Sparse Matrix-Vector Multiply on the Texas Instruments C6678 Digital Signal Processor

Yang Gao and Dr. Jason D. Bakos

Application-Specific Systems, Architectures, and Processors 2013

UNIVERSITY OF SOUTH CAROLINA

TI C6678 vs. Competing Coprocessors

Coprocessor	NVIDIA Tesla K20X GPU	Intel Xeon Phi 5110p	TI C66
Peak single precision performance	3.95 Tflops/s	2.12 Tflops/s	128 Gflops/s
Memory bandwidth	250 GB/s	320 GB/s	12.8 GB/s
Power	225 W	225 W	10 W
Primary programming model	CUDA/OpenCL	OpenMP	None, but OpenMP/ OpenCL in development

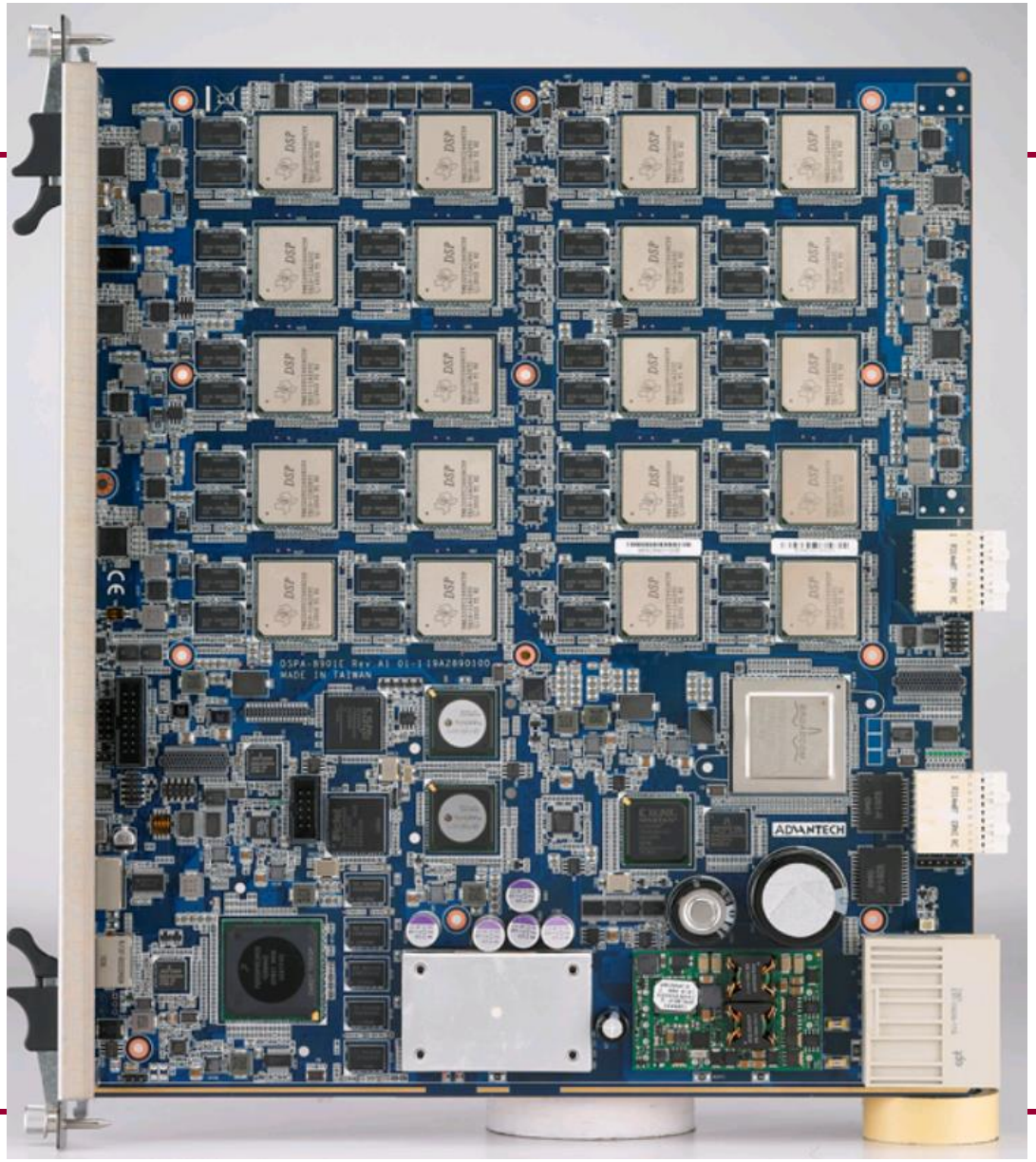
Why the C6678?

- **Unique architectural features**
  - 8 symmetric VLIW cores with SIMD instructions, up to 16 flops/cycle
  - No shared last level cache
  - 4MB on-chip shared RAM
  - L1D and L2 can be configured as cache, scratchpad, or both
  - DMA engine for parallel loading/flushing scratchpads

- **Power efficiency**
  - At 45 nm, achieves 12.8 ideal SP Gflops/Watt
  - Intel Phi [22 nm] is 9.4 Gflops/Watt
  - NVIDIA K20x [28 nm] is 17.6 Gflops/Watt

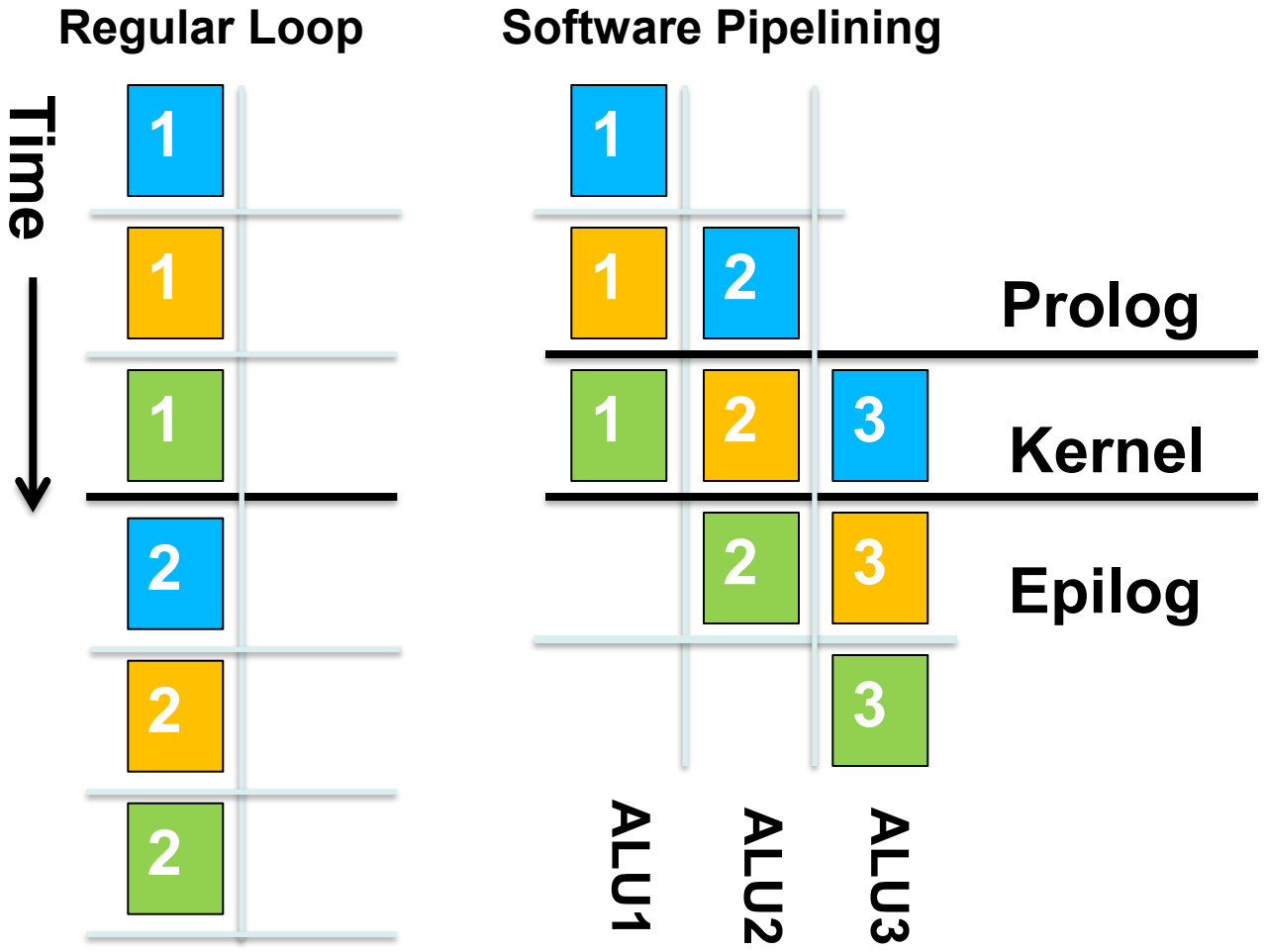
- **Fast on-chip interfaces for potential scalability**
  - 4 x Rapid IO(SRIO) 2.1: 20 Gb/s
  - 1 x Ethernet: 1 Gb/s
  - 2 x PCI-E 2.0: 10 Gb/s
  - HyperLink: 50 Gb/s

C66 Platforms



Software Pipelining

- VLIW architecture requires explicit usage of functional units
- C66 compiler uses software pipelining to maximize FU utilization
- Conditional prevents SP and lowers utilization



Sparse Matrices

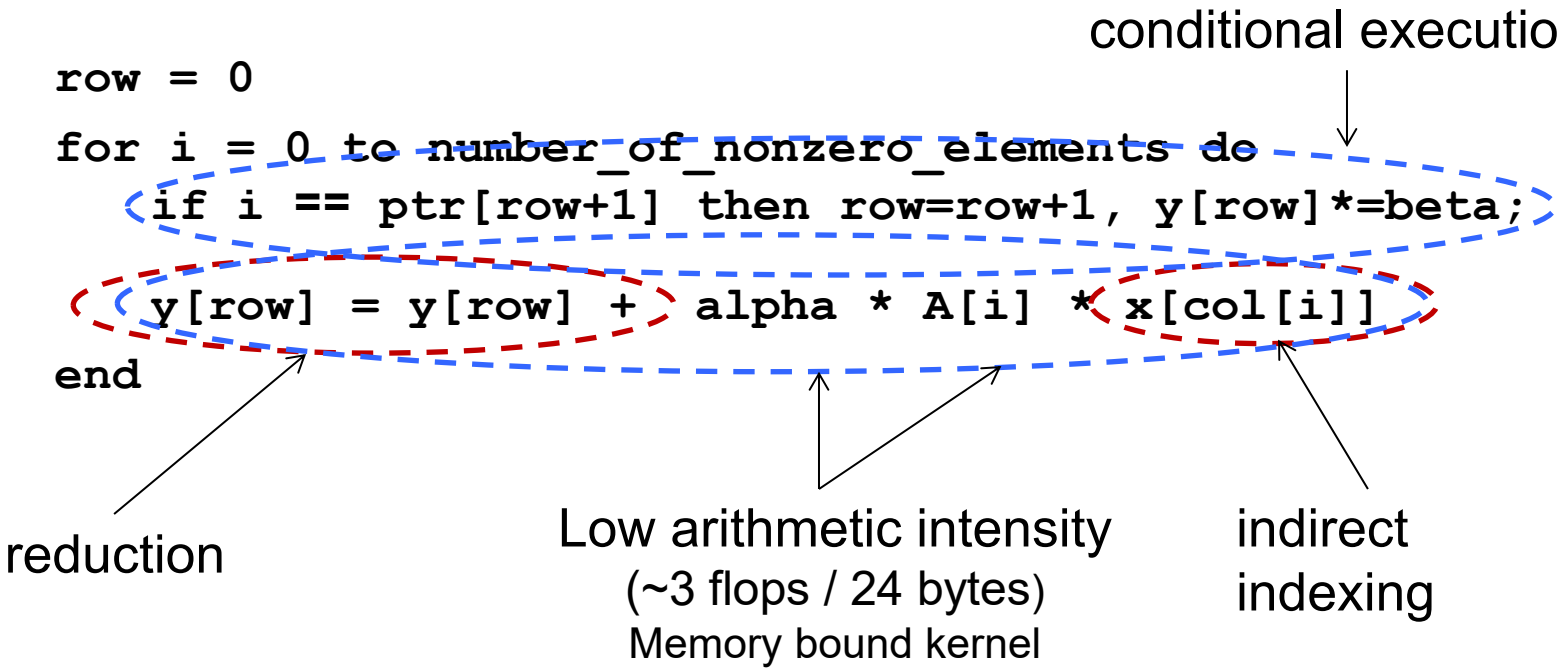
- We evaluated the C66 relative using a SpMV kernel
  - GPUs achieve only 0.6% to 6% of their peak performance with CSR SpMV
- Sparse Matrices can be very large but contain few non-zero elements
- Compressed formats are often used, e.g. Compressed Sparse Row (CSR)

$$\begin{pmatrix} 1 & -1 & 0 & -3 & 0 \\ -2 & 5 & 0 & 0 & 0 \\ 0 & 0 & 4 & 6 & 4 \\ -4 & 0 & 2 & 7 & 0 \\ 0 & 8 & 0 & 0 & -5 \end{pmatrix}$$

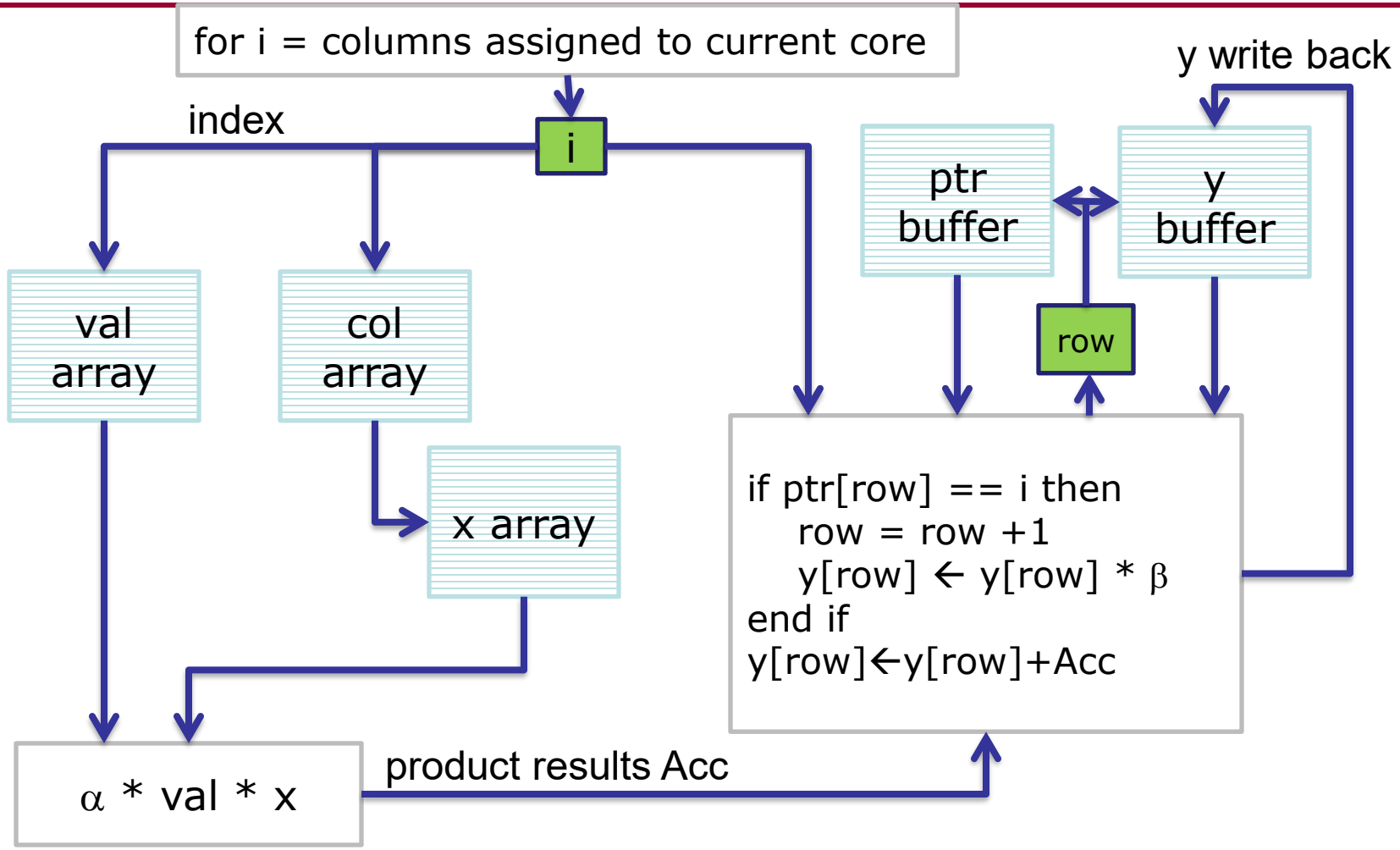
val	(1	-1	-3	-2	5	4	6	4	-4	2	7	8	-5)
col	(0	1	3	0	1	2	3	4	0	2	3	1	4)
ptr	(0	3	5	8	11	13)							

Sparse Matrix-Vector Multiply

- Code for y = **A**αx + βy

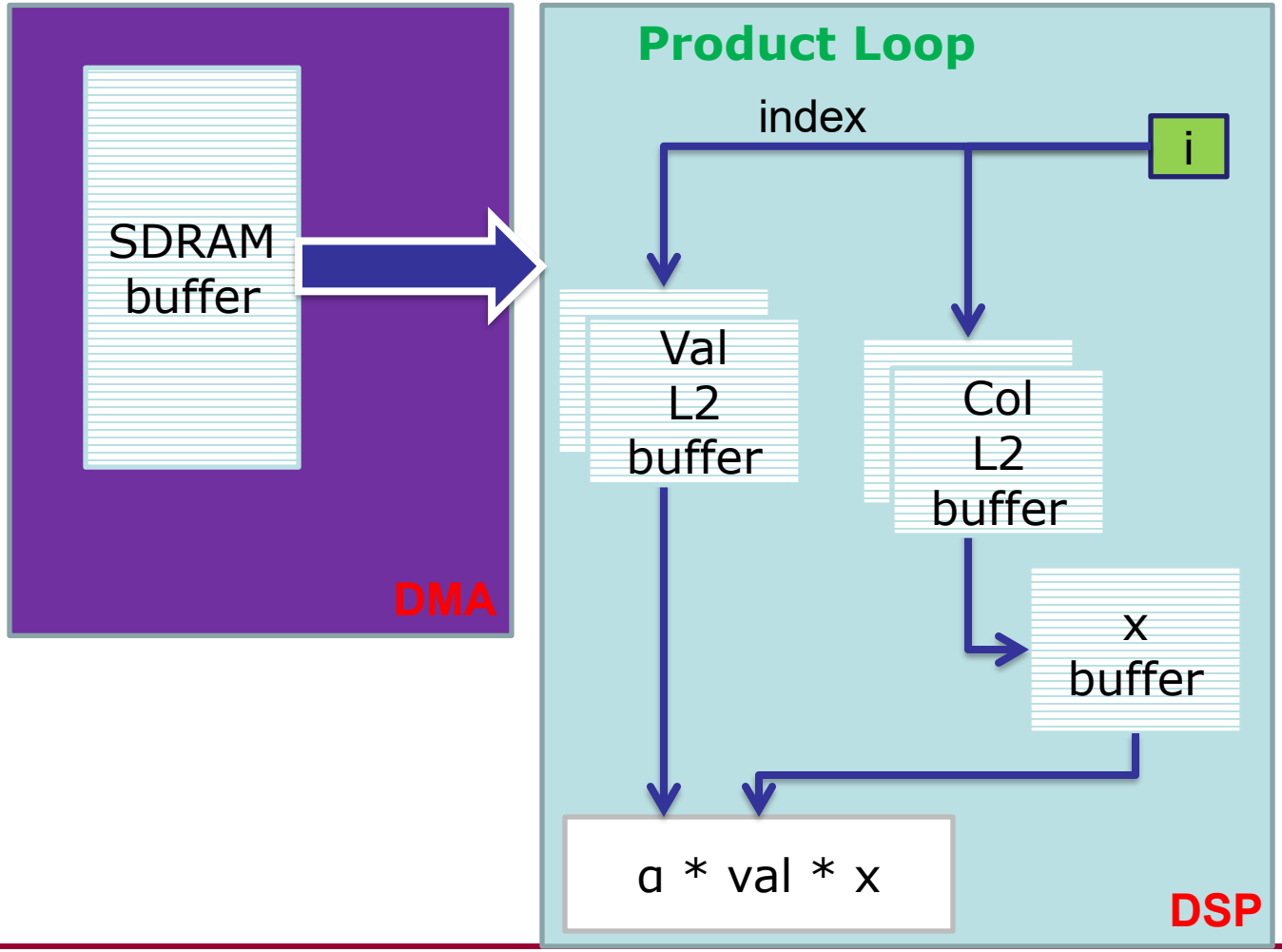


Naïve Implementation



0.55 Gflops/s
60.4% of cycles were uncovered memory latency

Double Buffer and DMA

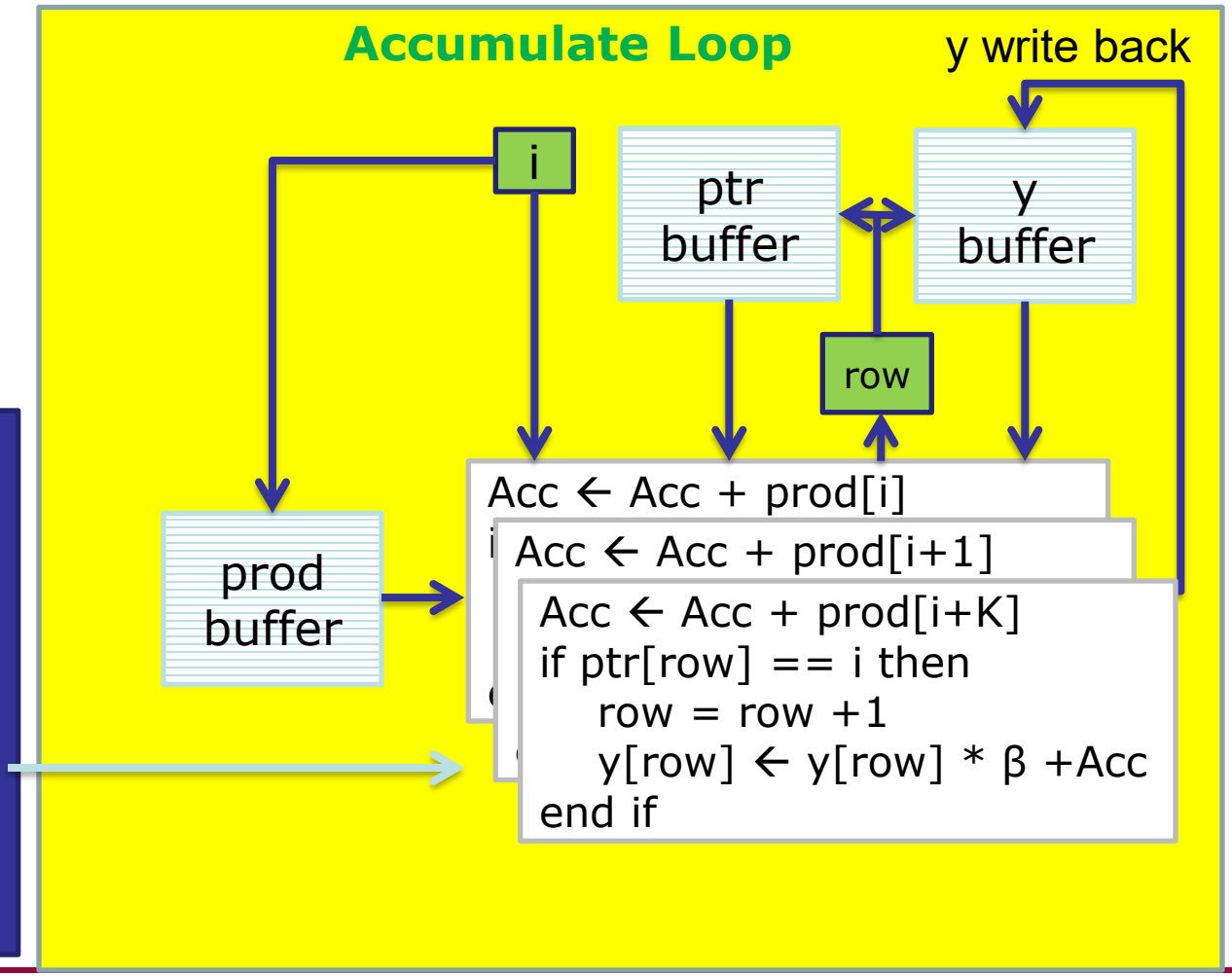


0.78 Gflops/s
28.8% of cycles were uncovered memory latency

Loop Unroll and Predicate Instruction

The accumulate loop is **manually unrolled** by 8

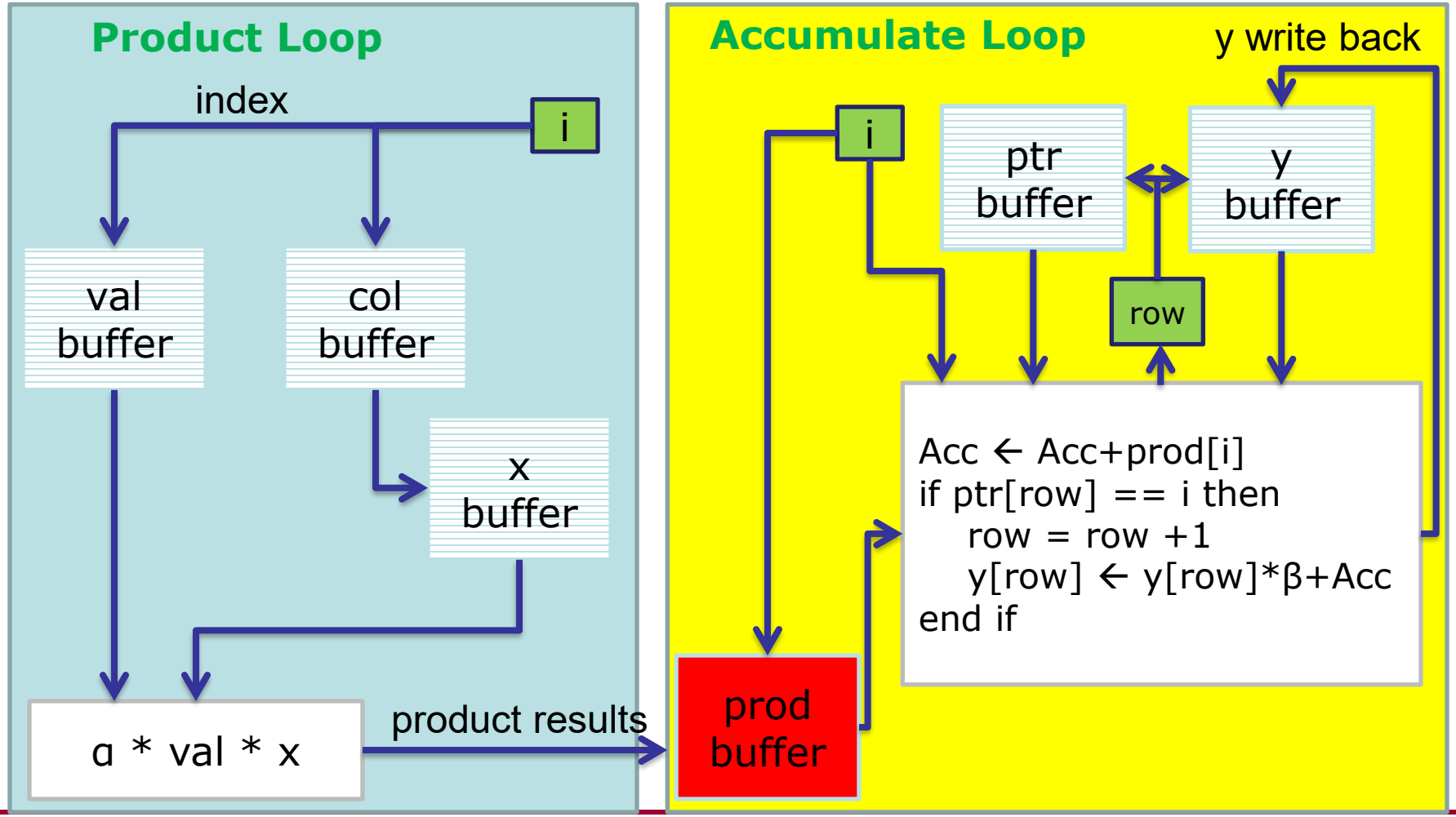
Predicate instructions are applied to replace the if-statements in **assembly**.



1.63 Gflops/s

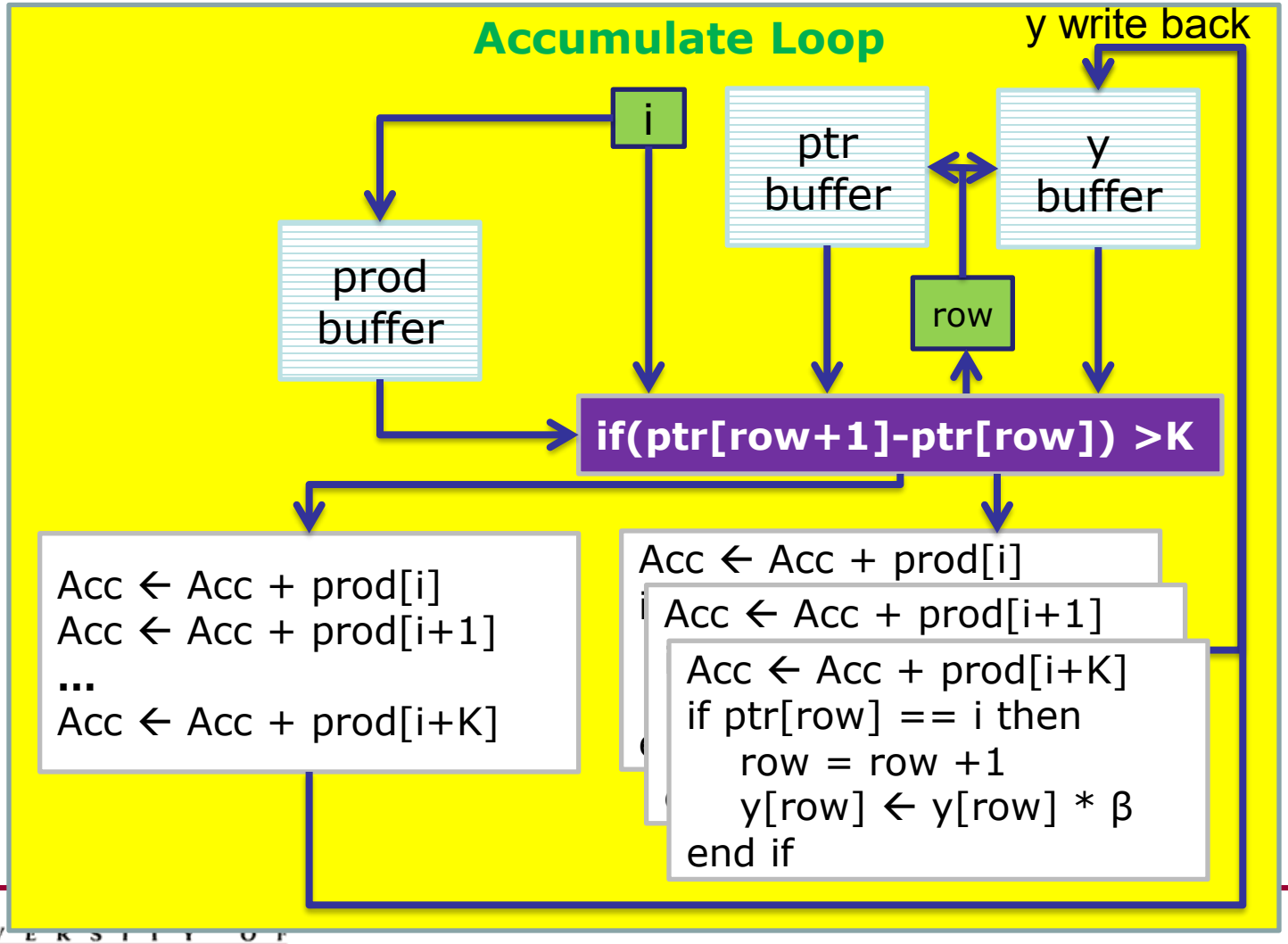
50.1% cycles were uncovered memory latency

Loop Fission



2.08 Gflops/s
36.6% cycles were uncovered memory latency

Adaptive Row Pointer



Test Environment

	i5 650 MKL	GTX680 CUSPARSE	GTX650Ti CUSPARSE	C66
Architecture	Clarkdale	Kepler	Kepler	Shannon
Process(nm)	32	28	28	45
Memory throughput (GB/s)	21	192.3	86.4	12.8
TDP (W)	73	195	110	10
Single precision performance (Gflops)	26	3090	1425	128

Power Analyzer

Power Socket Provide by WT500

110 V

PSU with EVM board

12 V

Matrix

- Tri-diagonal

$$\begin{bmatrix} 2 & 4 & 0 & 0 & 0 & 0 \\ 5 & 4 & 7 & 0 & 0 & 0 \\ 0 & 6 & 2 & 4 & 0 & 0 \\ 0 & 0 & 3 & 10 & 1 & 0 \\ 0 & 0 & 0 & 4 & 6 & 8 \\ 0 & 0 & 0 & 0 & 2 & 12 \end{bmatrix}$$

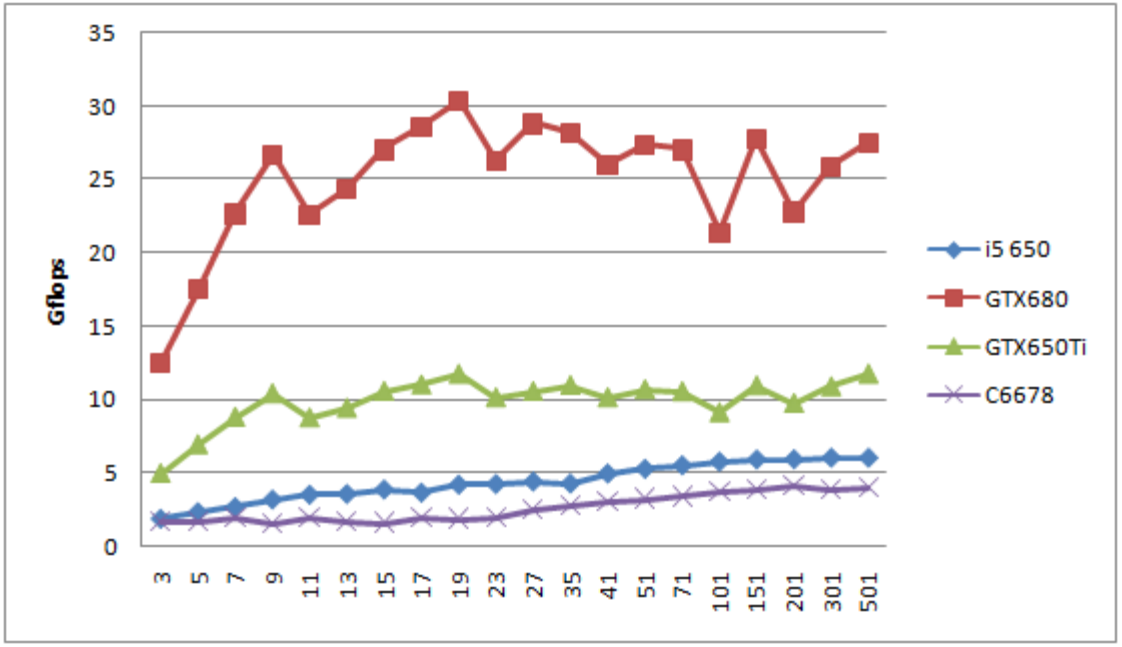
- N-diagonal
  3 - 501

- University of Florida sparse matrix collection
- Matrix Market

Matrix	Rows	Columns	Nonzeros	Nonzeros /Row
TSOPF_FS_b300_c3	84414	84414	13135930	155.6
pdb1HYS	36417	36417	4344765	119.3
m_t1	97578	97578	9753570	99.9
audikw_1	943695	943695	77651847	82.3
consph	83334	83334	6010480	72.1
cant	62451	62451	4007383	64.2
pwtk	217918	217918	11524432	52.9
shipsec1	140874	140874	3568176	25
ldoor	952203	952203	23737339	24.9
lhr71c	70304	70304	1528092	21.7
thermal1	82654	82654	574458	6.9
mac_econ_fwd500	206500	206500	1273389	6.1
ASIC_100ks	99190	99190	578890	5.8
scircuit	170998	170998	958936	5.6
shyy161	76480	76480	329762	4.3
mc2depi	525825	525825	2100225	4.0

SpMV Performance

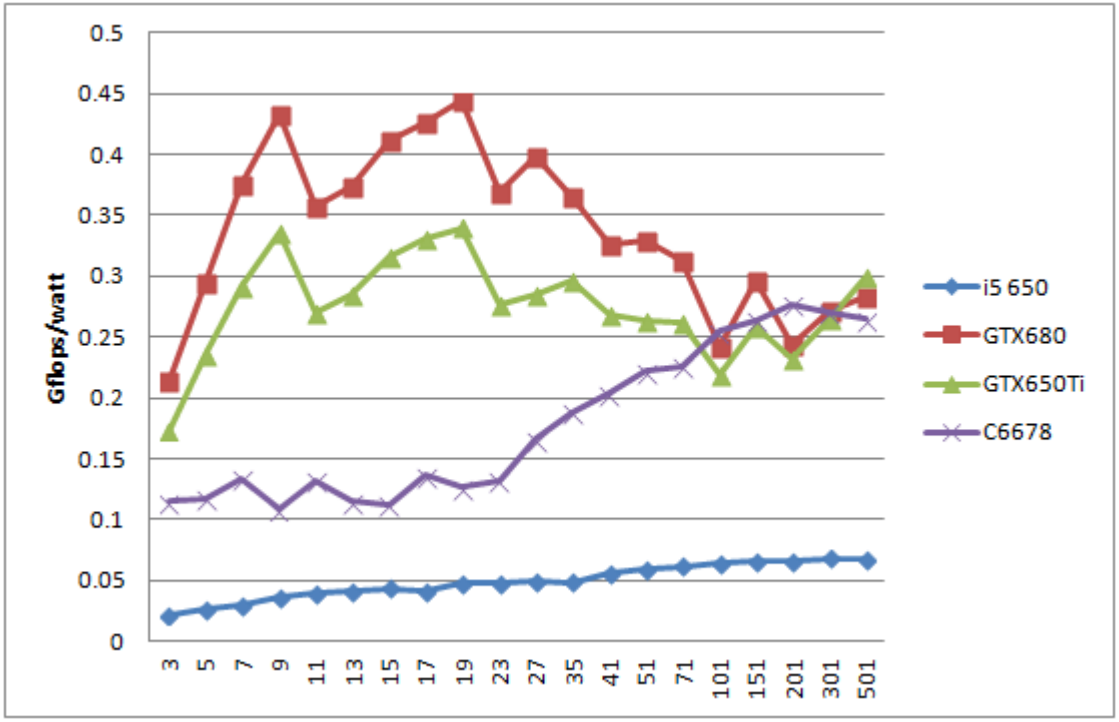
- N-diagonal Matrix



- Generally, the C66 achieves ~2/3 CPU performance

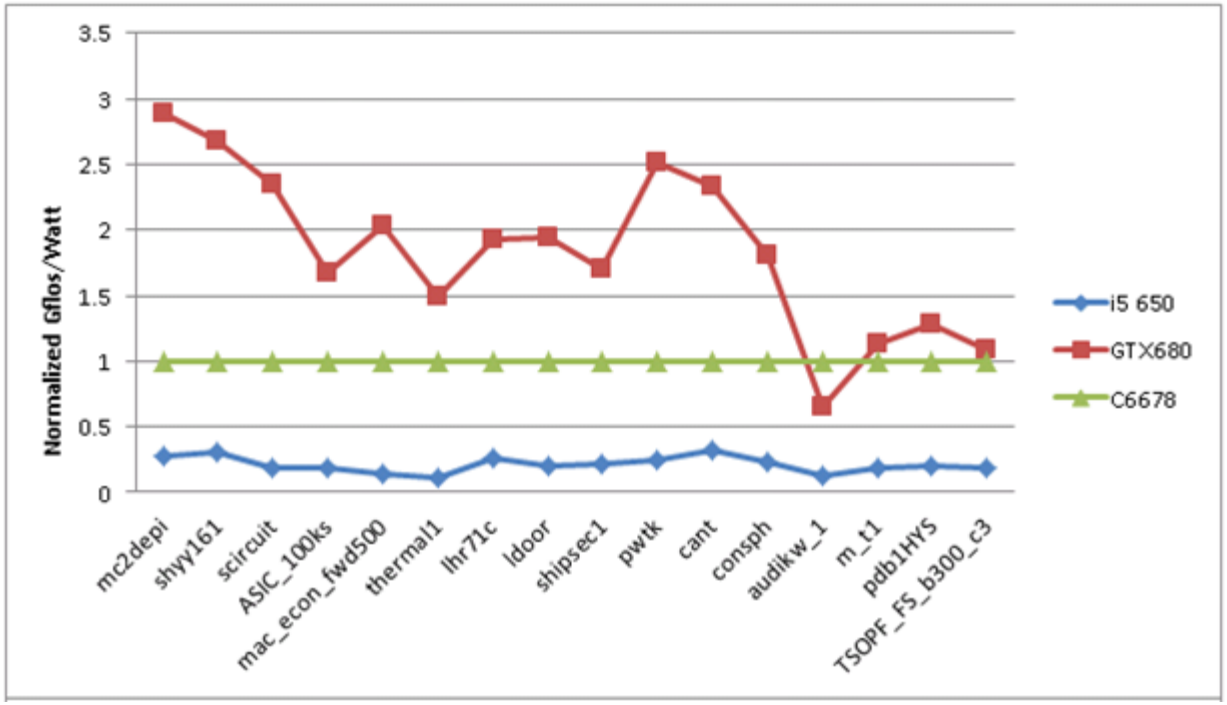
SpMV Gflops/Watt

- N-diagonal Matrix



- C66 is equivalent to GPUs when N > 51

Gflops/Watt for Nonsynthetic Matrices



- C66 power efficiency also scales with density for real-world matrices

Memory Efficiency

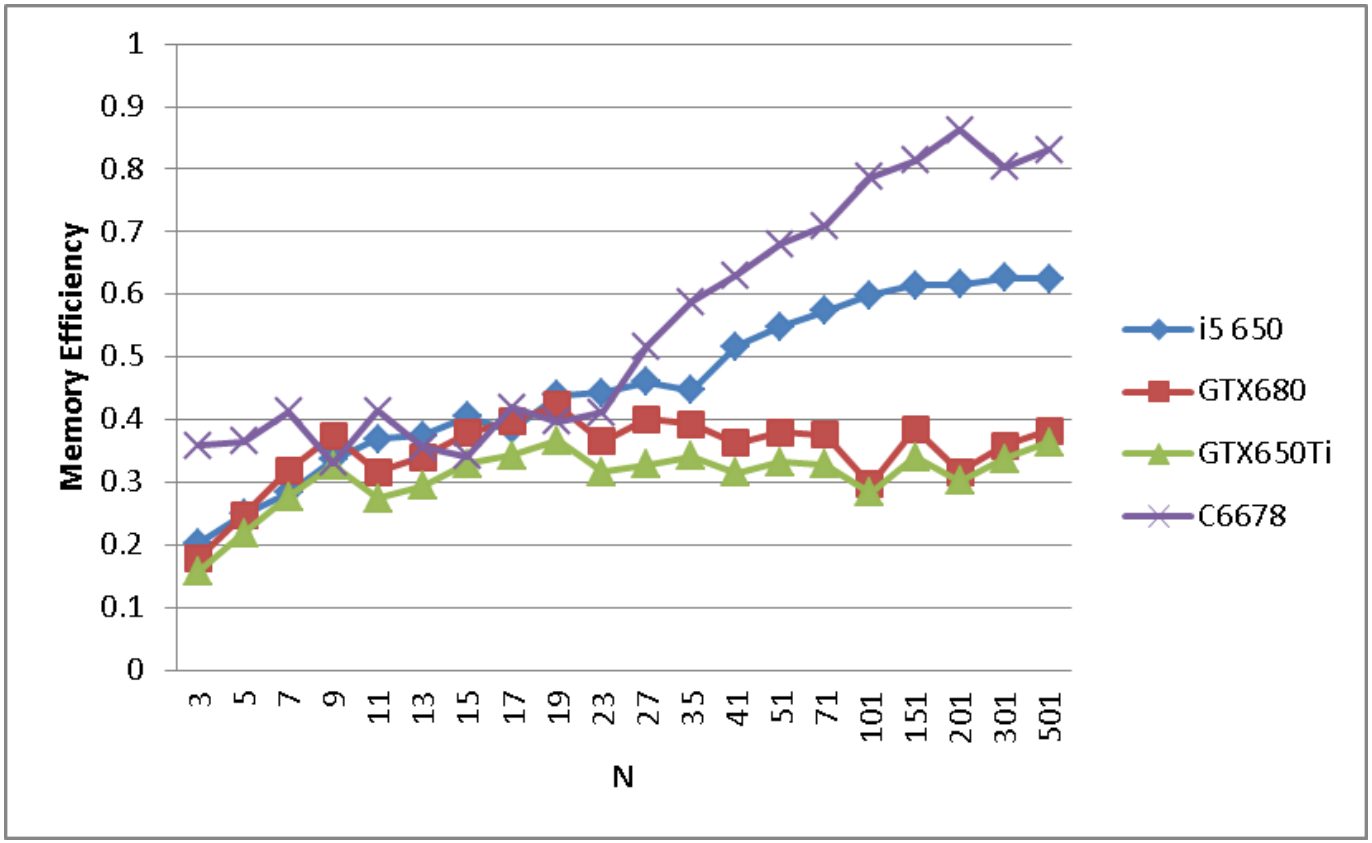
$$AI = \frac{9 * rows * n + 8 * rows + 2}{12(2 * rows * n + n + 2 * rows + 1)} ops/byte$$

	Intel i5 650	Nvidia GTX680	Nvidia GTX650Ti	Texas Instruments C6678
Max memory throughput	25.6 Gbytes/s	**192.3** Gbytes/s	86.4 Gbytes/s	12.8 Gbytes/s
Peak computational throughput	9.59 Gflops	**72.1** Gflops	32.4 Gflops	4.80 Gflops
Actual performance	5.89 Gflops	**27.8** Gflops	11.0 Gflops	3.9 Gflops
Memory efficiency	0.61	0.39	0.34	**0.81**

N = 151
rows = 208326

$AI * 12.8$

Memory Efficiency



Next Generation

- Keystone-II
  - 28 nm
  - Doubles caches
  - Increases memory bandwidth by 125%

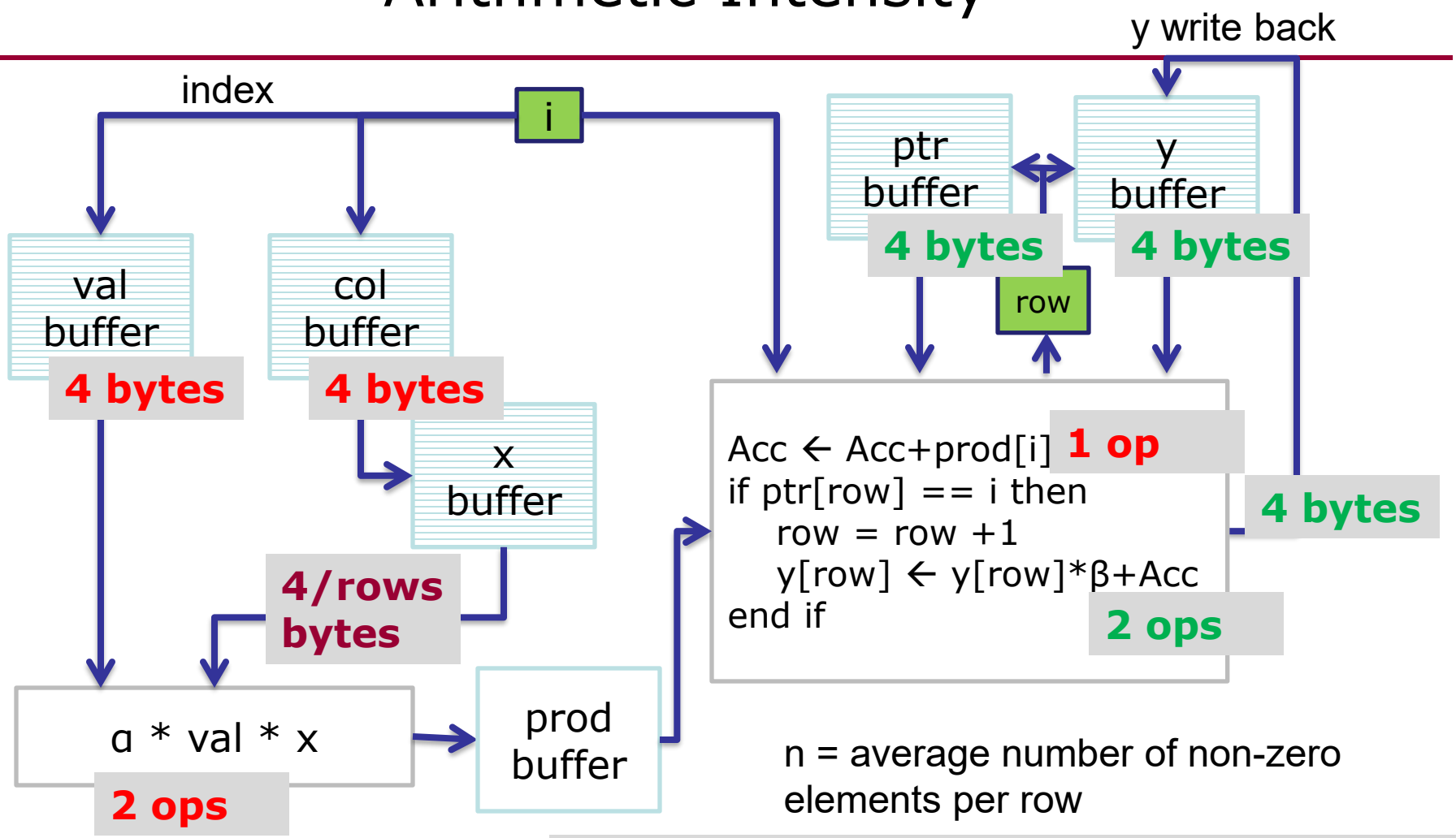
	C66	66AK2H12 "Keystone-II"
CPU	n/a	4 x ARM A15
DSP	8 Cores	8 Cores
DSP L2	512 KB	1024 KB
DDR3	64 bit	2 x 72 bit
Process	45 nm	28 nm
Power	10 W	?

Conclusions

- TI DSP is a promising coprocessor technology for HPC
- Advantages:
  1. Unique architectural features that facilitate automated parallelization (easier to program?)
  2. Inherently power efficient microarchitecture
     - Equivalent to modern GPUs and Phi despite older process technology
  3. Has advanced memory system for memory bound kernels
     - Simultaneous DMA and caching to match access pattern of indvidivual arrays
  4. Has advanced onchip interfaces for efficient scalability
     - Large-scale multi-DSP platforms already exist
- Looking forward:
  - Keystone II will:
    1. Improve efficiency and memory performance (cache + b/w)
    2. Has onboard host CPUs to facilitate runtimes for multi-DSP scaling

Q & A

Arithmetic Intensity



$$AI = \left(\frac{3ops}{8 + \frac{4}{rows}bytes} + \frac{1}{n} * \frac{2\text{ops}}{12\text{byts}}\right)/\left(1 + \frac{1}{n}\right)$$