Accuracy, Cost, and Performance Trade-offs for Floating Point Accumulation

Krishna K. Nagar and Jason D. Bakos

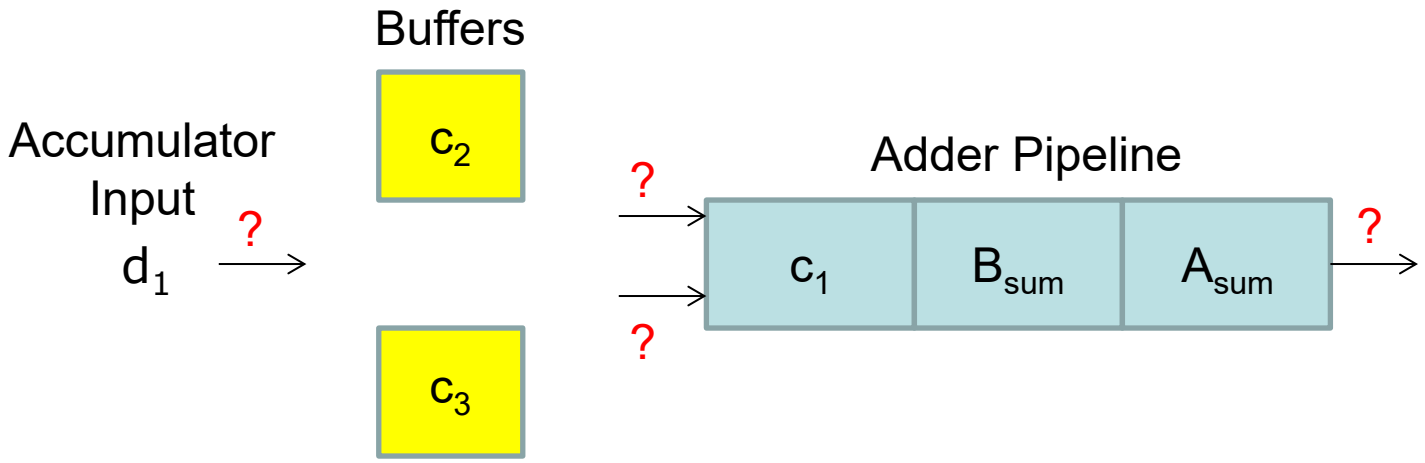
Univ. of South Carolina

UNIVERSITY OF SOUTH CAROLINA

Floating Point Accumulation

- Problem:
  - For floating point accumulation, high throughput and high accuracy are competing design goals
- Motivation:
  - Fully pipelined, streaming d.p. f.p. accumulator
  - Accepts one new value and set ID on every clock cycle
  - Exploits parallelism both within and across accumulation sets
  - Based on dynamically scheduling inputs to a single floating point adder
  - However, accuracy was inconsistent and data dependent

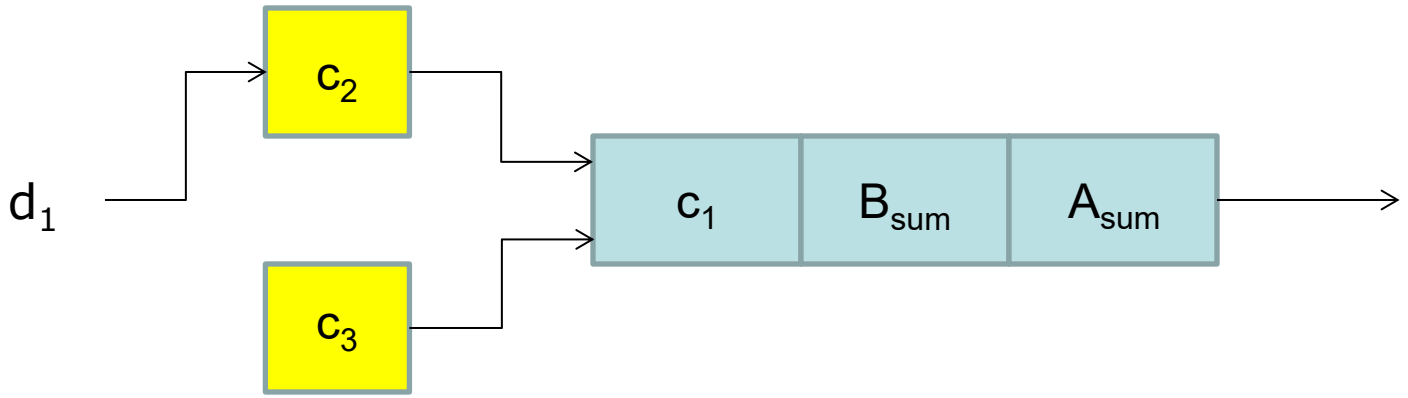
High Throughput Accumulation



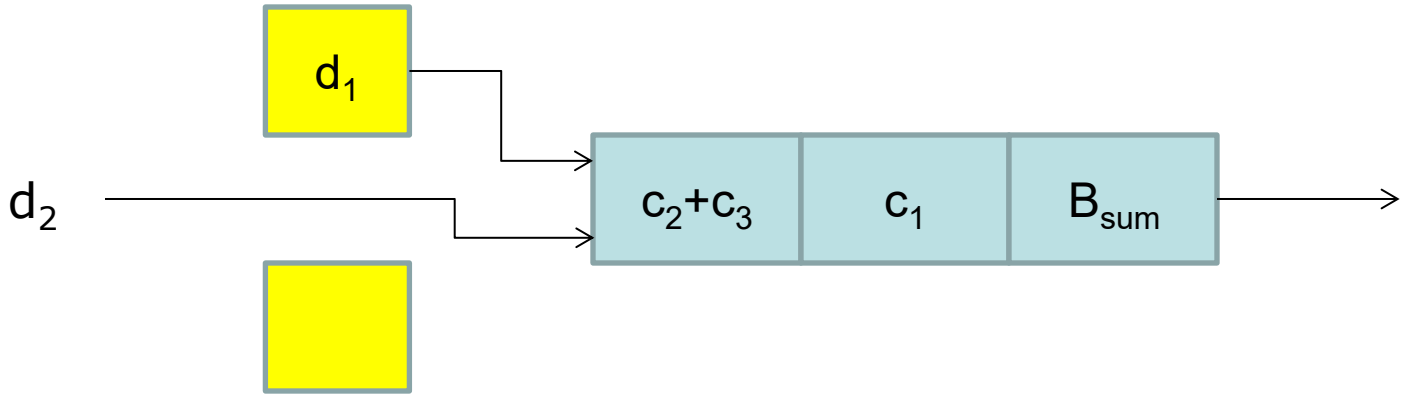
- Dynamically schedule:
  - Adder inputs
  - Next input value
  - Output of adder
- …based on datapath priorities and the set IDs

Accumulator Design

Rule 2

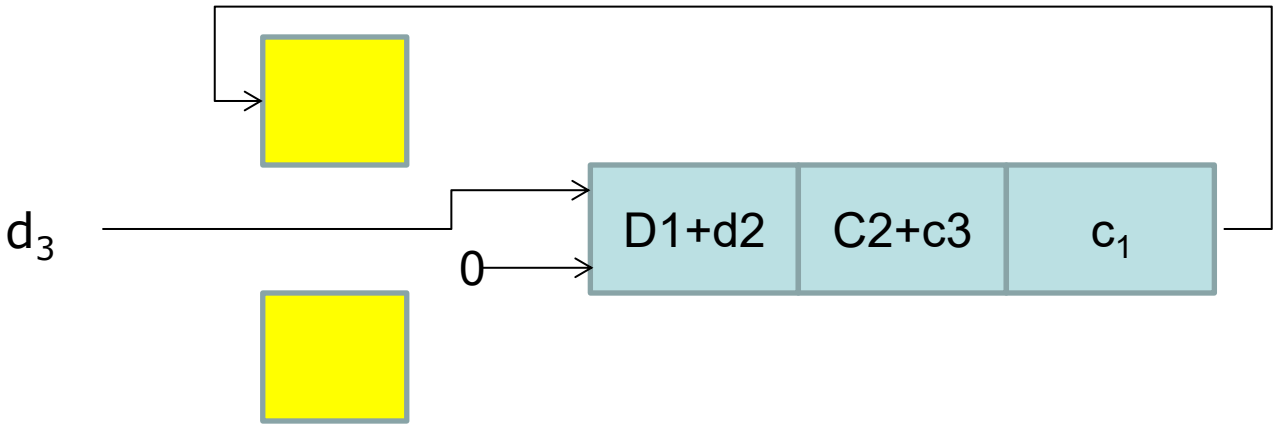


Rule 4

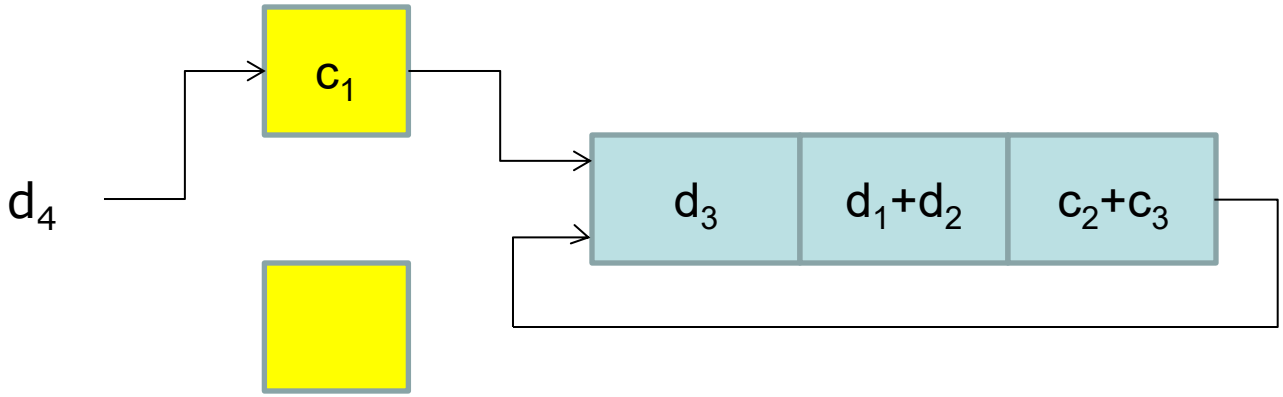


Accumulator Design

Rule 5



Rule 1



Inaccuracies in Floating Point Addition

- Floating point addition has inherent rounding error
  - Shift, round, cancellation
- F.p. addition is not associative
  - For the accumulator, the total error is data dependent
- We've seen as many as 50% of mantissa bits being in error after accumulating large synthetic data sets
- In general, to improve accuracy:
  - multiple passes over dataset (sorting),
  - multiple operations per input (additional dependencies), or
  - increased precision (lower clock rate, higher latency)
- …each reduces throughput

Compensated Summation

- Compensated summation
  - Preserve the rounding error from each addition
  - Incorporate the errors into the final result

Error Free Transformation

$$a + b = x + e, \qquad x = fl(a \oplus b)$$

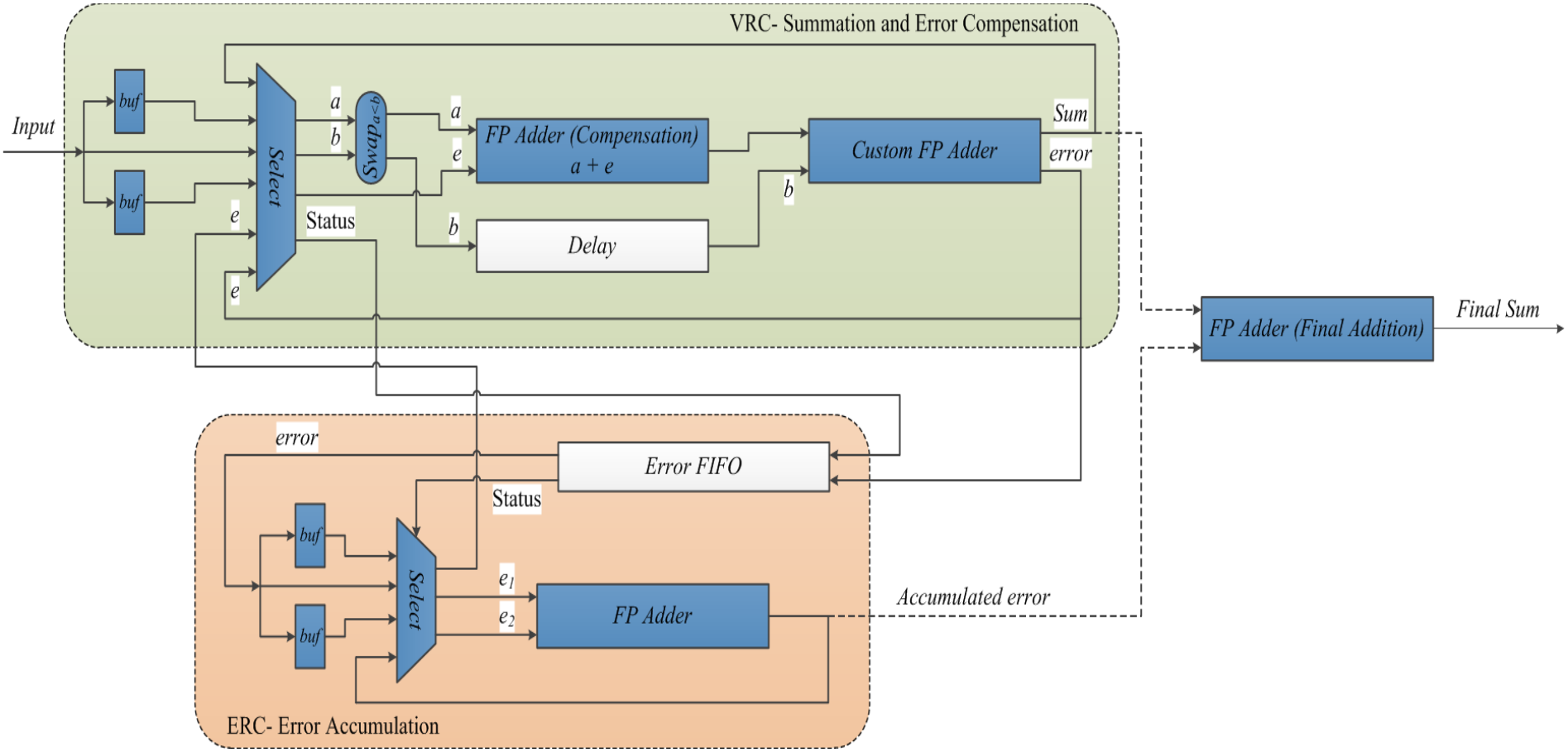
- Three strategies:
  1. Incorporate rounding errors from previous addition into subsequent addition
  2. Accumulate the error separately and incorporate total error into the final result
  3. Use extended precision adder (80- and 128-bit)

Error Extraction

- Relies on custom floating point adder that produces the roundoff error as a second output

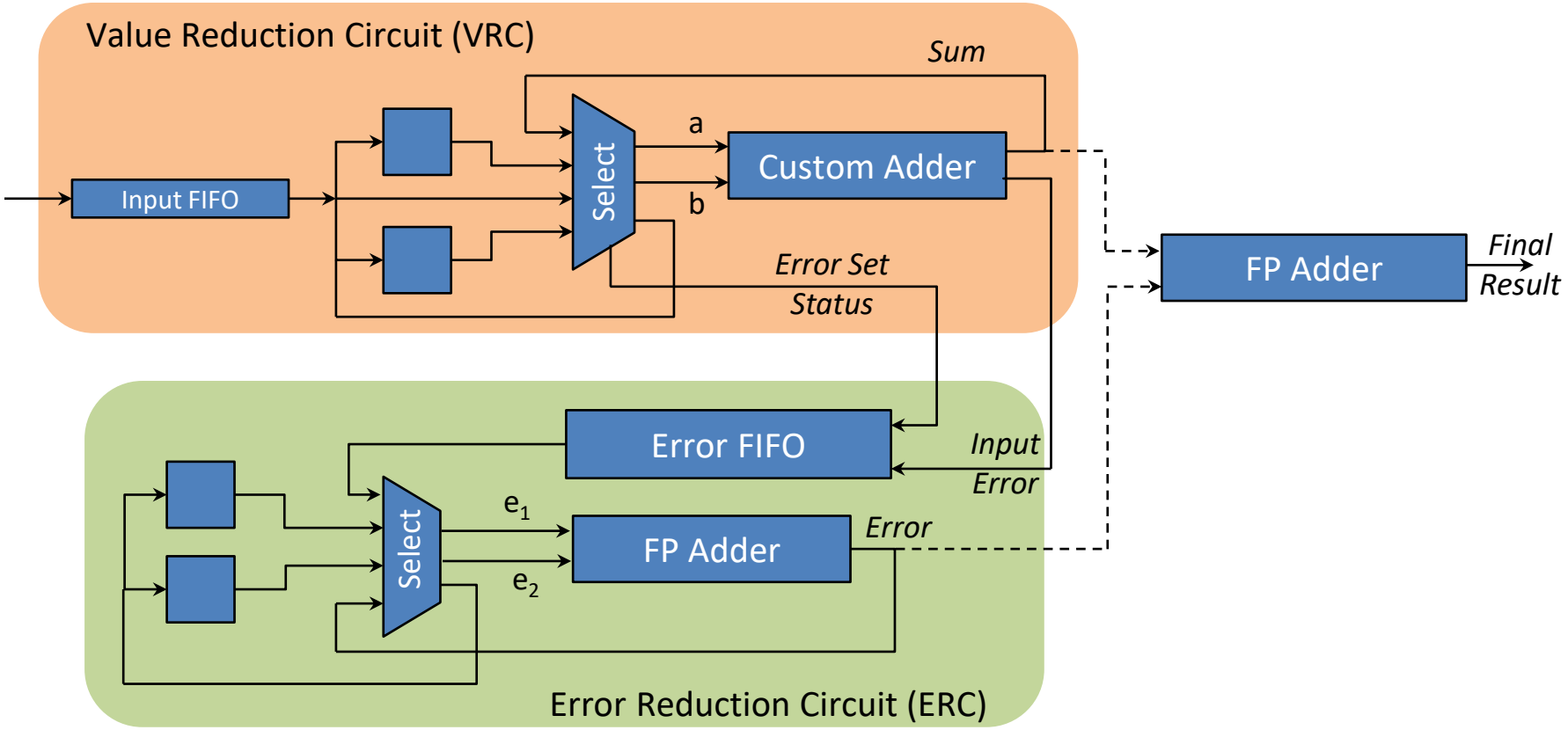
Design	Slices	Latency	%Change Slices
64-bit FP adder	1130	14	
64-bit "error producing" adder	2310	14	+104.4%
80-bit FP adder	1715	19	+51.7%
128-bit FP adder	3327	26	+194.4%

Adaptive Error Compensation in Subsequent Addition (AECSA)



Accumulated Error Compensation

- Accumulate the extracted error, compensate in the end



Preview of Results

Varying κ and Exp. Range = 32, Set Size = 100

$$\kappa = \frac{\sum_{i=1}^{n} |ai|}{|\sum_{i=1}^{n} a_i|}$$

Number of LSBs in Mantissa in Error
(Compared with infinite precision)

Exp. Range	κ	Red. Ckt.	AECSA	AEC	EPRC80	EPRC128
32	10.0	1.6	0	0	0	0
32	95.0	3.5	0.3	0.3	0.2	0.2
32	800.0	4.2	0.7	0.6	0.3	0.3
32	1600.0	7.7	0.9	0.7	0.4	0.4
32	6000.0	7.9	1.4	1.1	1.1	0.9
32	11000.0	8.3	2.5	1.6	1.4	1.4