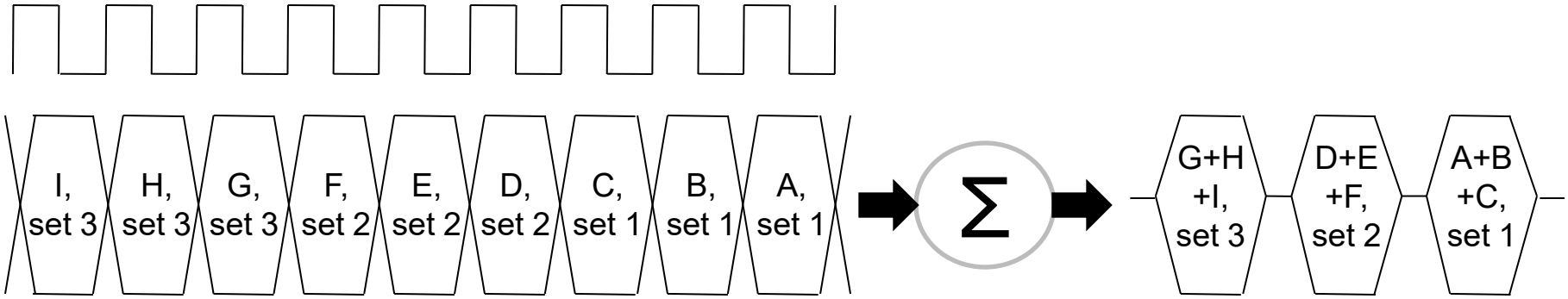
An Integrated Reduction Technique for a Double Precision Accumulator

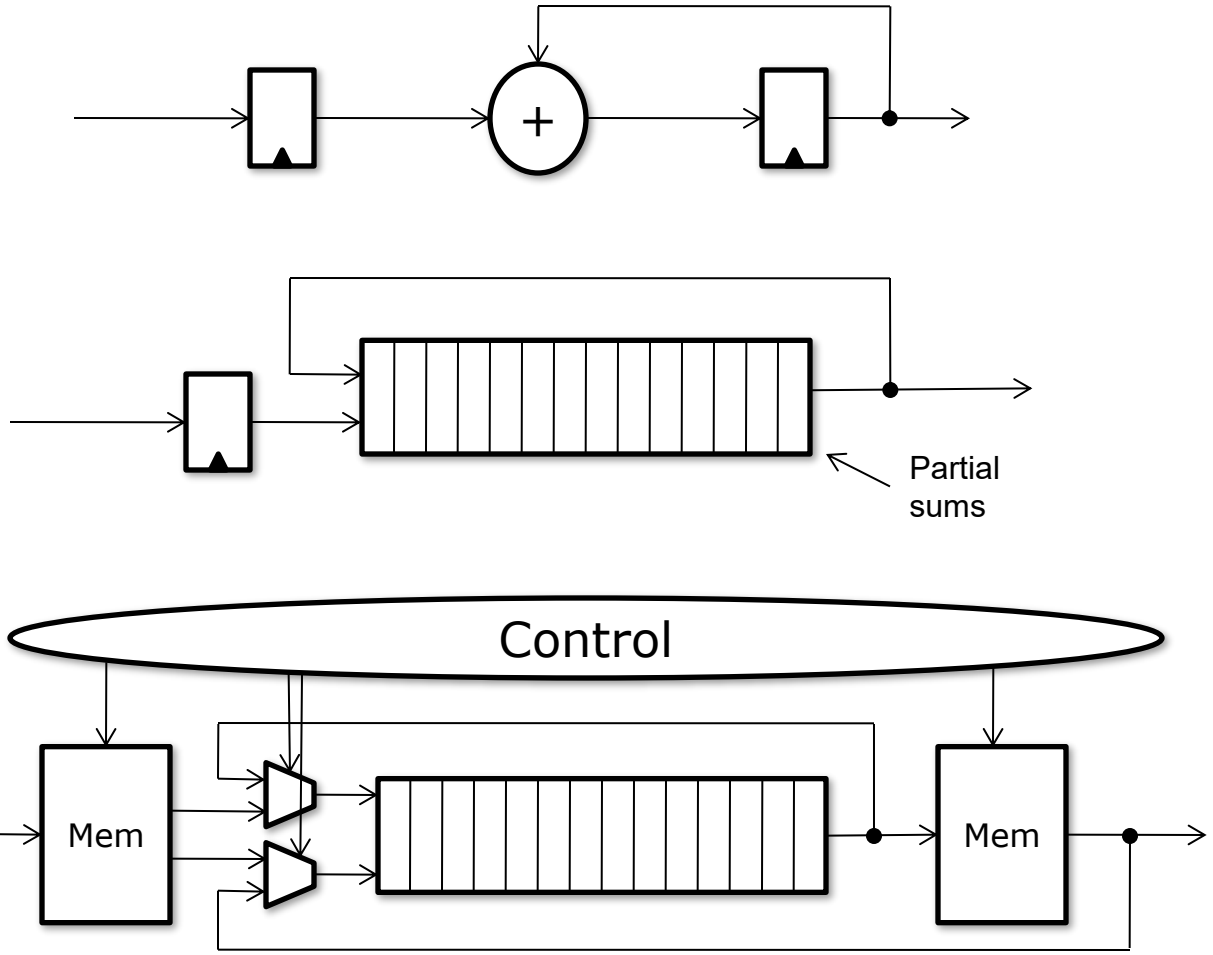
Krishna Nagar, Yan Zhang, Jason Bakos
Dept. of Computer Science and Engineering
University of South Carolina

Double Precision Accumulation

- Many kernels targeted for acceleration include $\sum_{i=1}^{n} f(i)$
- For large datasets, values delivered serially to an accumulator



The Reduction Problem



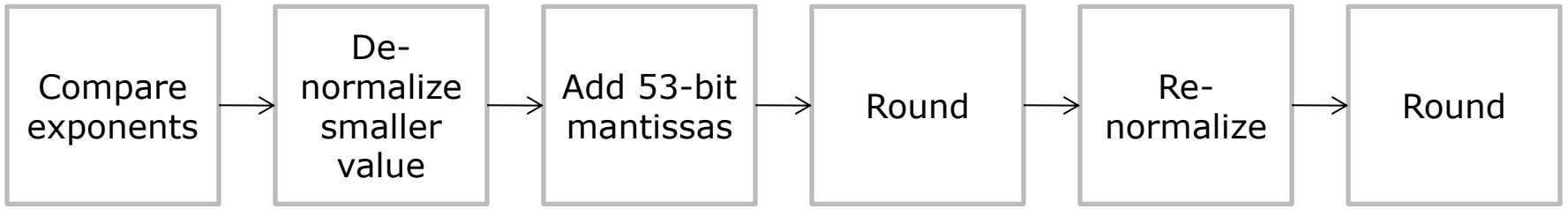
Reduction-Based Accumulator: Previous Work

Paper	# d.p. adder IP (~1000 slices/ea)	Reduc'n Logic	Reduc'n BRAM	# DSP48	D.p. adder speed	Accumulator speed	Out-of-order outputs
Prasanna DSA '07 (Virtex 2P)	2	2215 slices	3	n/a	170 MHz	142 MHz	Yes
Prasanna SSA '07 (Virtex 2P)	1	1804 slices	6	n/a	170 MHz	165 MHz	Yes
Gerards '08 (Virtex 4)	1	2722 slices	9	3 (from d.p. adder)	324 MHz	200 MHz	No
This work (Virtex 5)	0	< 1000 slices	0	3	355 MHz	300+ MHz	No

Approach

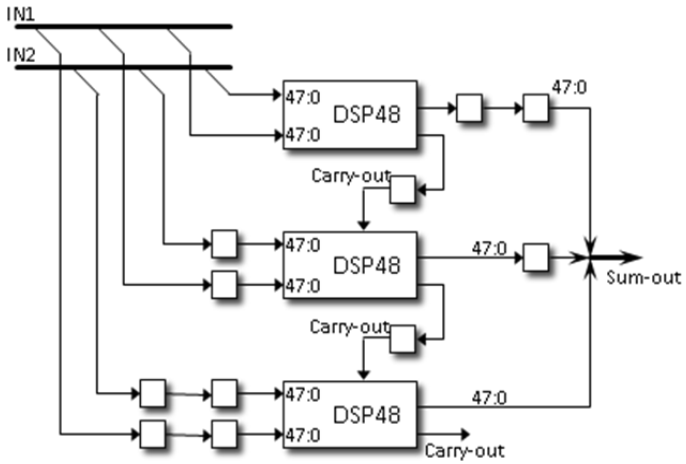
- Reduction complexity scales with the latency of the core operation
  - Reduce latency of double precision add?

- IEEE 754 adder pipeline (assume 4-bit significand):



Adder Pipeline

- Mantissa addition
  - Cascaded, pipelined DSP48 adders
  - Scales well, operates fast
- De-normalize
  - Exponent comparison and a variable shift of one significand
  - Xilinx IP uses a DSP48 for the 11-bit comparison (waste)



Base Conversion

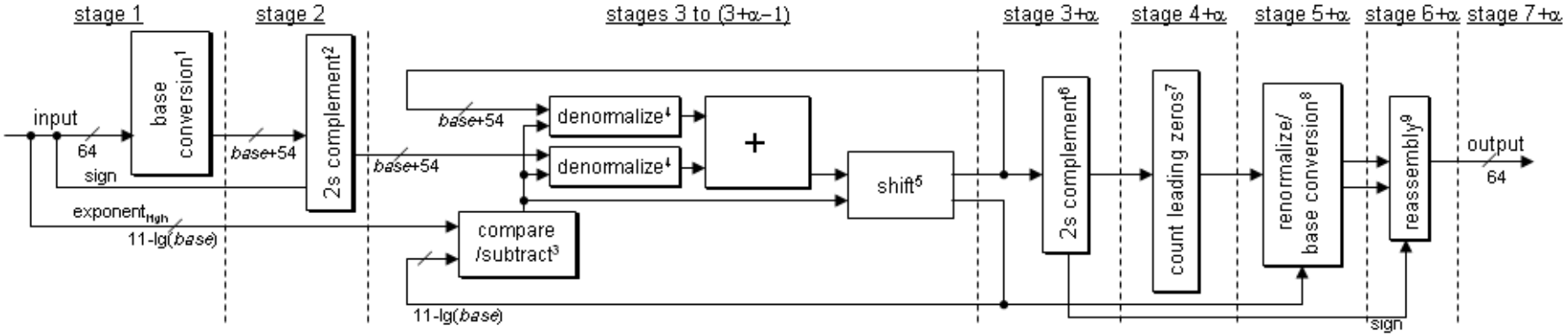
- Previous work in s.p. MAC designs base conversion
  - Idea:
    - Shift both inputs to the left by amout specified in low-order bits of exponents
    - Reduces size of exponent, requires wider adder

- Example:
  - Base-8 conversion:
    - 1.01011101, exp=10110 (1.36328125×2^{22} => ~5.7 million)
    - Shift to the left by 6 bits…
    - 1010111.01, exp=10 ($87.25 \times 2^{8*2}$ = > ~5.7 million)

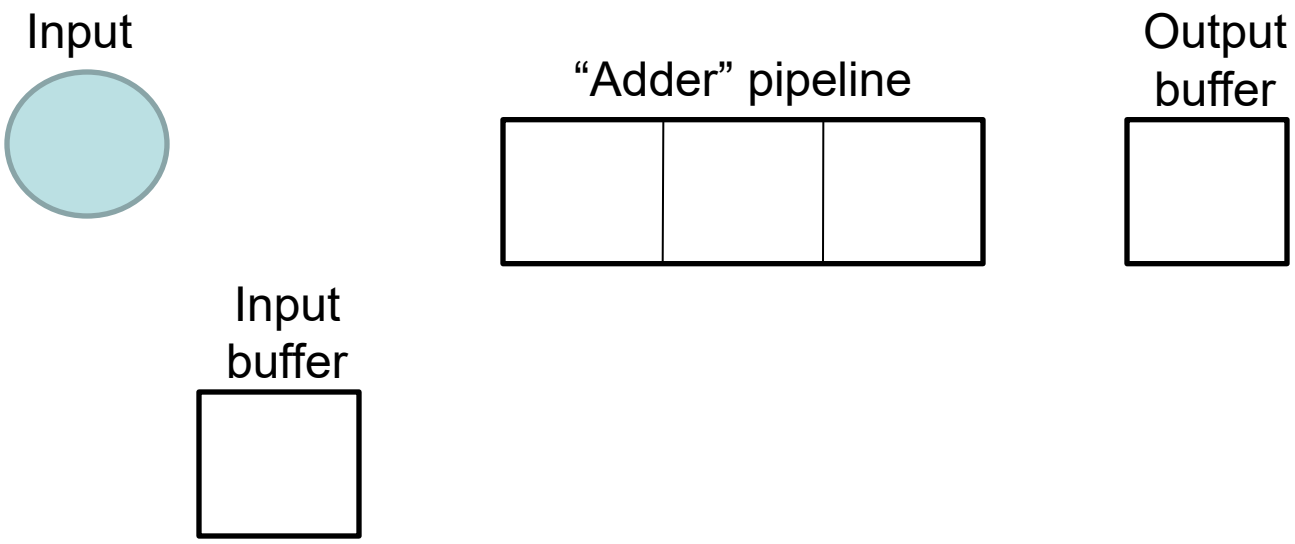
Exponent Compare vs. Adder Width

Base	Exponent Width	Denormalize speed	Adder Width	#DSP48s
16	7	119 MHz	54	2
32	6	246 MHz	86	2
64	5	368 MHz	118	3
128	4	372 MHz	182	4
256	3	494 MHz	310	7

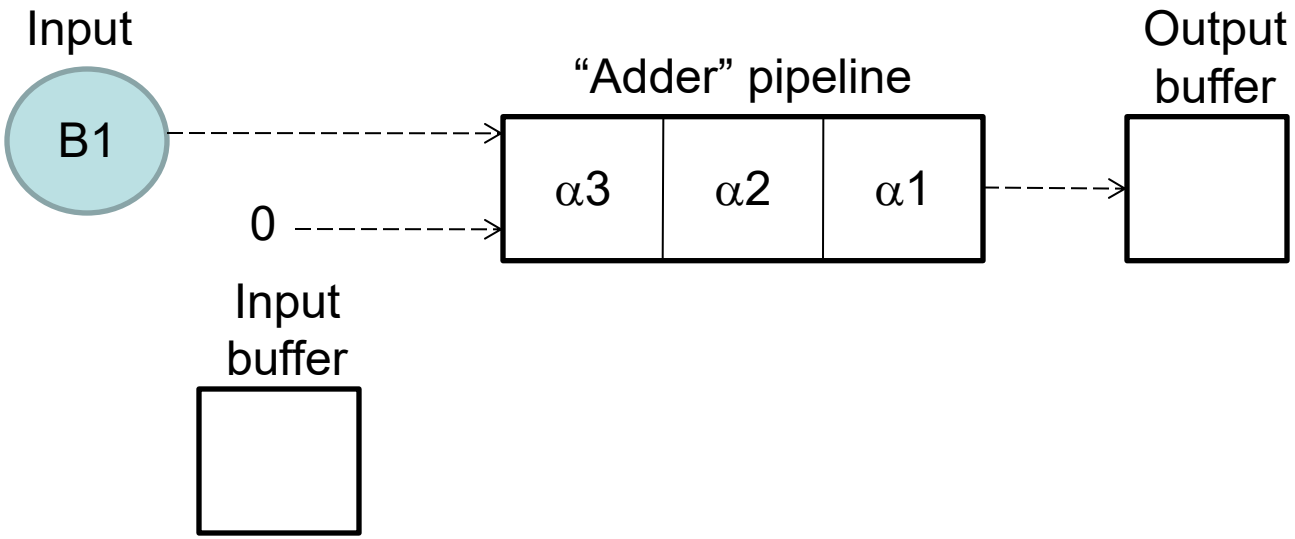
Accumulator Design



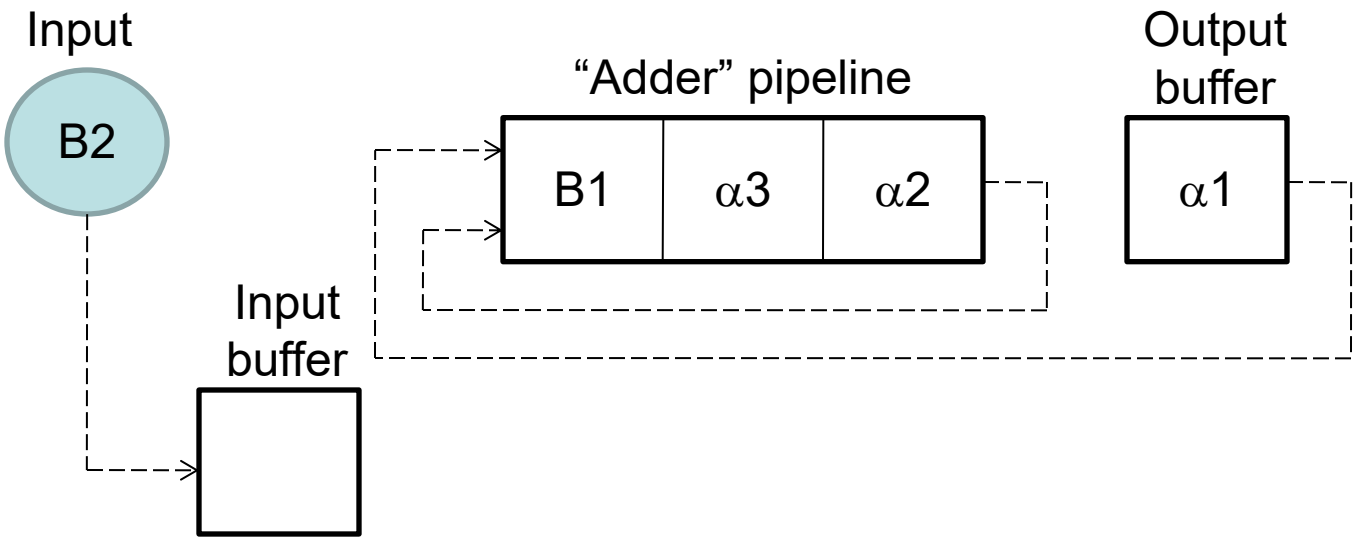
Three-Stage Reduction Architecture



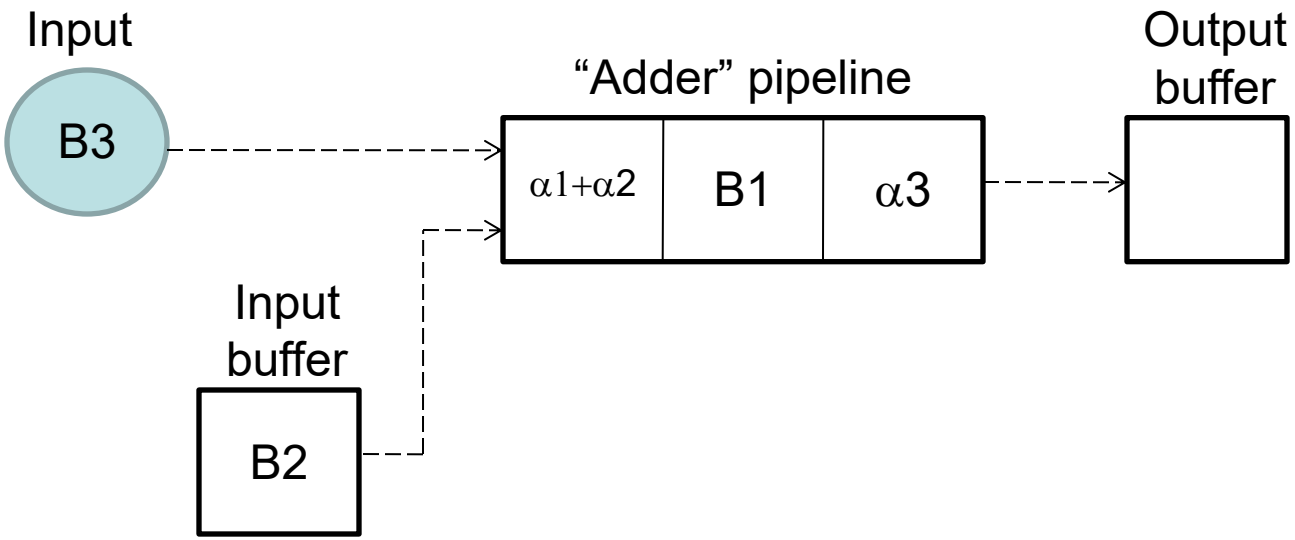
Three-Stage Reduction Architecture



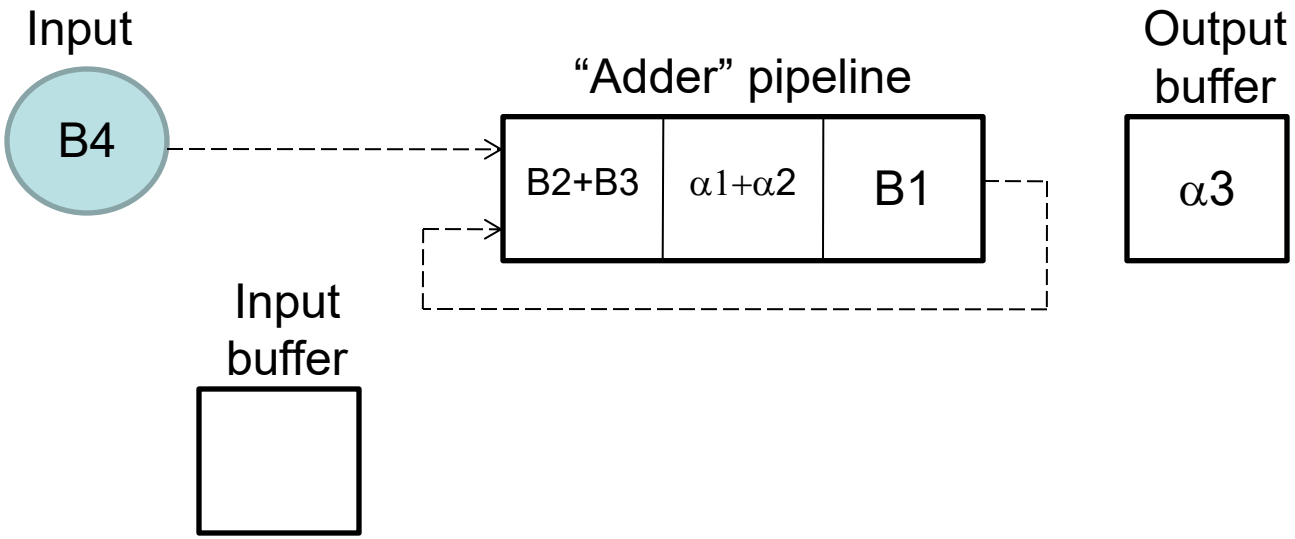
Three-Stage Reduction Architecture



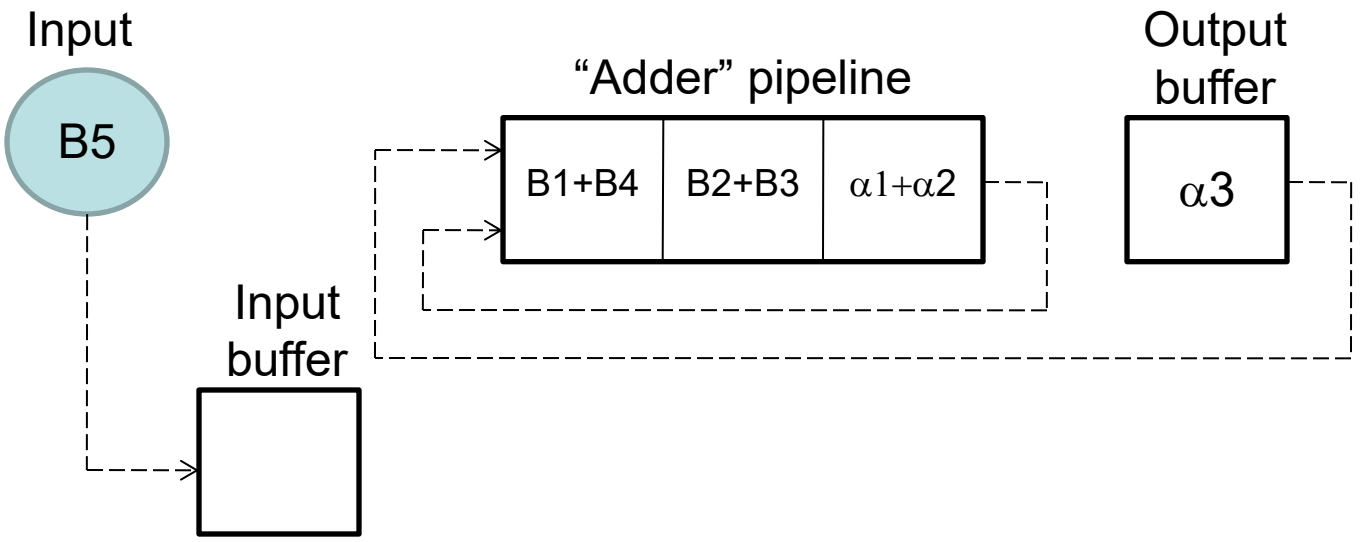
Three-Stage Reduction Architecture



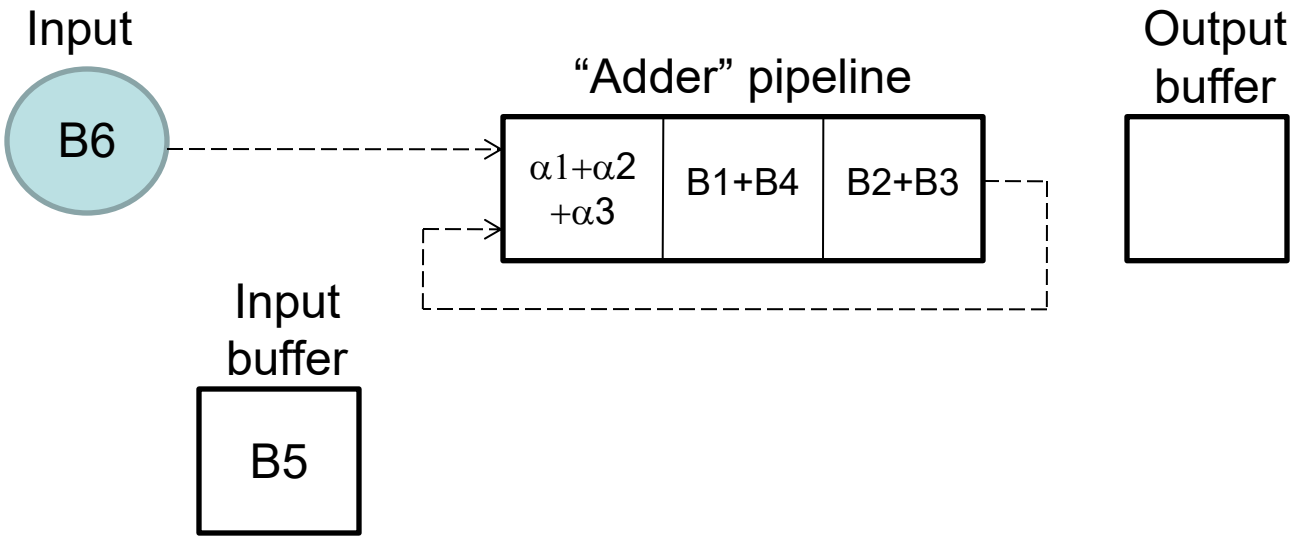
Three-Stage Reduction Architecture



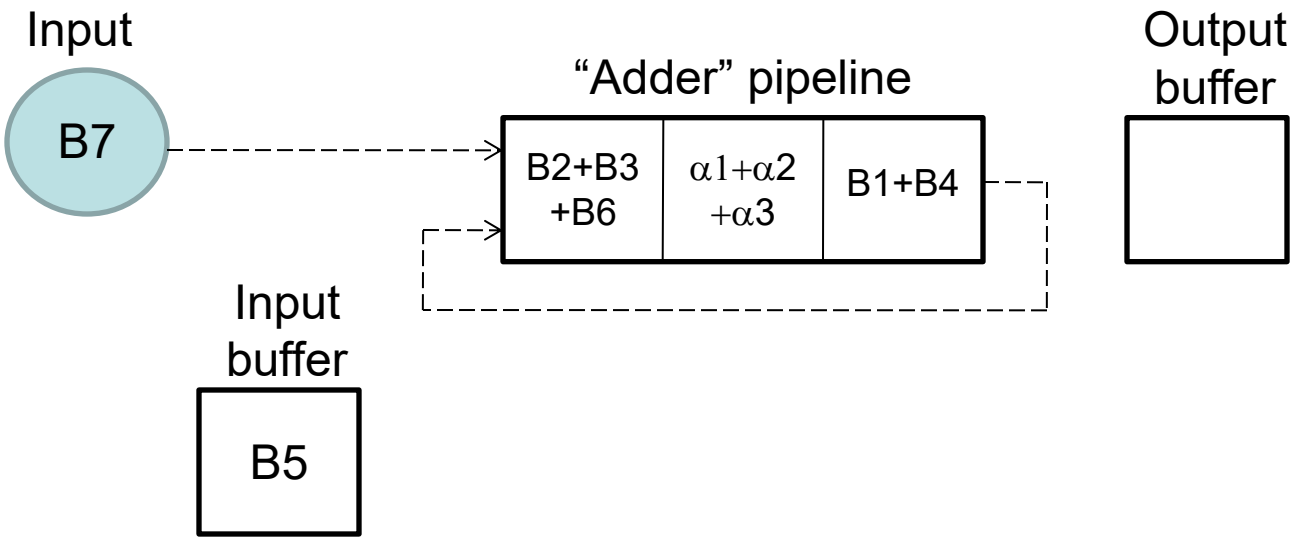
Three-Stage Reduction Architecture



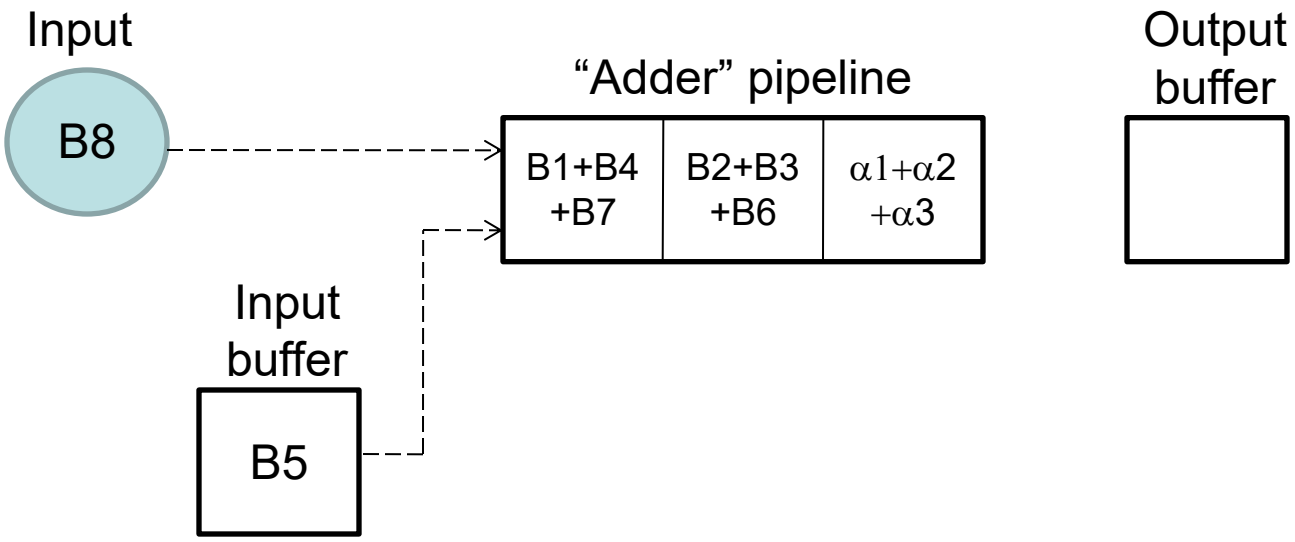
Three-Stage Reduction Architecture



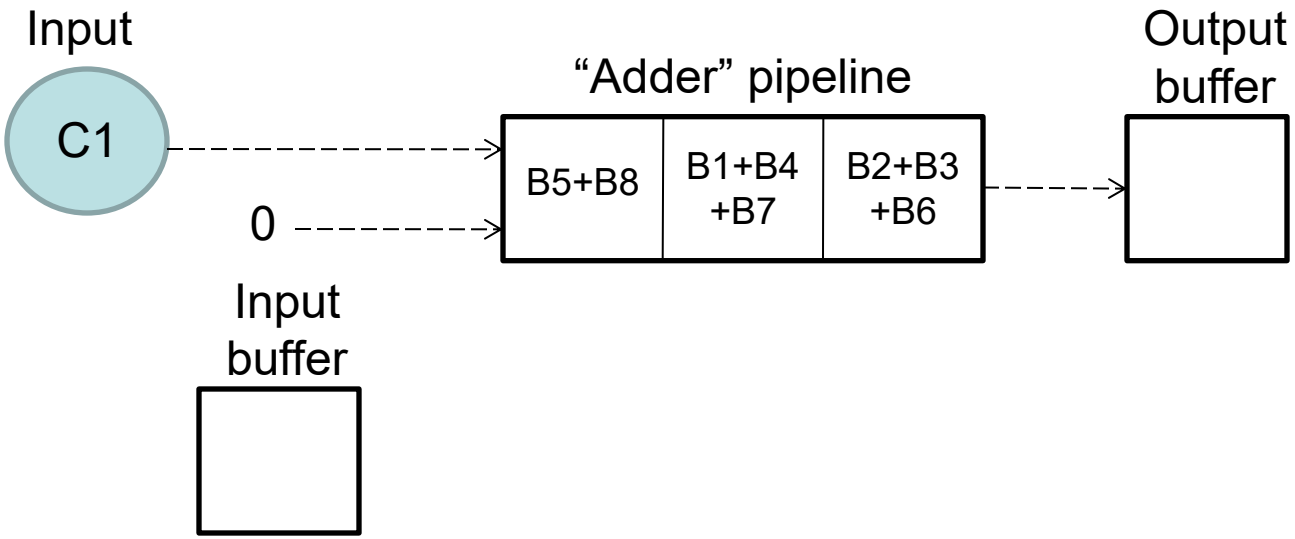
Three-Stage Reduction Architecture



Three-Stage Reduction Architecture

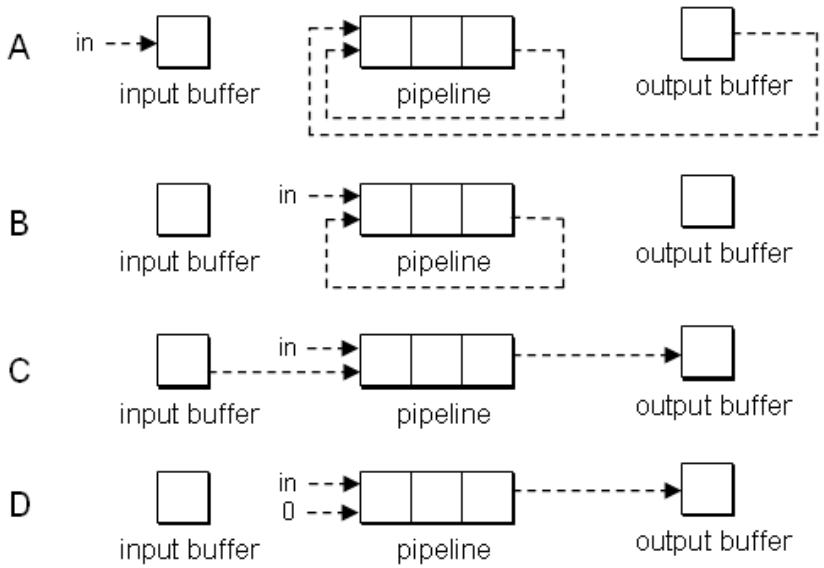


Three-Stage Reduction Architecture



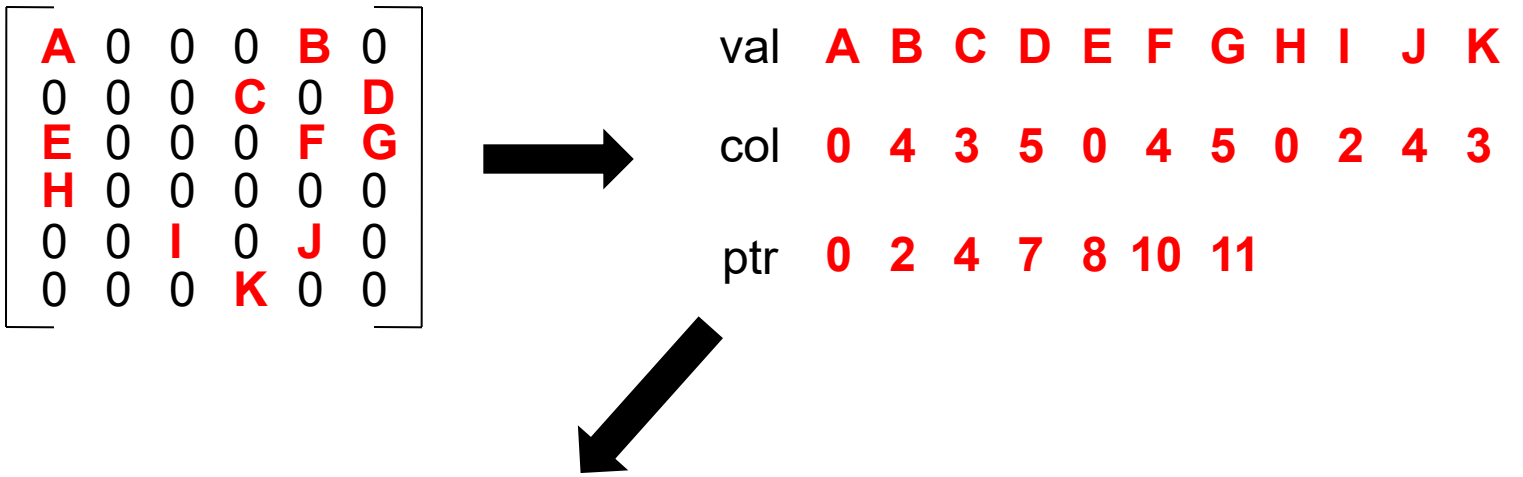
Minimum Set Size

- Four "configurations":



- Deterministic control sequence, triggered by set change:
  - D, A, C, B, A, B, B, C, B/D
- Minimum set size is 8

Use Case: Sparse Matrix-Vector Multiply

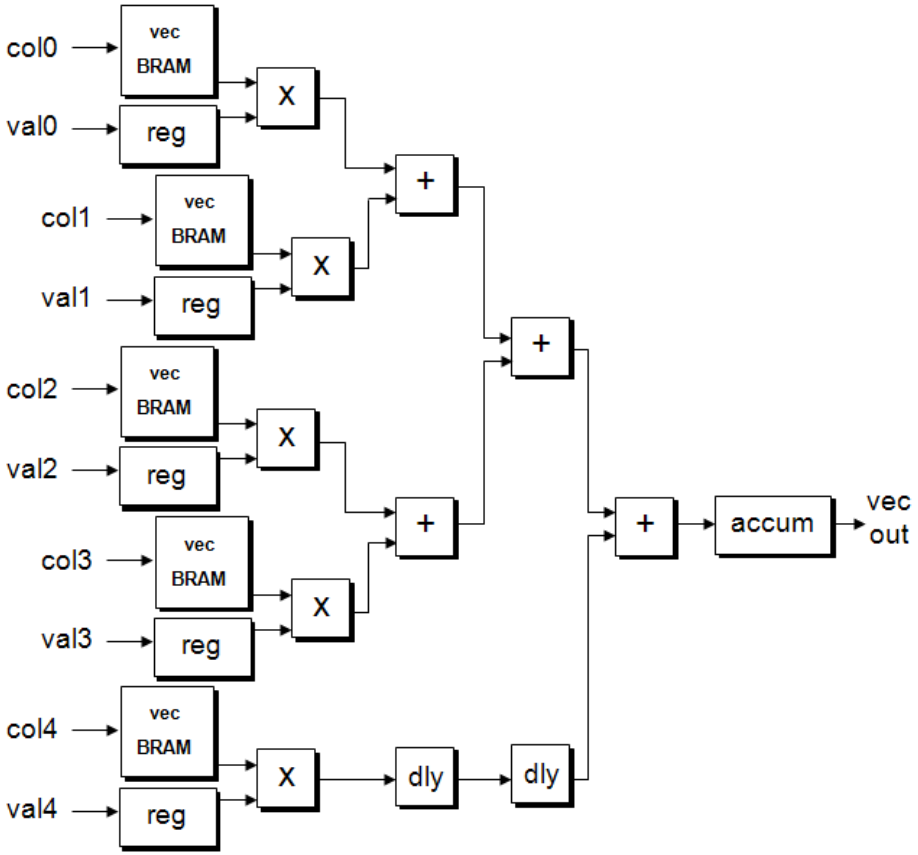


- Group vol/col
- Zero-terminate

(A,0) (B,4) (0,0) (C,3) (D,4) (0,0)…

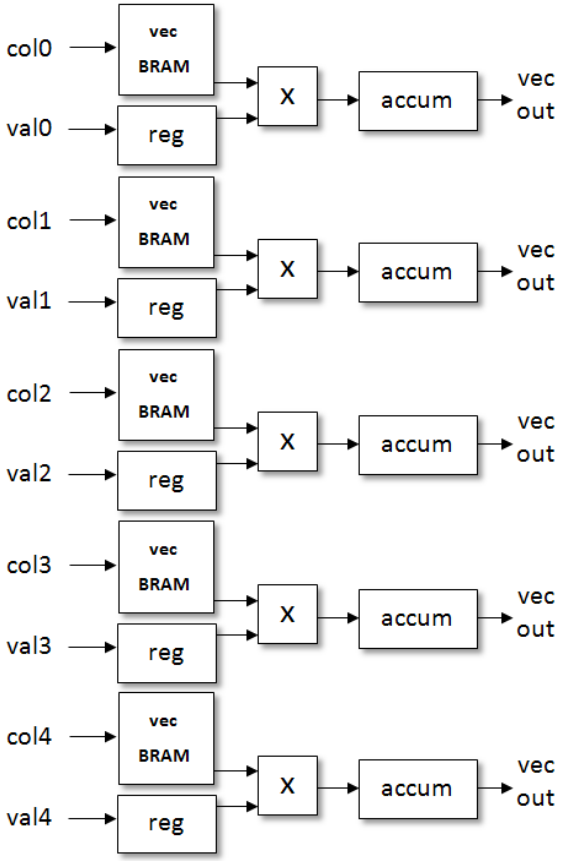
SpMV Architecture

- Enough memory bandwidth to read:
  - 5 val/col pairs (80 x 5 bits) per cycle
  - ~15-20 GB/s
- Requires minimum number of entries per row:
  - 5 x 8 = 40
  - Many sparse matrices don't have this many values per row
  - Zero padding will degrade performance for many matrices



New SpMV Architecture

- Delete tree, replicate accumulator, schedule matrix data:



400 bits

$val_{0,0}$	$col_{0,0}$	$val_{1,0}$	$col_{1,0}$	$val_{2,0}$	$col_{2,0}$	$val_{3,0}$	$col_{3,0}$	$val_{4,0}$	$col_{4,0}$
$val_{0,1}$	$col_{0,1}$	$val_{1,1}$	$col_{1,1}$	$val_{2,1}$	$col_{2,1}$	$val_{3,1}$	$col_{3,1}$	$val_{4,1}$	$col_{4,1}$
$val_{0,2}$	$col_{0,2}$	$val_{1,2}$	$col_{1,2}$	$val_{2,2}$	$col_{2,2}$	$val_{3,2}$	$col_{3,2}$	$val_{4,2}$	$col_{4,2}$
$val_{0,3}$	$col_{0,3}$	$val_{1,3}$	$col_{1,3}$	$val_{2,3}$	$col_{2,3}$	$val_{3,3}$	$col_{3,3}$	$val_{4,3}$	$col_{4,3}$
$val_{0,4}$	$col_{0,4}$	$val_{1,4}$	$col_{1,4}$	$val_{2,4}$	$col_{2,4}$	$val_{3,4}$	$col_{3,4}$	$val_{4,4}$	$col_{4,4}$
$val_{0,5}$	$col_{0,5}$	$val_{1,5}$	$col_{1,5}$	$val_{2,5}$	$col_{2,5}$	$val_{3,5}$	$col_{3,5}$	$val_{4,5}$	$col_{4,5}$
$val_{0,6}$	$col_{0,6}$	0.0	0.0	$val_{2,6}$	$col_{2,6}$	$val_{3,6}$	$col_{3,6}$	$val_{4,6}$	$col_{4,6}$
$val_{0,7}$	$col_{0,7}$	0.0	5	$val_{2,7}$	$col_{2,7}$	$val_{3,7}$	$col_{3,7}$	$val_{4,7}$	$col_{4,7}$
$val_{0,8}$	$col_{0,8}$	$val_{5,0}$	$col_{5,0}$	$val_{2,8}$	$col_{2,8}$	$val_{3,8}$	$col_{3,8}$	$val_{4,8}$	$col_{4,8}$

Performance Results

Matrix	Order/ dimensions	Ave. non-zeroes per row	Number of non-zero entries	Entries after zero-padding	Entries after zero termination	Number of 400-bit packets	Data Utilization	MFLOPS at 370MHz
TSOPF_RS_b162_c3	15374	40	610299	633231	648625	129725	0.941	3482
E40r1000	17281	32	553562	553562	570855	114171	0.970	3589
Simon/olafu	16146	32	1015156	1015156	1031330	206266	0.984	3641
Garon/garon2	13535	29	373235	373243	386800	77360	0.965	3571
Mallya/lhr11c	10964	21	233741	256014	267020	53404	0.875	3236
Hollinger/ mark3jac020sc	9129	6	52883	72608	81835	16367	0.646	2390
Bai/dw8192	8192	5	41746	57360	65575	13115	0.637	2357
YCheng/psse1	14318x11028	4	57376	100960	115295	23059	0.498	1843
GHS_indef/ ncvxqp1	12111	3	73963	99412	111535	22307	0.663	2453

Conclusions

- Developed serially-delivered accumulator using base-conversion technique
- Limited to shallow pipelines
  - Deeper pipelines require large minimum set size $\alpha\lceil \lg \alpha + 1 \rceil - 1$
    - 4 -> 11, 5 -> 19, 6 -> 23
- Goal: new reduction circuit to support deeper pipelines with no minimum set size
- Acknowledgements:
  - NSF awards CCF-0844951, CCF-0915608