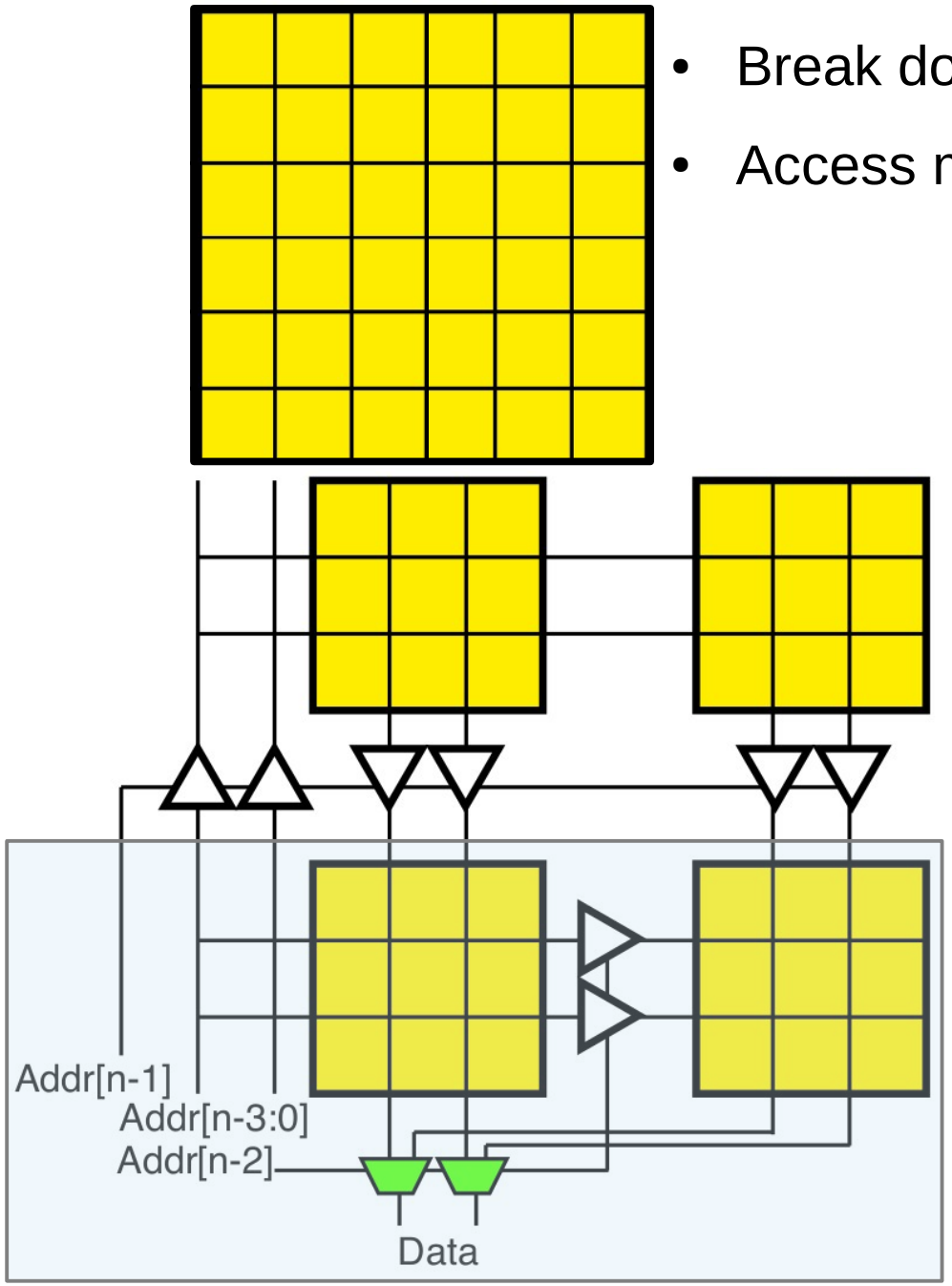
"Kung Fu data energy, minimizing communication energy in FPGA computations"
(Kadric *et al.* FCCM 2014)



- Break down into smaller banks
- Access memory closest to I/O

Continuous Hierarchy Memory

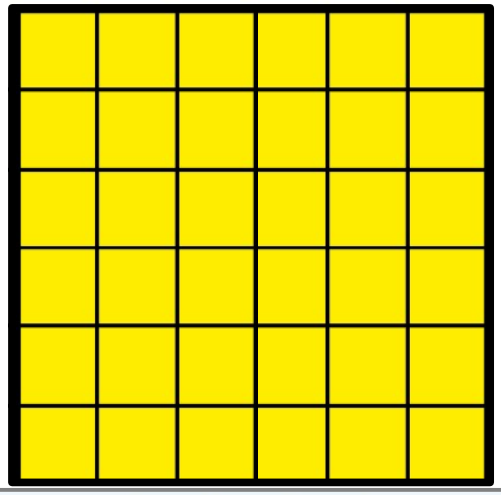
"Kung Fu data energy, minimizing communication energy in FPGA computations"
(Kadric *et al.* FCCM 2014)



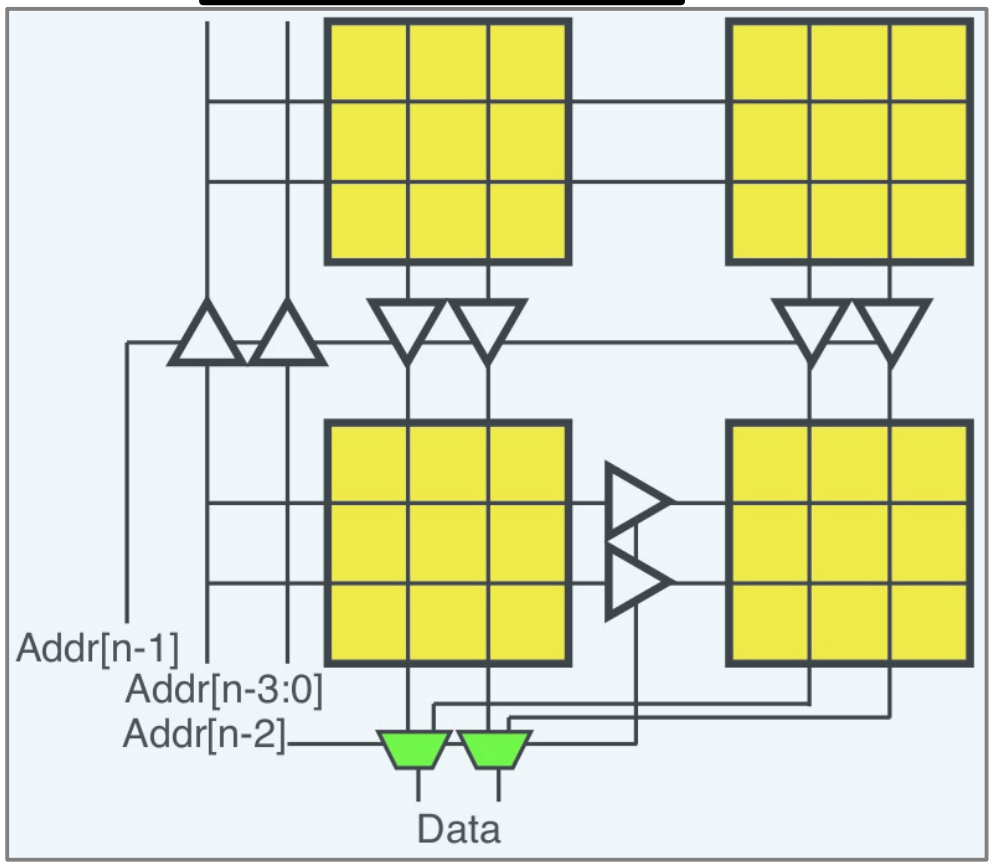
- Break down into smaller banks
- Access memory closest to I/O

Continuous Hierarchy Memory

"Kung Fu data energy, minimizing communication energy in FPGA computations"
(Kadric *et al.* FCCM 2014)

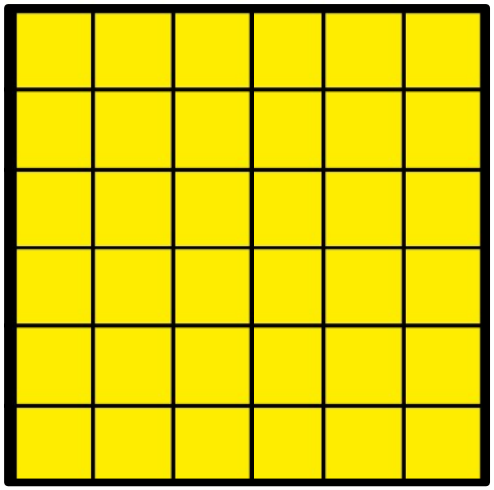


- Break down into smaller banks
- Access memory closest to I/O

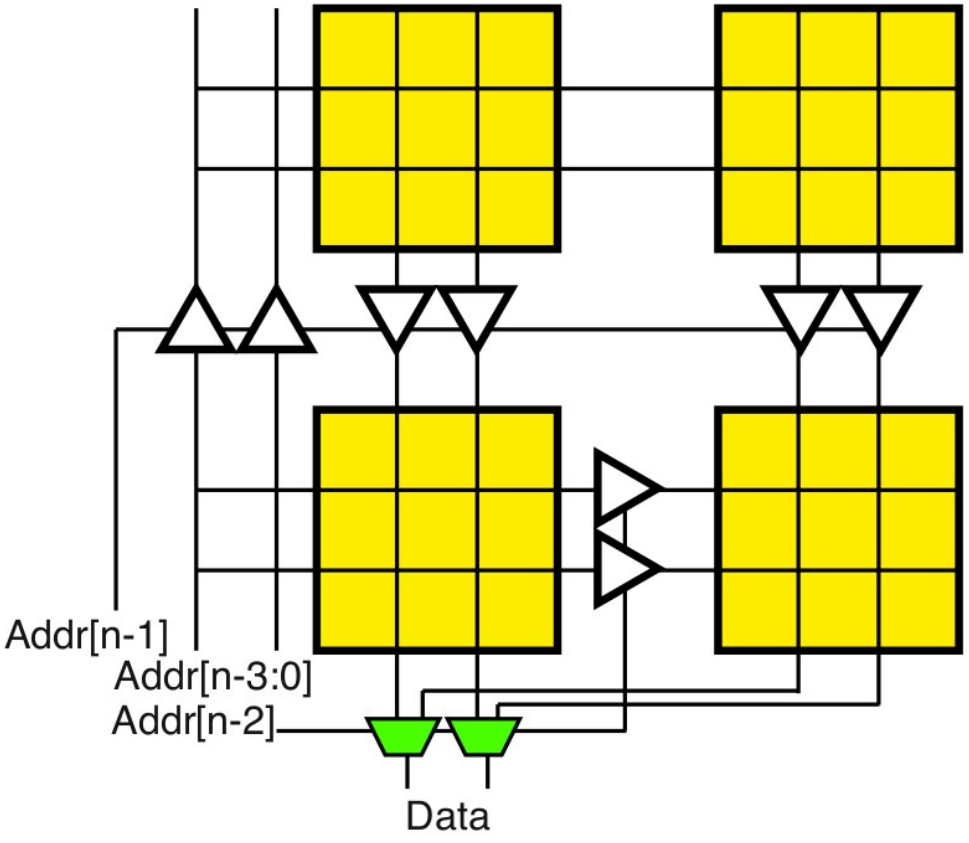


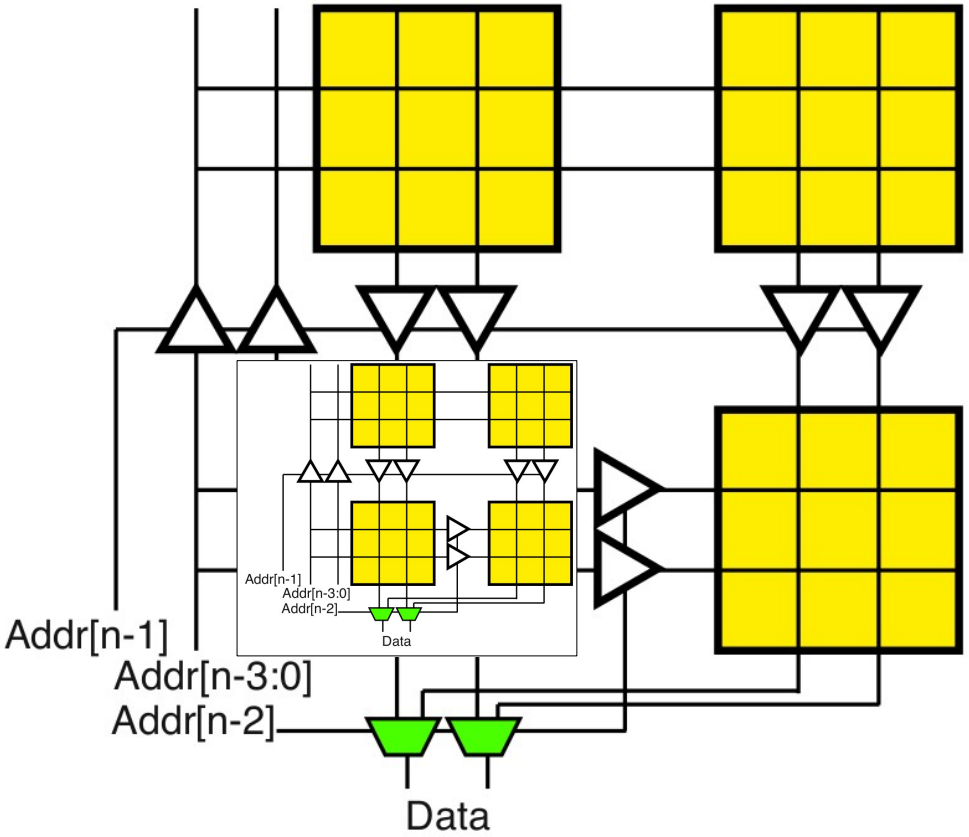
Continuous Hierarchy Memory

"Kung Fu data energy, minimizing communication energy in FPGA computations"
(Kadric *et al.* FCCM 2014)



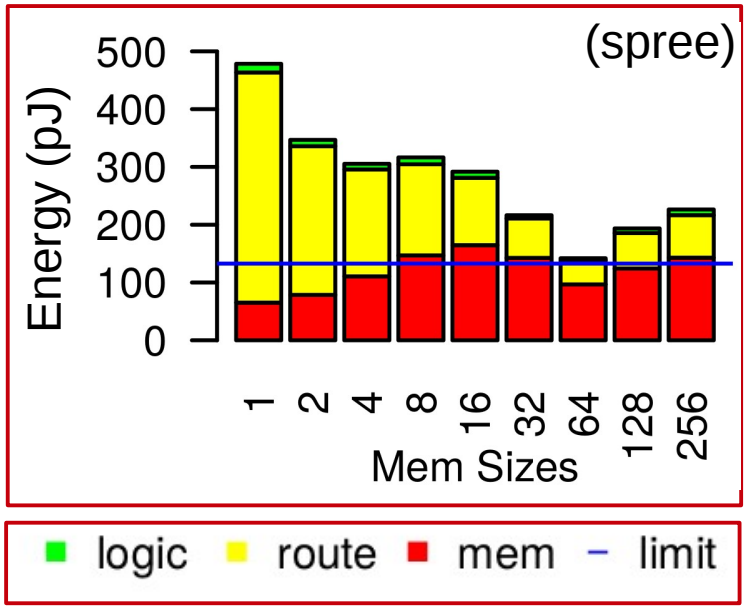
- Break down into smaller banks
- Access memory closest to I/O
- Recursively break down bank closest to I/O
- The hierarchy becomes more "*continuous*"





Outline

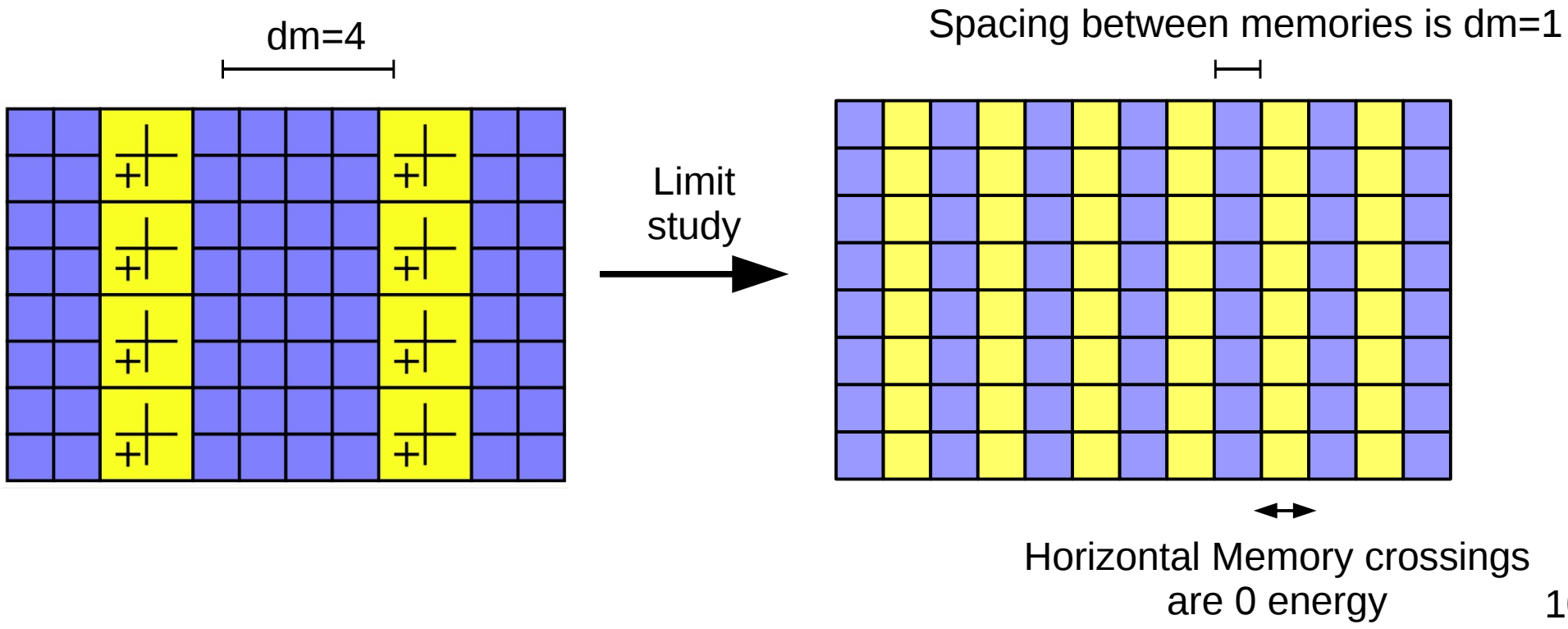
1) Motivation
2) Mismatch Issues
3) Experimental Architecture Exploration



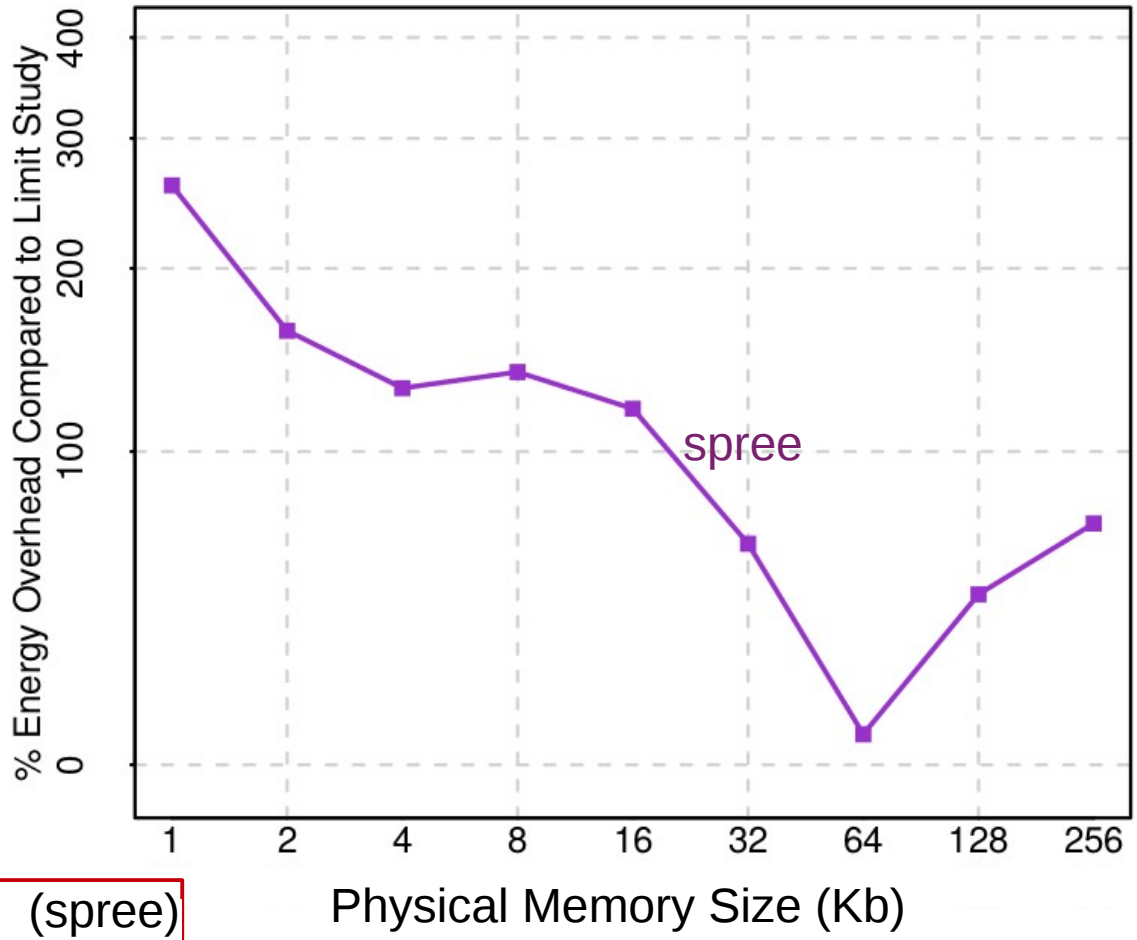
Limit Study

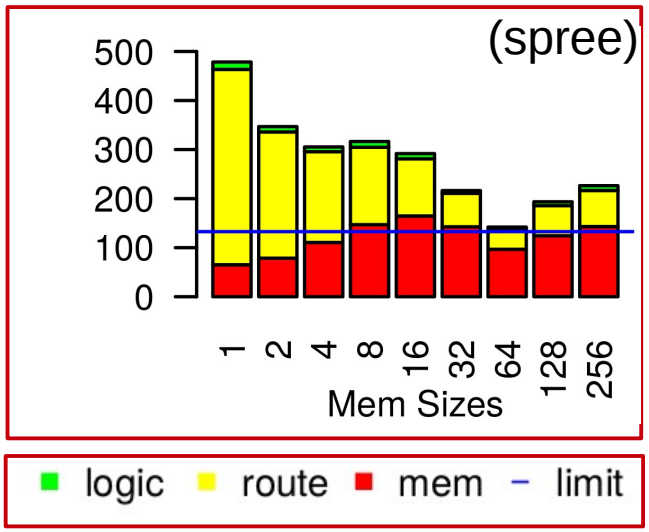
- We want memories of the right size at the right place
  - Assume the right size when calculating energy
  - Place them everywhere (every other column)

    → pretend horizontal memory crossings are free

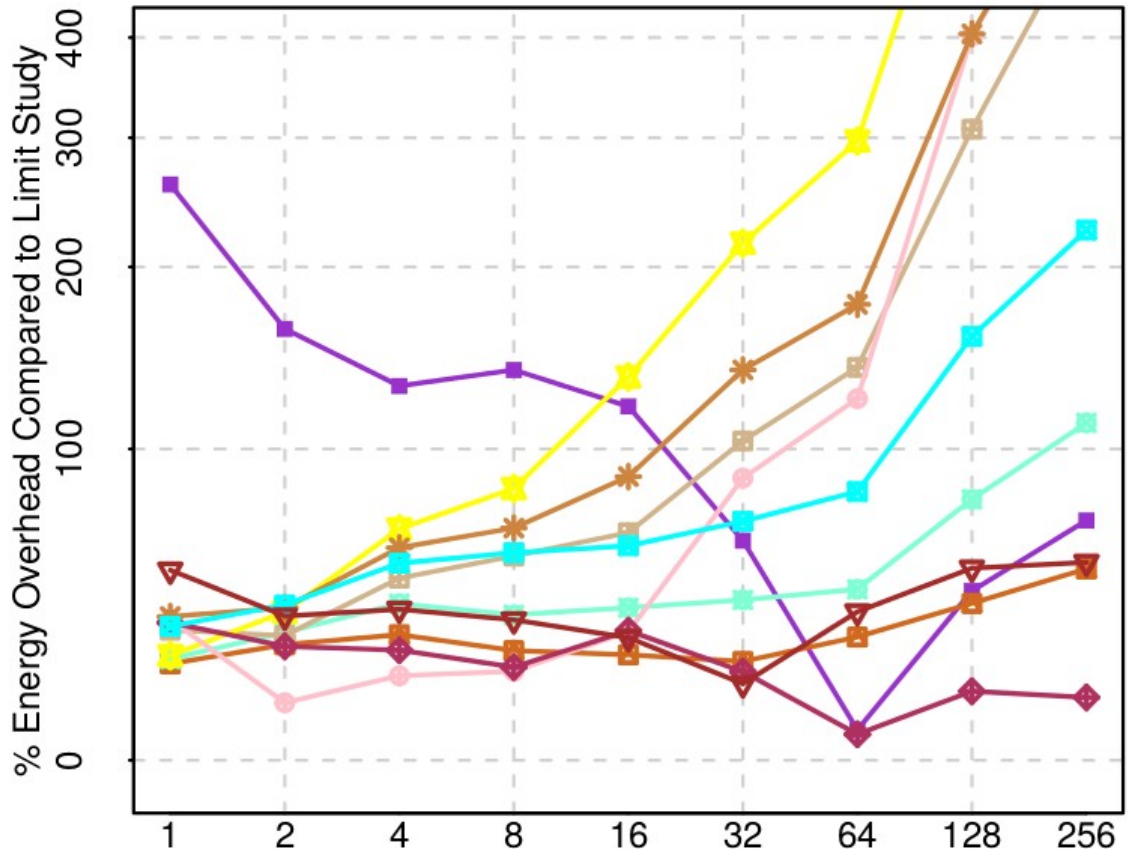


Single Memory Sweep

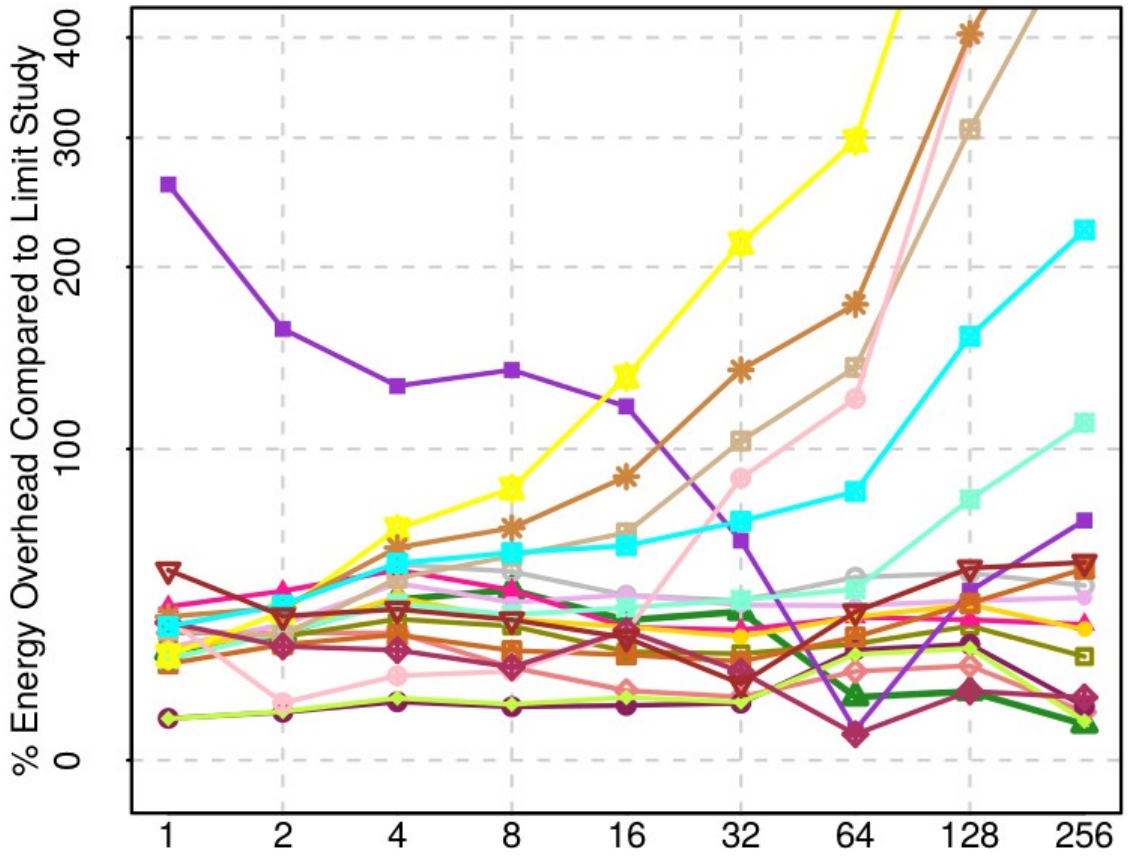




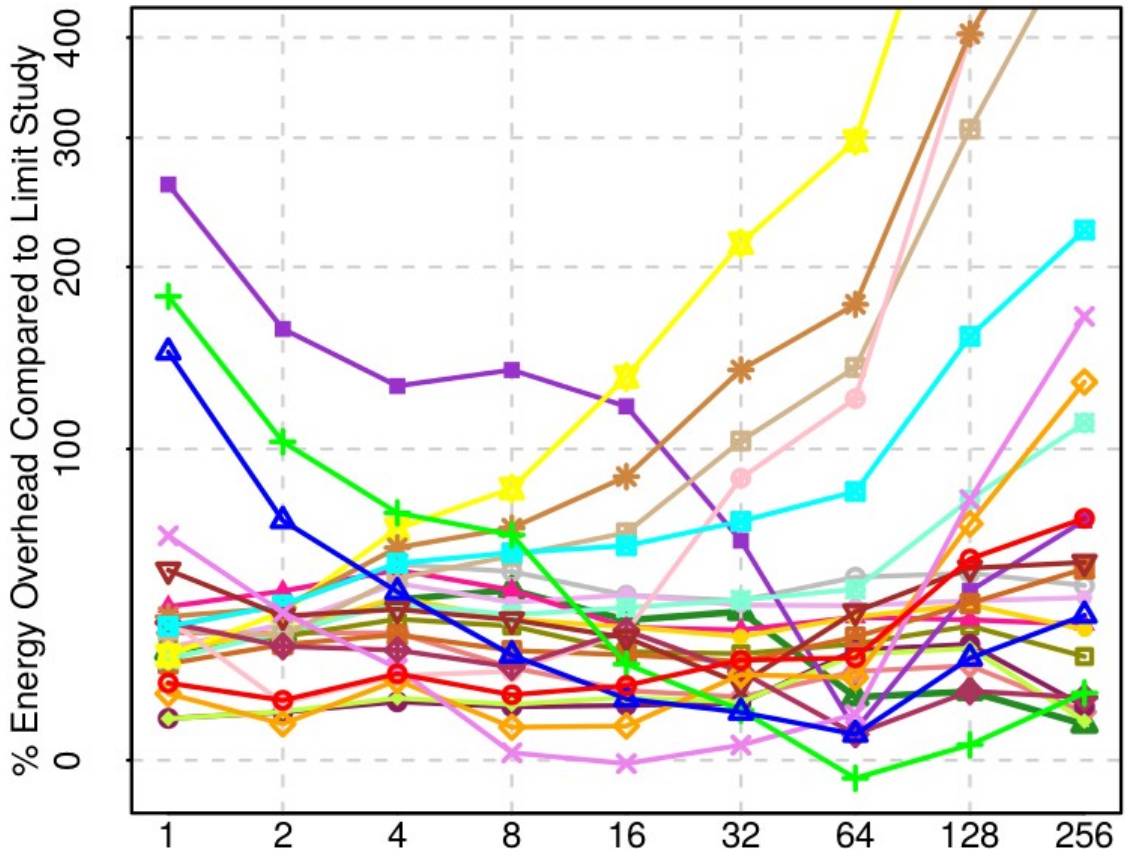
Single Memory Sweep



Single Memory Sweep



Single Memory Sweep

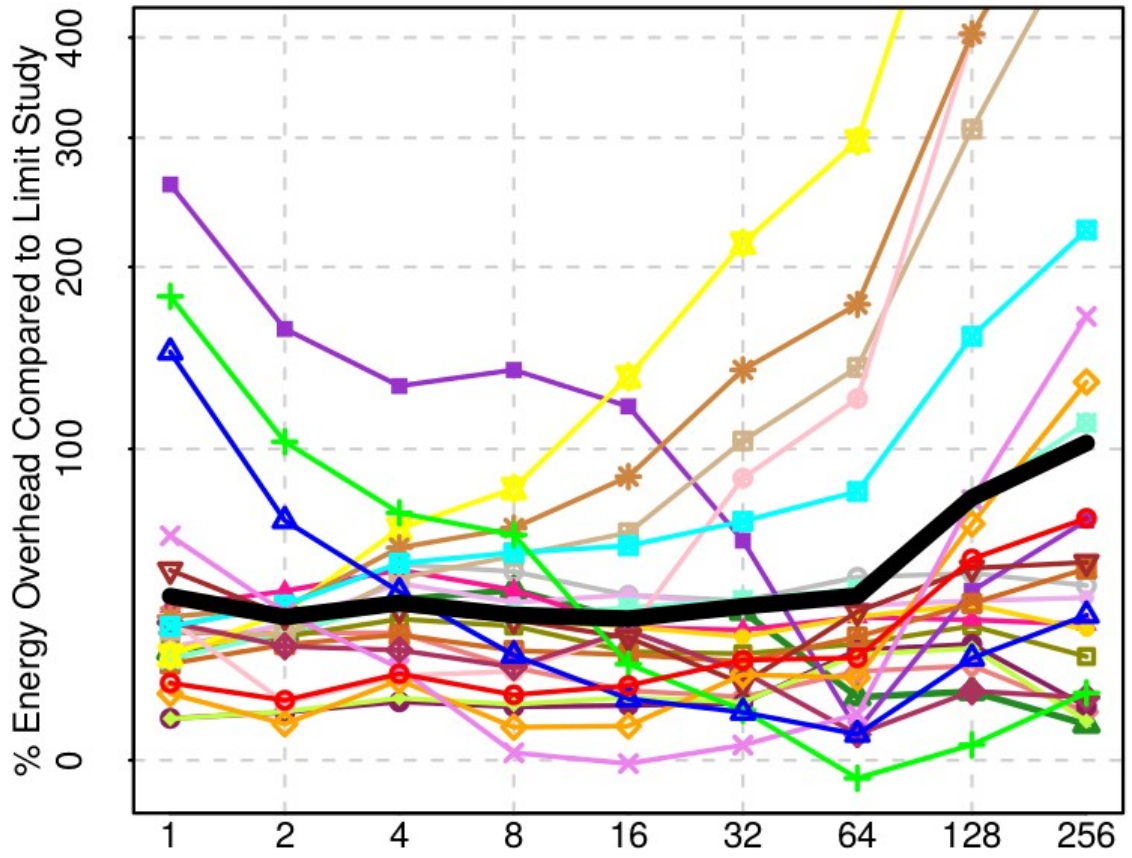


Custom
- FFT8K
- MMul128
- GMM128
- Sort8K
- WFilter128

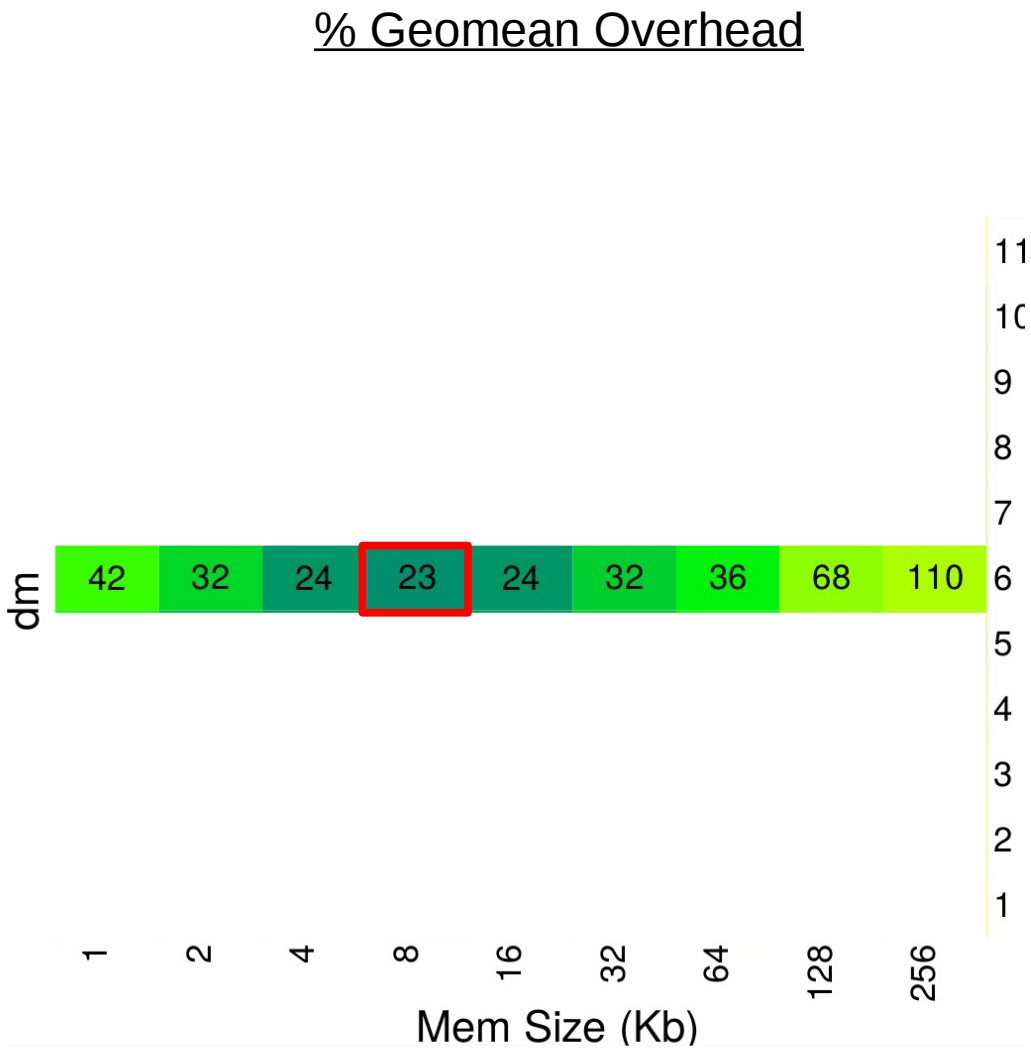
VTR with memory
- boundtop
- ch_intrinsics
- LU8PEEng
- mcml
- mkDelayWorker32B
- mkPktMerge
- mkSMAdapter4B
- or1200
- raygentop
- spree

VTR without memory
- bgm
- blob_merge
- diffeq1
- diffeq2
- sha
- stereovision0
- stereovision1
- stereovision2
- stereovision3

Single Memory Sweep

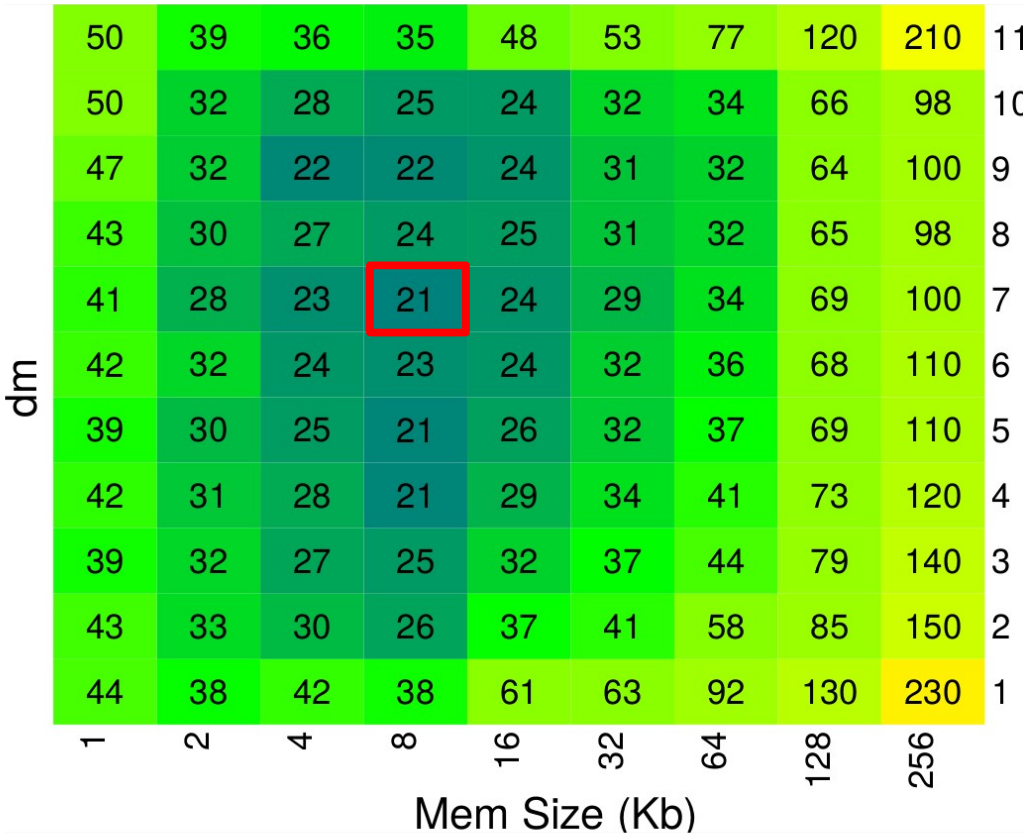


Full Architectural Sweep



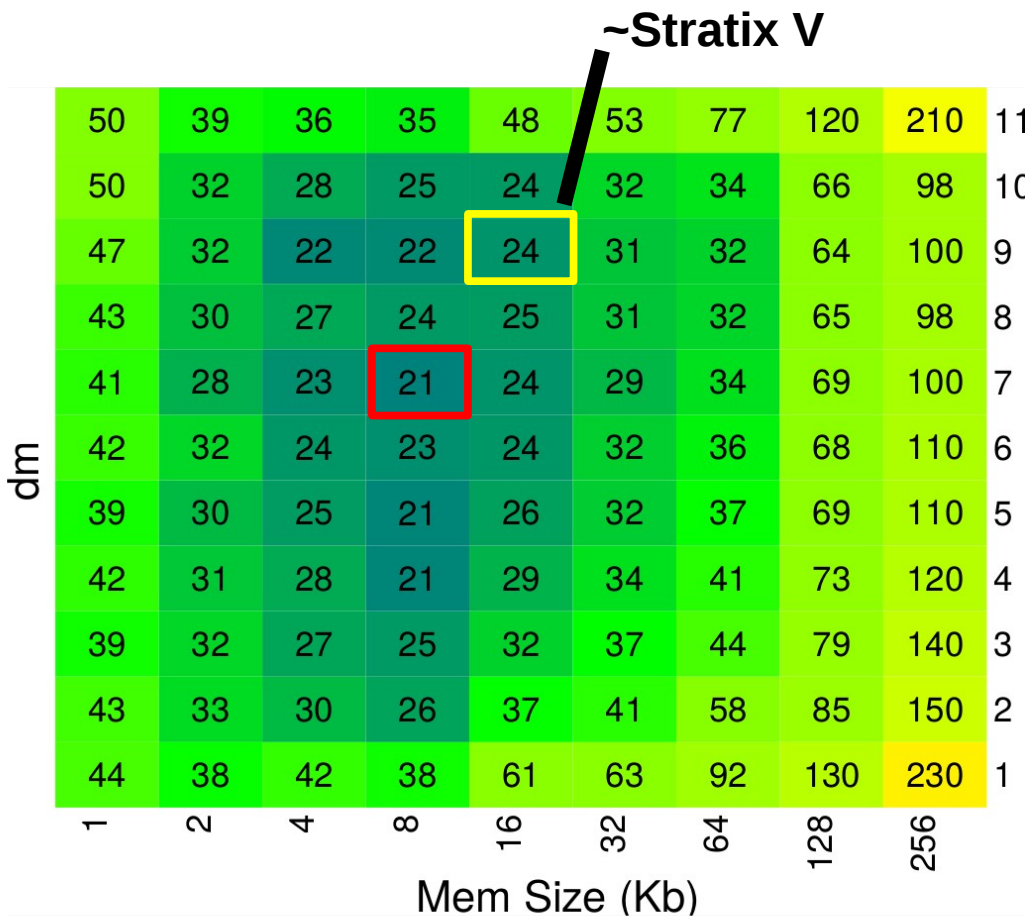
Full Architectural Sweep

% Geomean Overhead



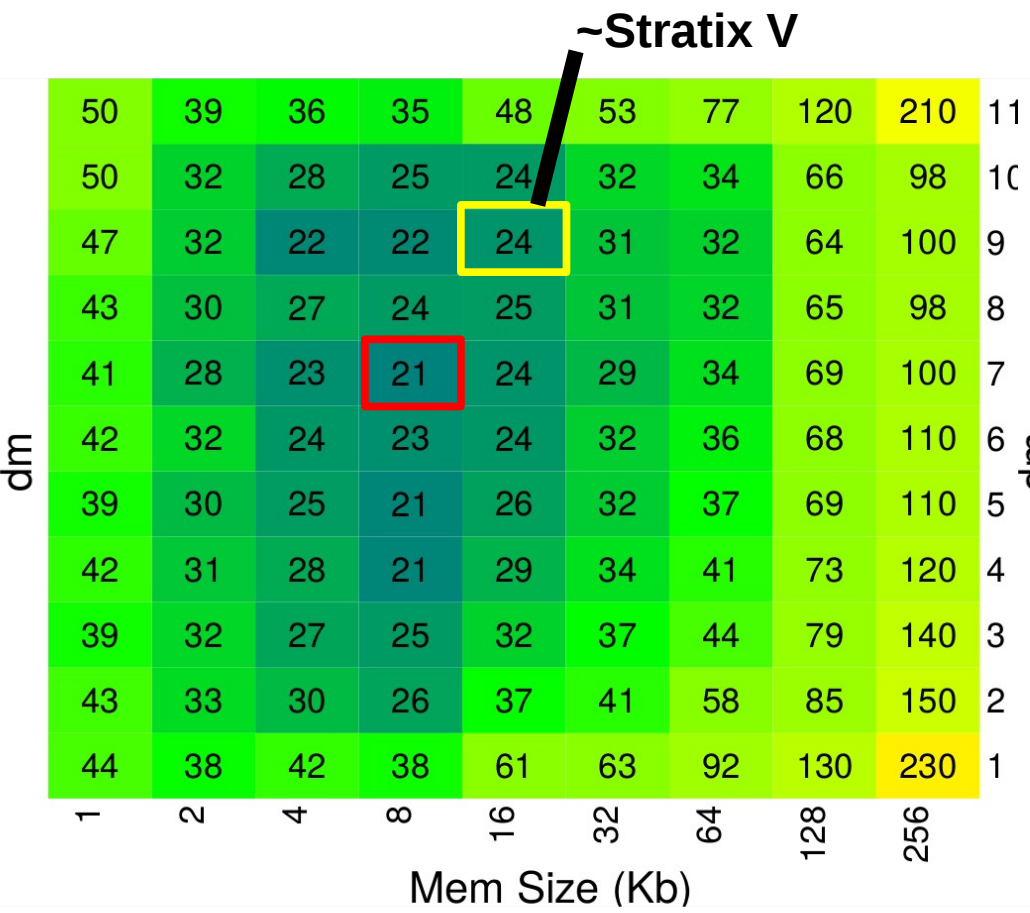
Full Architectural Sweep

% Geomean Overhead

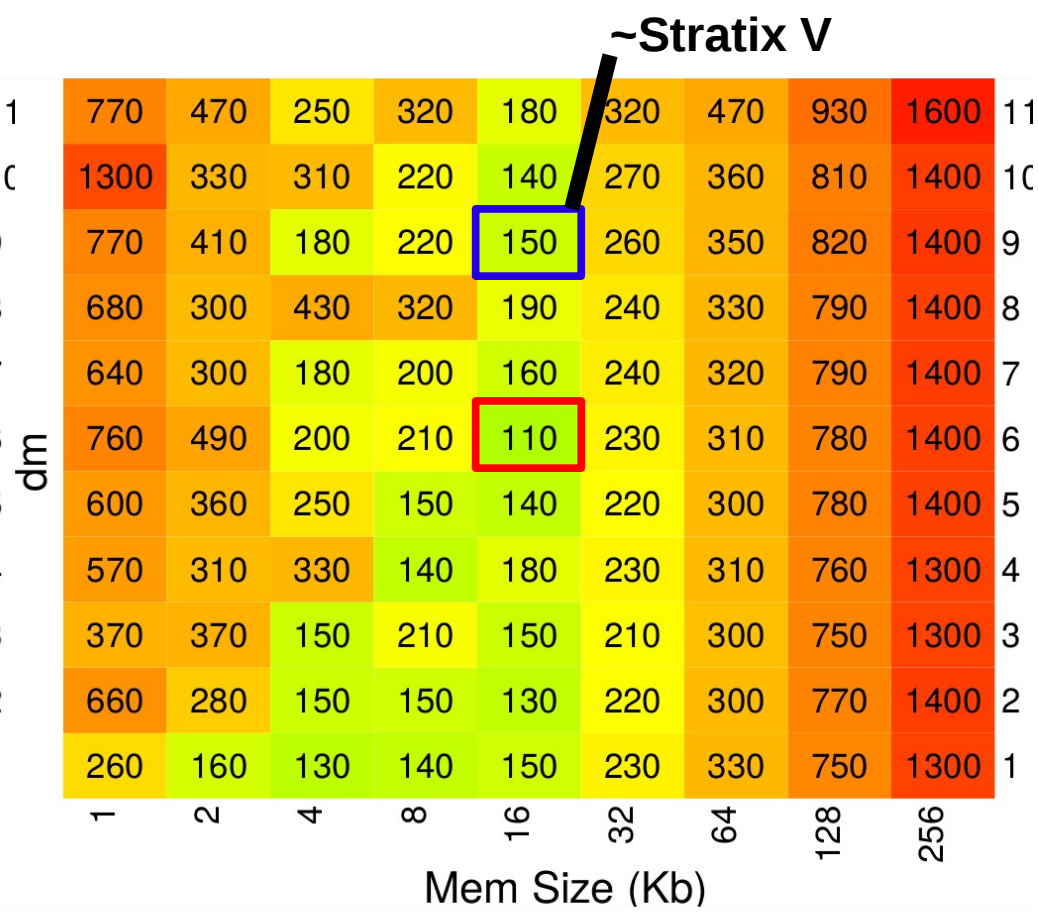


Full Architectural Sweep

% Geomean Overhead



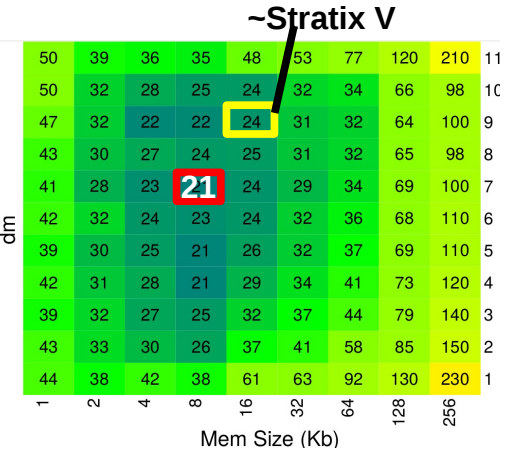
% Worst-case Overhead



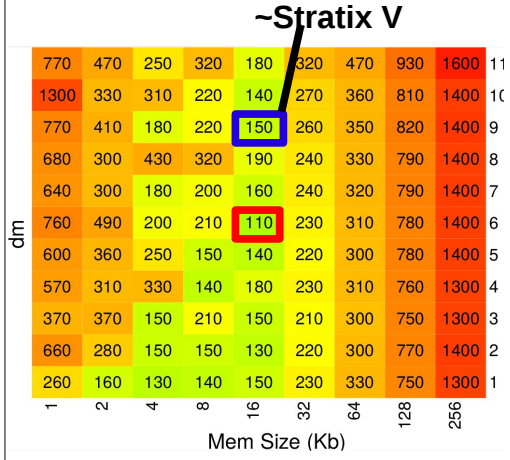
- Larger overheads
- Smaller optimum region
- Mem Size has more impact than dm

Full Architectural Sweep

% Geomean Overhead



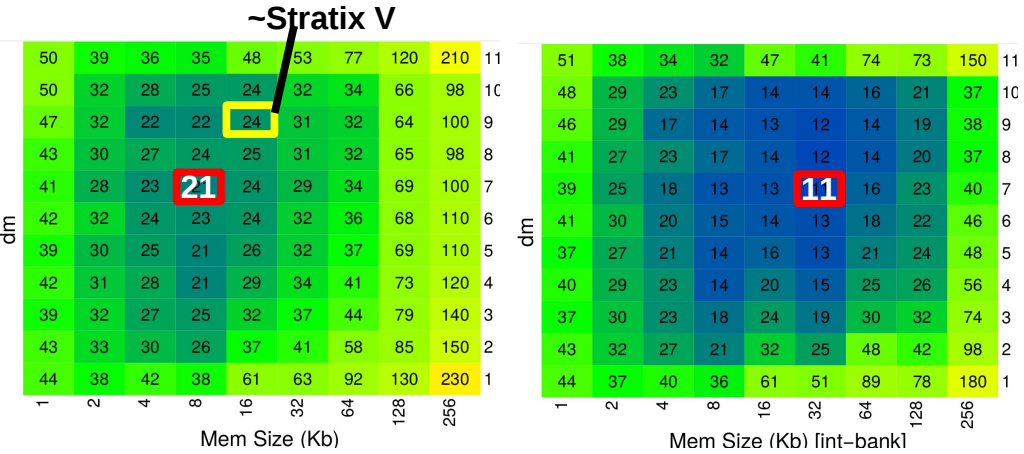
% Worst-case Overhead



- Next: 4 sets of architecture styles to explore:
  - 1 or 2 memory size(s), with or without CHM

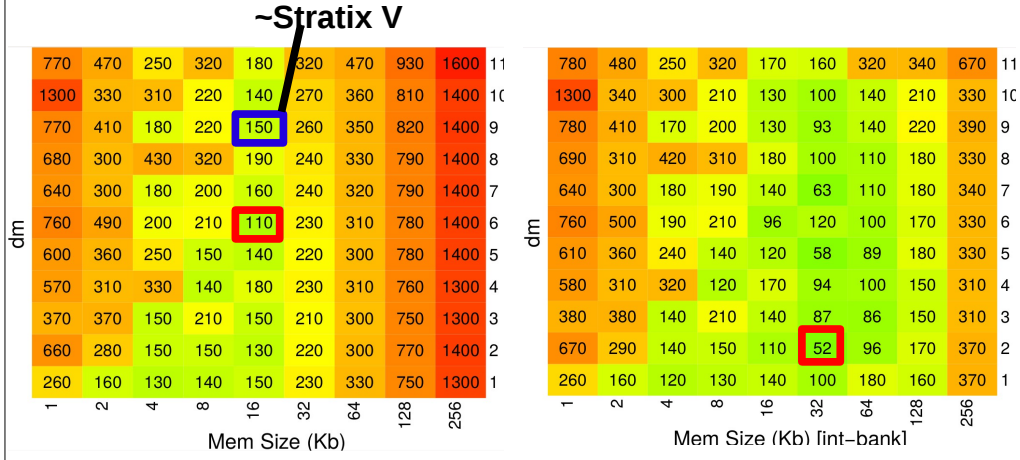
Full Architectural Sweep

% Geomean Overhead



[With CHM]

% Worst-case Overhead

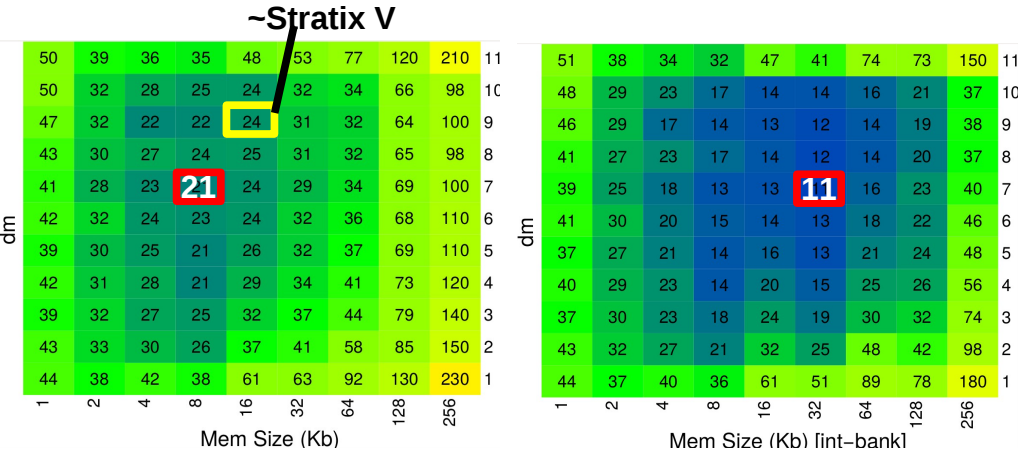


[With CHM]

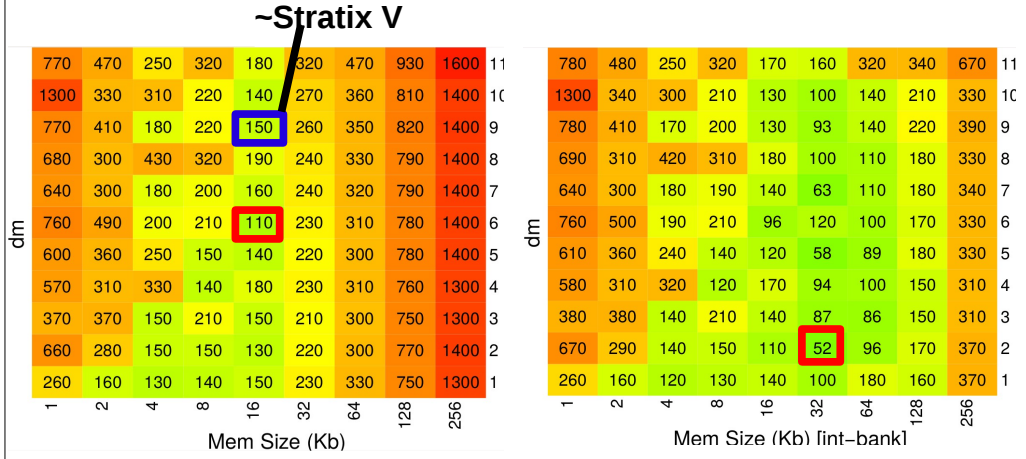
- CHM:
  - Extends optimum region
  - Shifts towards larger memories
  - Absolute value of minimum is reduced by ~2x
  - Area is increased

Full Architectural Sweep

% Geomean Overhead

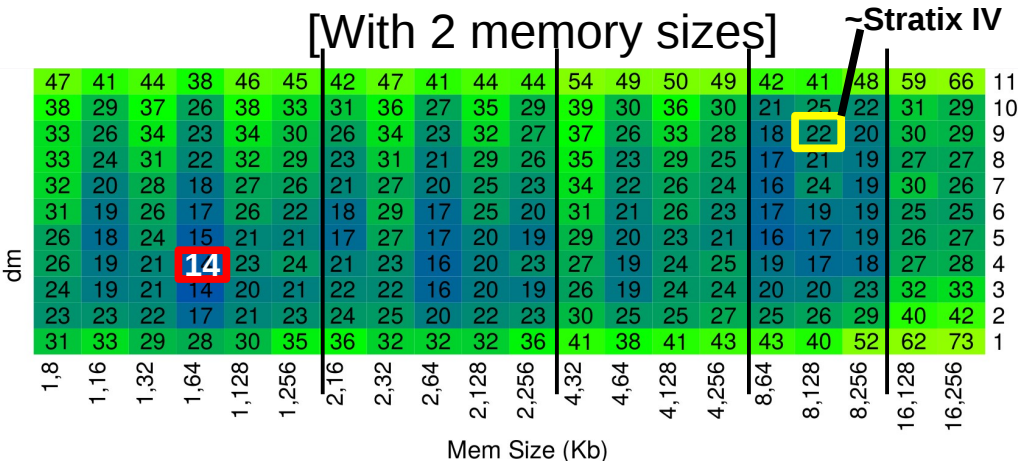


% Worst-case Overhead



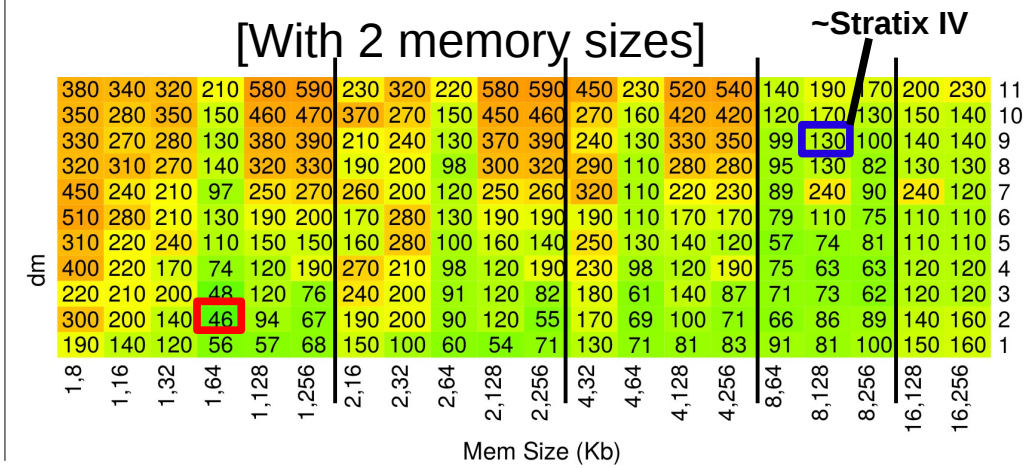
[With CHM]

[With 2 memory sizes]



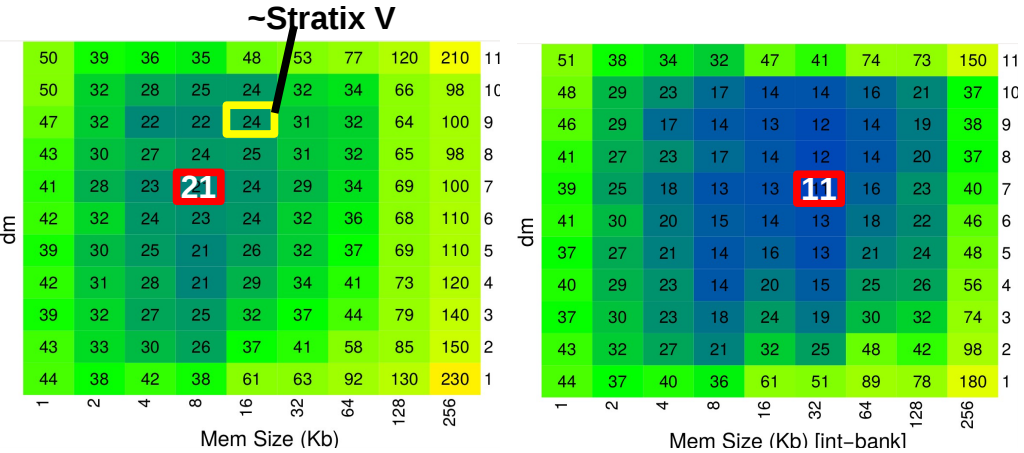
[With CHM]

[With 2 memory sizes]



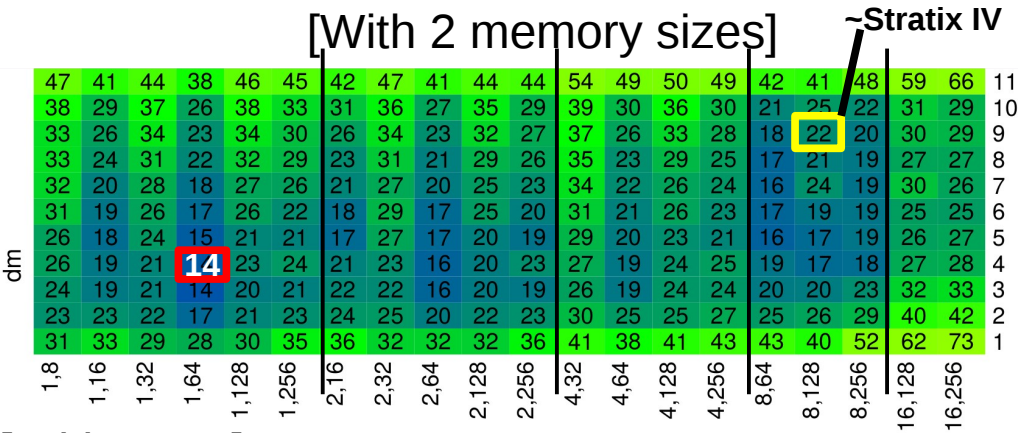
Full Architectural Sweep

% Geomean Overhead



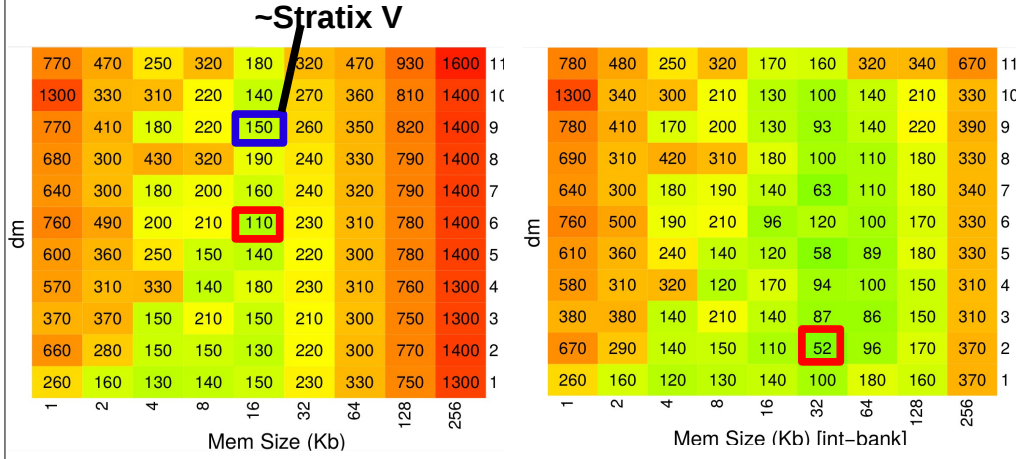
[With CHM]

[With 2 memory sizes]



[With CHM]

% Worst-case Overhead

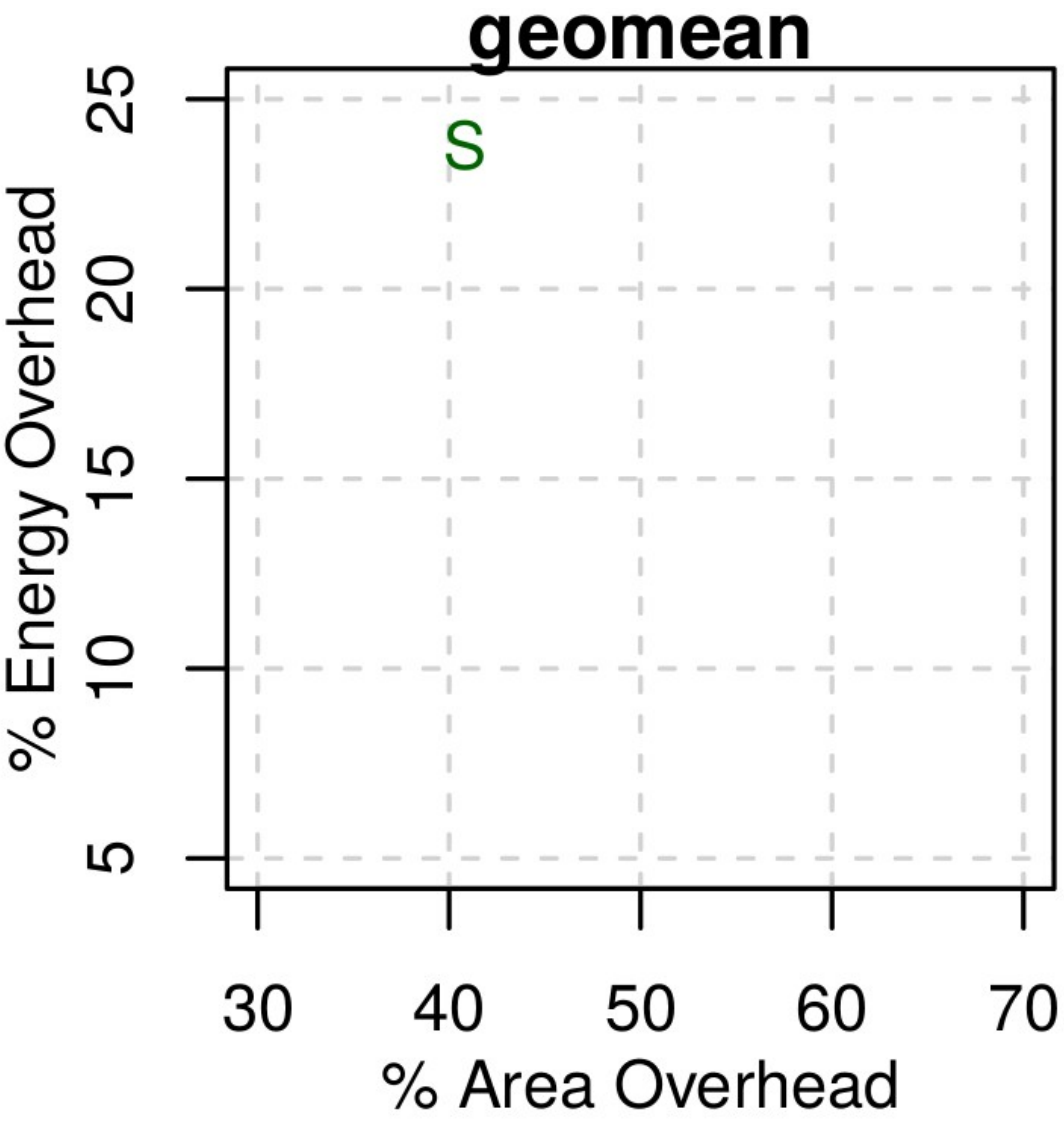


[With CHM]

[With 2 memory sizes]

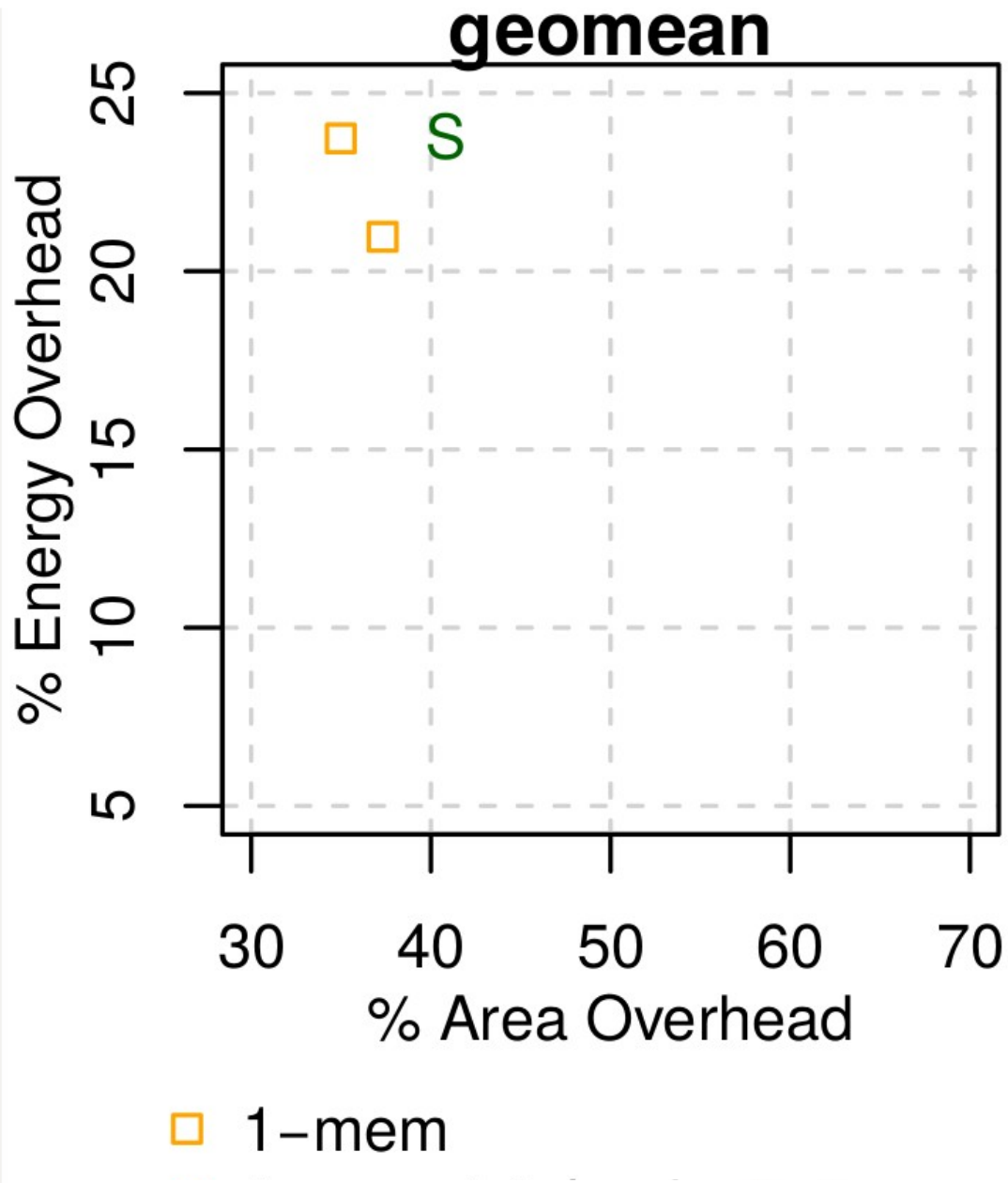
[With CHM]

Area-Energy Trade-off

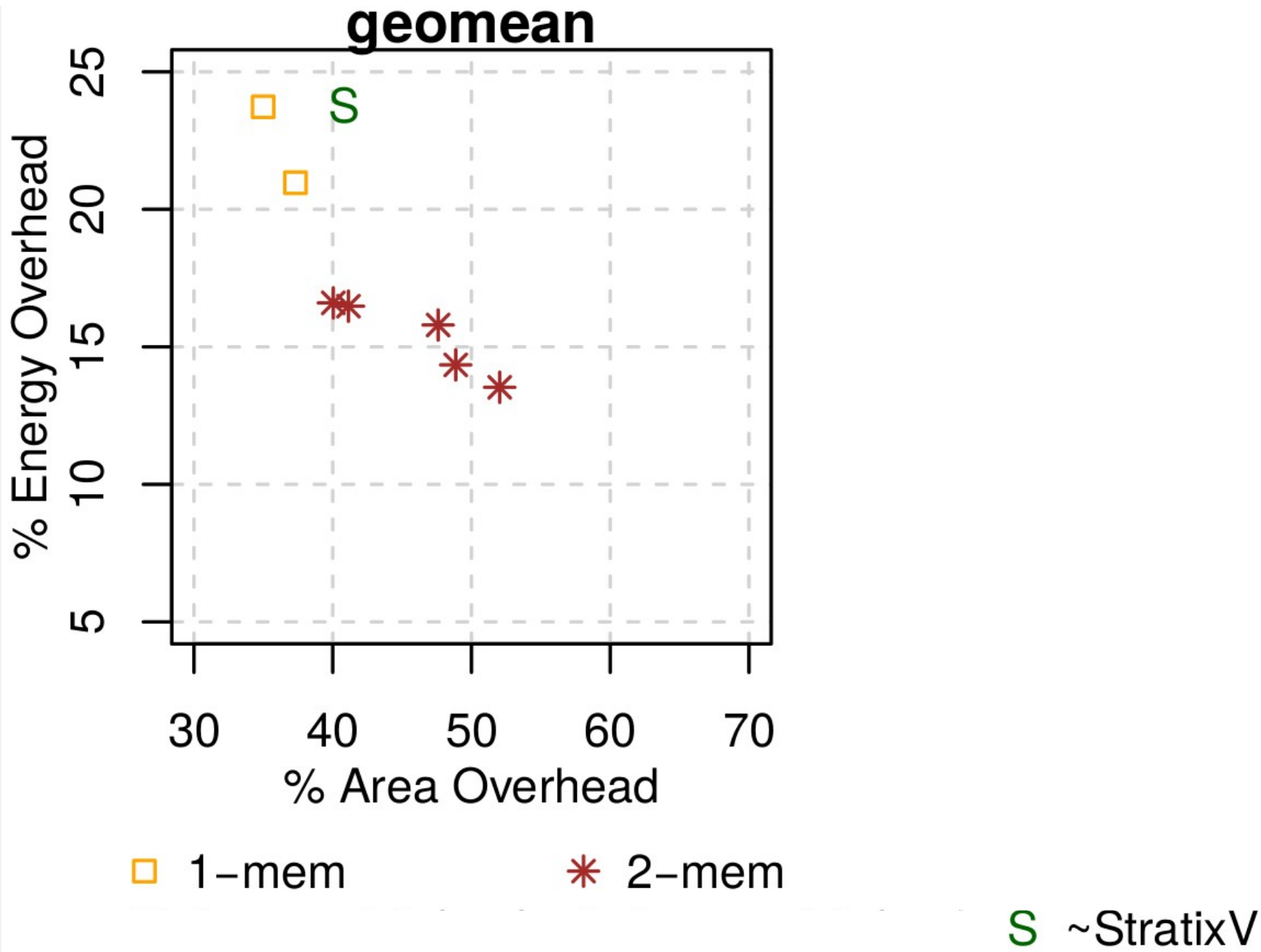


S ~StratixV

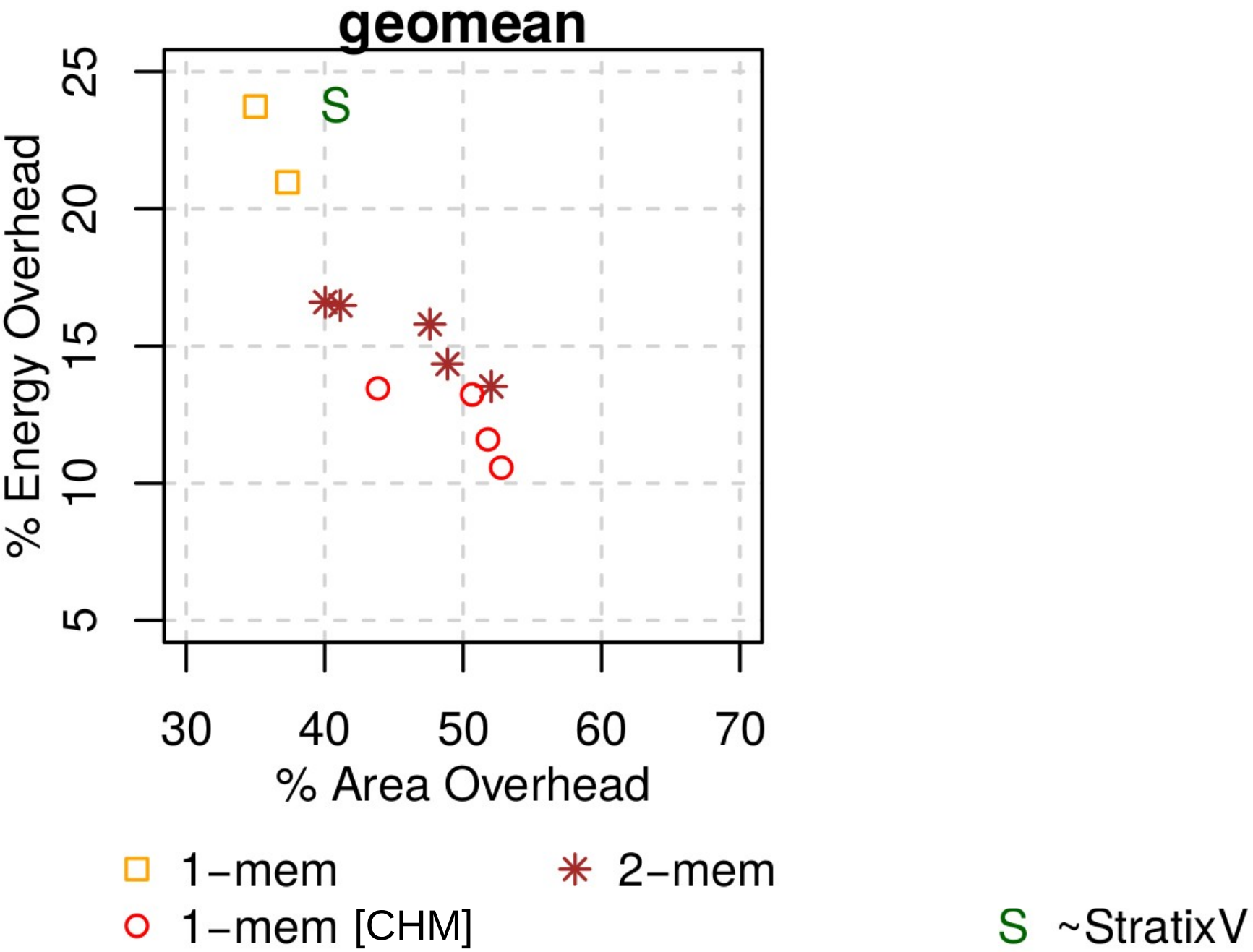
Area-Energy Trade-off



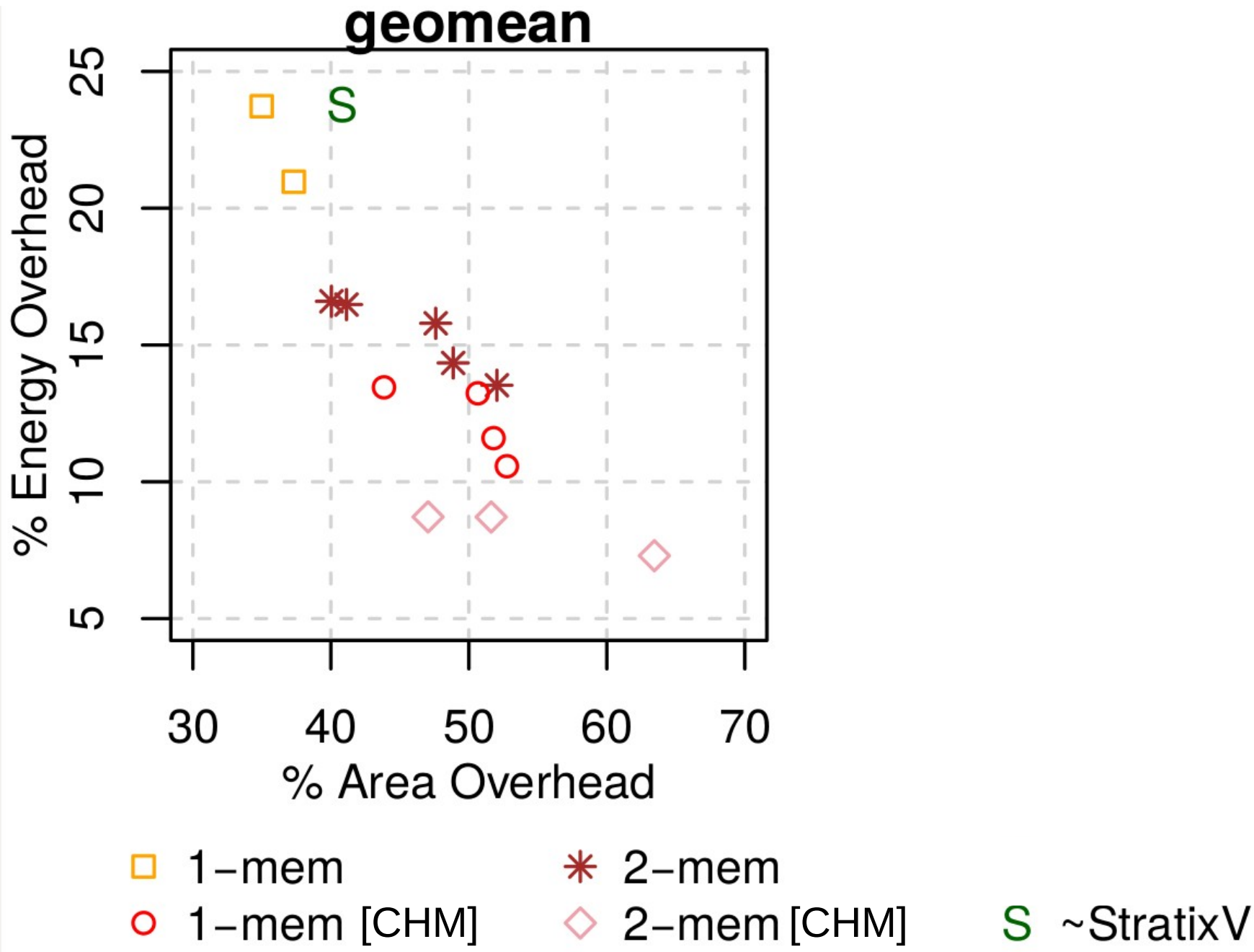
Area-Energy Trade-off



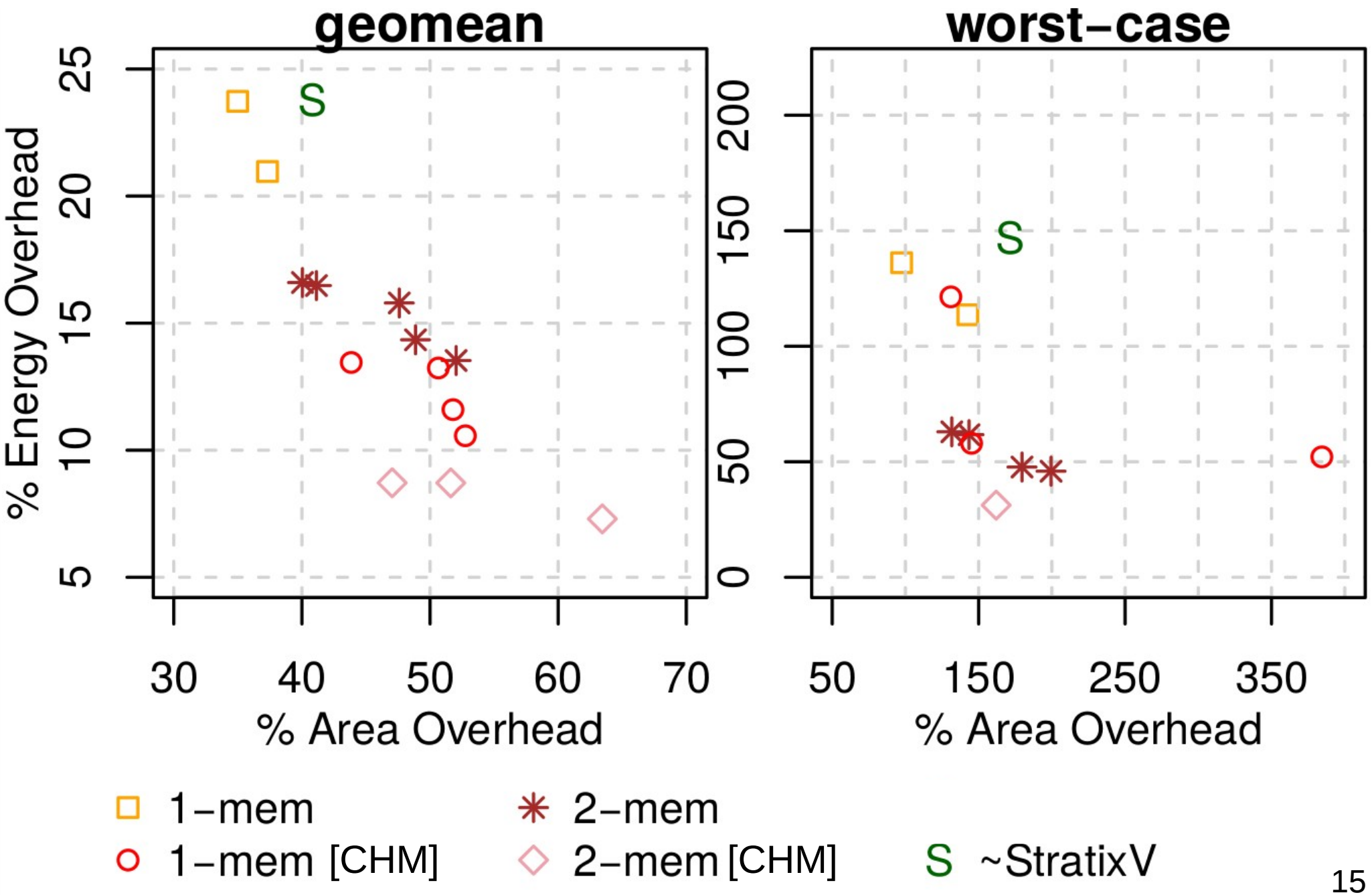
Area-Energy Trade-off



Area-Energy Trade-off



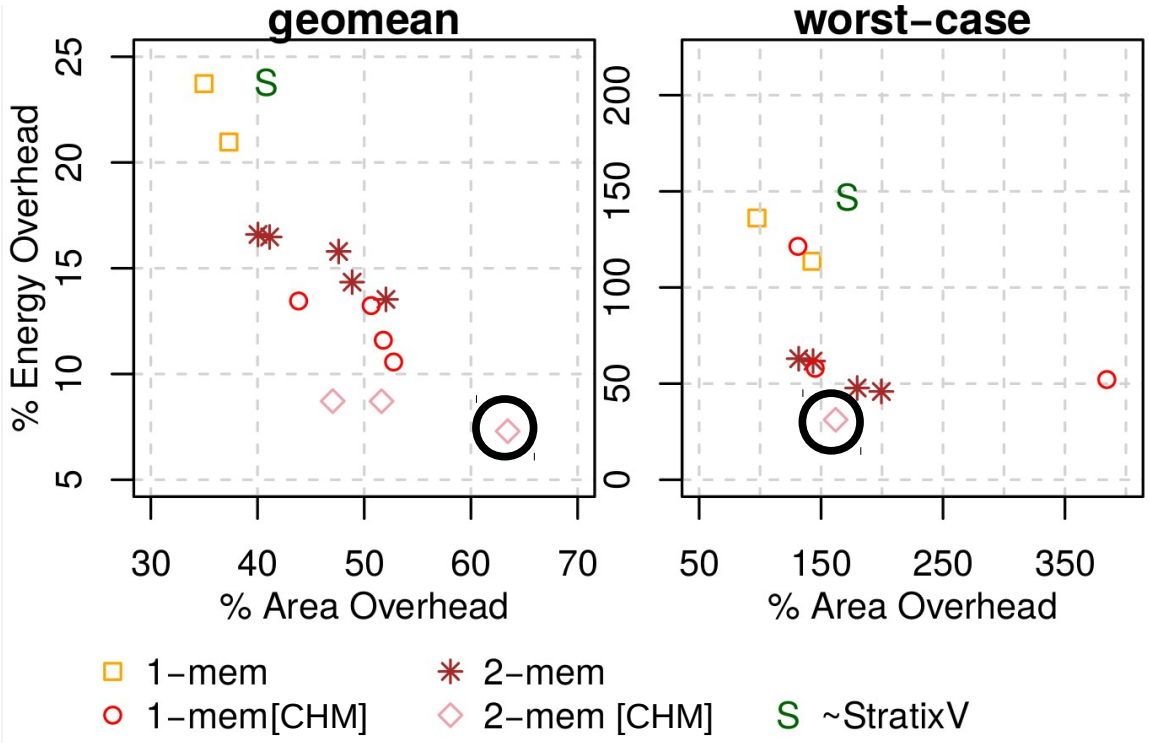
Area-Energy Trade-off



Conclusions

- Energy-optimum is different from area-optimum:
  - Multiple memory levels
  - Continuous Hierarchy Memories (CHM)
  - Placed more frequently than on commercial FPGAs
- 8-32Kb are good memory sizes in general

- Best geomean energy:
  - 8Kb and 128Kb CHM, dm=5
  - 7.3% overhead
- Best worst-case energy:
  - 8Kb and 256Kb CHM, dm=4
  - 31% overhead



Questions?

Power-Optimized Memory Mapping

($M_{arch} < M_{app}$)

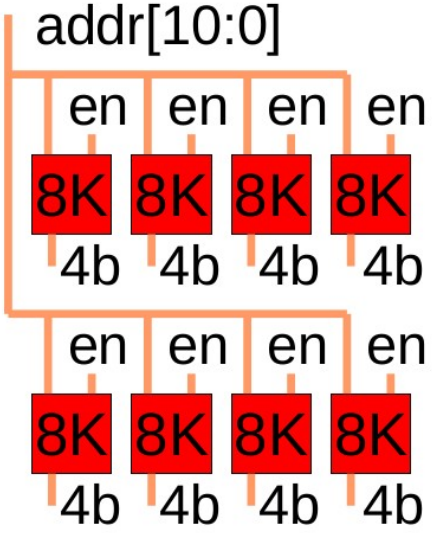
- Memories on FPGAs have a native shape, *e.g.* 8K=256x32
- Can be used in different modes:
  - 256x32
  - 512x16
  - 1024x8
  - 2048x4
  - 4096x2
  - 8192x1
- Each mode costs the same (cost of native shape, 256x32)

Power-Optimized Memory Mapping

($M_{arch} < M_{app}$)

- *e.g.*:
  - M_{app} = 2Kx32
  - M_{arch} = 256x32

(8K in 2048x4 mode)



Delay-optimized

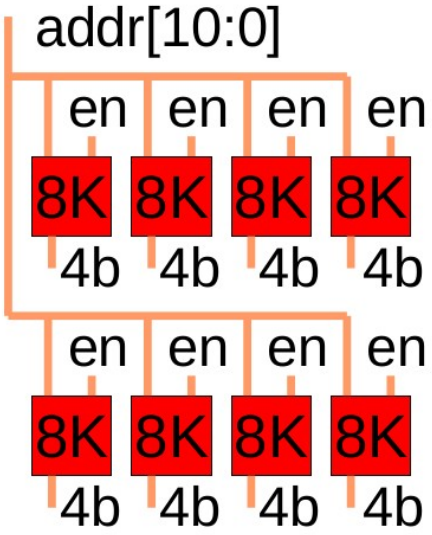
"Power-efficient RAM mapping algorithms for FPGA embedded memory blocks"
(Tessier *et al.* IEEE Trans. On CAD 2007)

Power-Optimized Memory Mapping

($M_{arch} < M_{app}$)

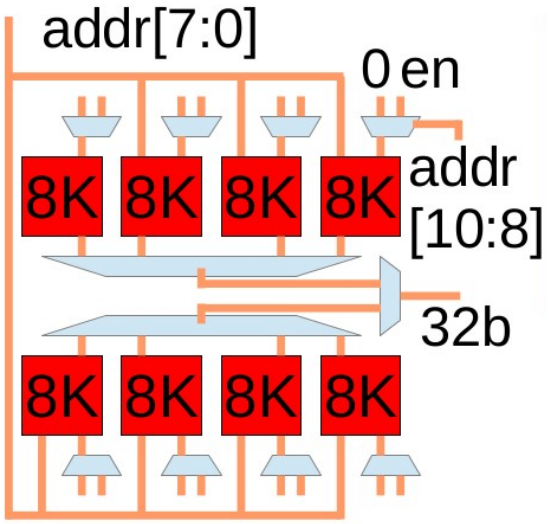
- *e.g.*:
  - M_{app} = 2Kx32
  - M_{arch} = 256x32

(8K in 2048x4 mode)



Delay-optimized

(8K in 256x32 mode)



Power-optimized

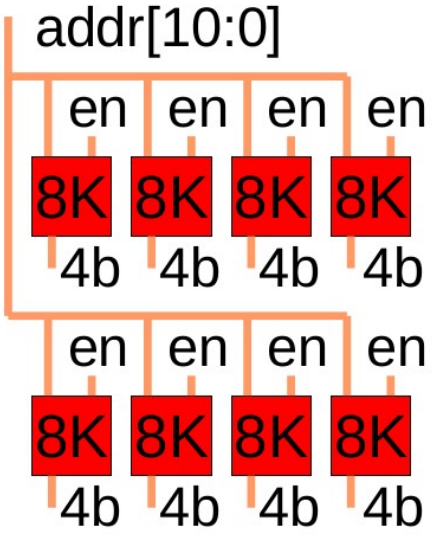
"Power-efficient RAM mapping algorithms for FPGA embedded memory blocks"
(Tessier *et al.* IEEE Trans. On CAD 2007)

Power-Optimized Memory Mapping

($M_{arch} < M_{app}$)

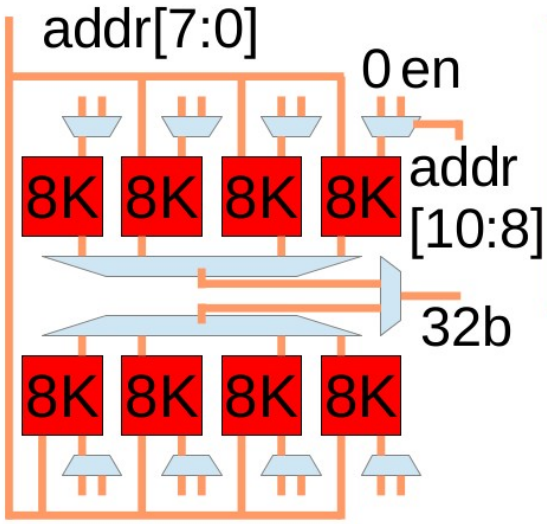
- *e.g.*:
  - M_{app} = 2Kx32
  - M_{arch} = 256x32

(8K in 2048x4 mode)

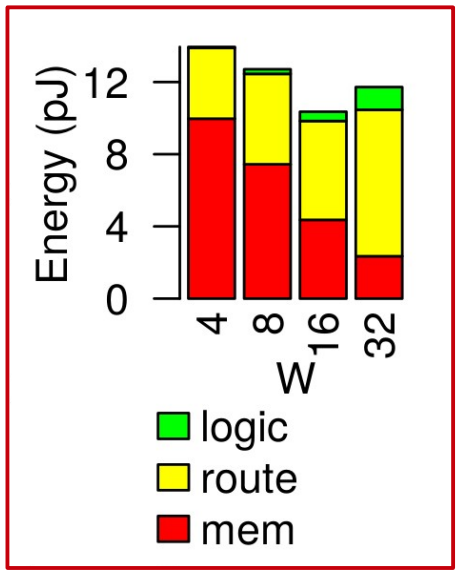


Delay-optimized

(8K in 256x32 mode)



Power-optimized



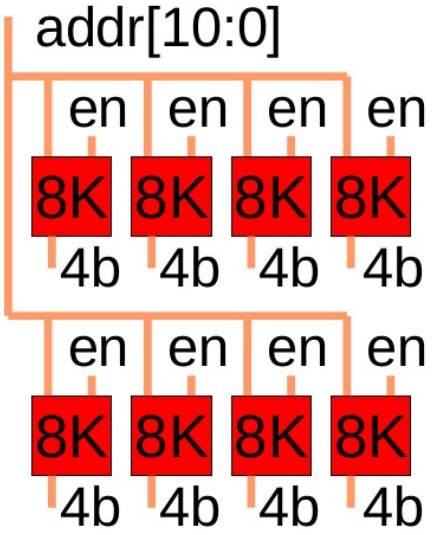
"Power-efficient RAM mapping algorithms for FPGA embedded memory blocks"
(Tessier *et al.* IEEE Trans. On CAD 2007)

Power-Optimized Memory Mapping

($M_{arch} < M_{app}$)

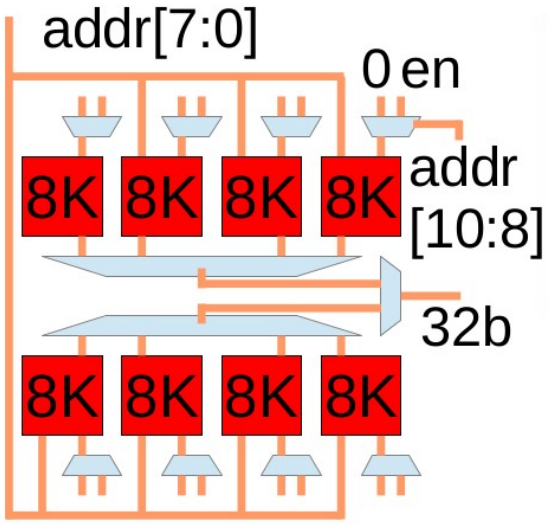
- *e.g.*:
  - M_{app} = 2Kx32
  - M_{arch} = 256x32

(8K in 2048x4 mode)

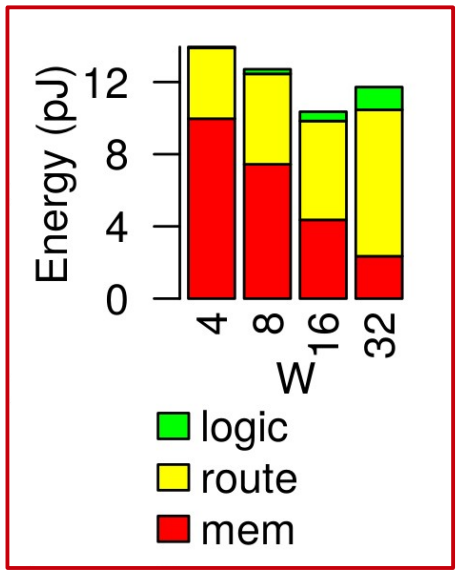


Delay-optimized

(8K in 256x32 mode)

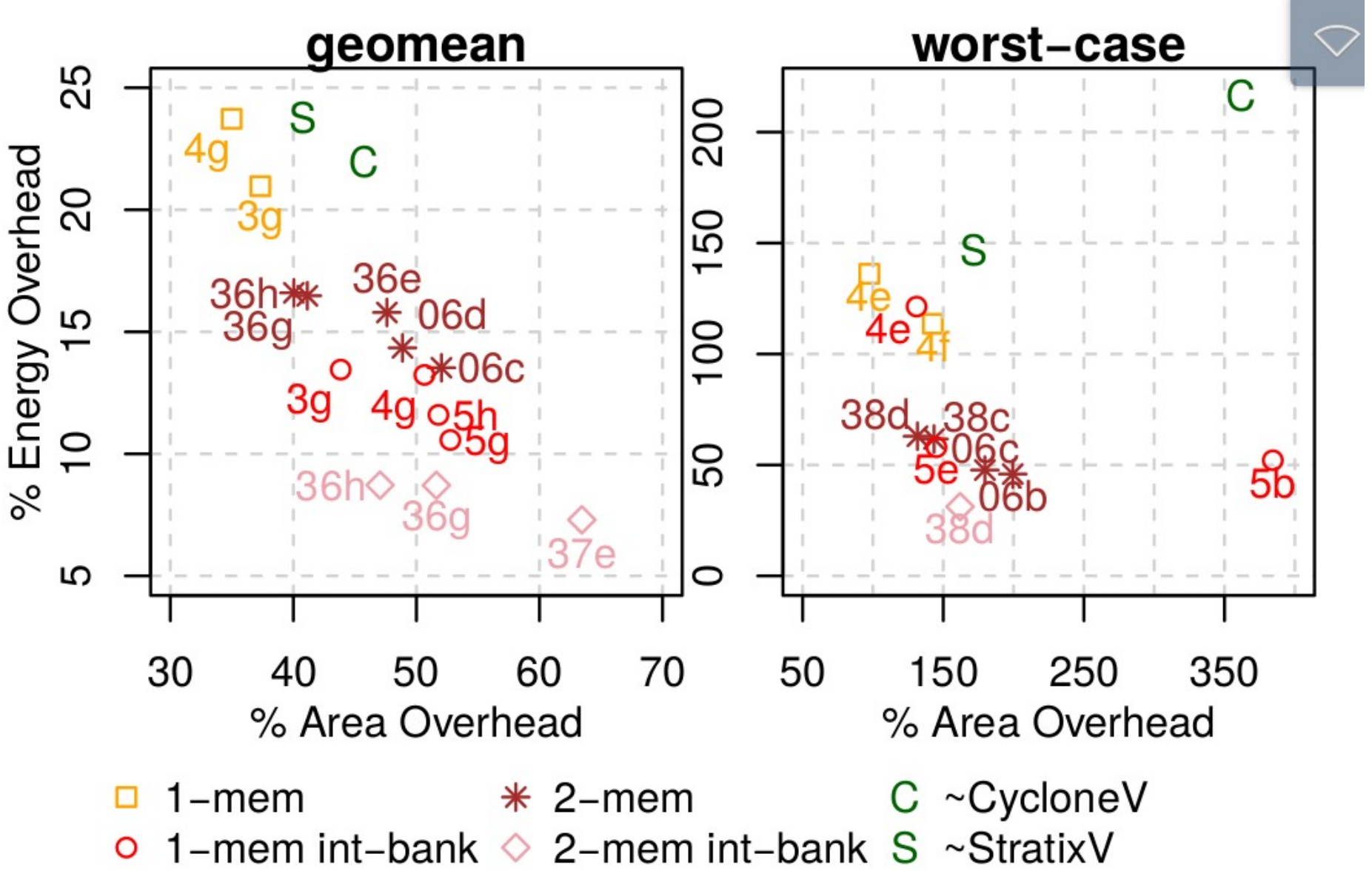


Power-optimized



"Power-efficient RAM mapping algorithms for FPGA embedded memory blocks"
(Tessier *et al.* IEEE Trans. On CAD 2007)

What we did: Integrated with VPR
Without P-opt: +4-19% geomean energy overhead, **+40-108% worst-case overhead**
P-opt source release: http://ic.ese.upenn.edu/abstracts/meme_fpga2015.html



label	0	1	2	3	4	5	6	7	8
mem size (Kb)	1	2	4	8	16	32	64	128	256

label	a	b	c	d	e	f	g	h	i	j	k
dm=	1	2	3	4	5	6	7	8	9	10	11