Resistive Bloom Filters: From Approximate Membership to Approximate Computing with Bounded Errors

Vahideh Akhlaghi[1], Abbas Rahimi[2], Rajesh K. Gupta[1]
[1]Department of Computer Science and Engineering, UC San Diego, La Jolla, CA, USA
[2]Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA, USA
{vakhlagh,gupta}@cs.ucsd.edu, abbas@eecs.berkeley.edu

Approximate Computing

- Semiconductor technology scaling necessitates energy efficient design of massively parallel architectures.
- Emerging applications such as image processing, pattern recognition, and data mining show tolerance to the precision of computations.

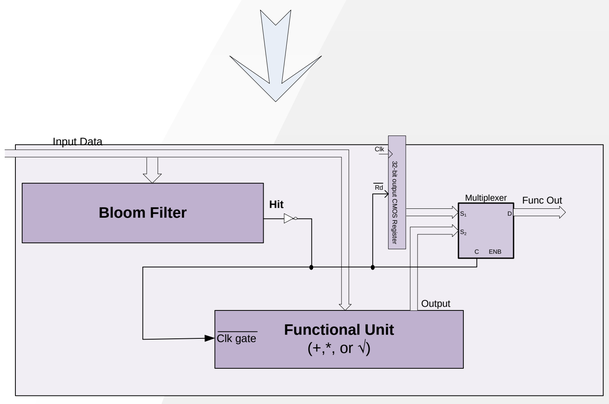
Sacrificing accuracy for energy efficiency

Non-Volatile Memories

- Energy efficient memories
- High density and near zero leakage power
- Resistive RAM (ReRAM): low-write energy, fast read access time and low read voltage
- A 1-Mb 2T-2R ternary content addressable memory (TCAM) compared to SRAM-based TCAMs: low power, high capacity

Function Approximation with Bounded Error Rate

- Static Profiling of functional units used by image processing applications
- Saving frequent input patterns of frequent outputs of functional units in Bloom filter
- Integrating the Bloom Filter into the Functional unit
- Search the input in the Bloom filter
- If input exists, clock gate the functional unit



- False positive (FP) errors appear in the result of Bloom filter queries.
- The error leads to incorrect function outputs.
- The rate of the error can be controlled by Bloom filter configuration.

$$FP = (1 - e^{-\frac{nk}{m}})^k$$

n = number of input patterns
k = number of hash functions
m = size of the bloom vector

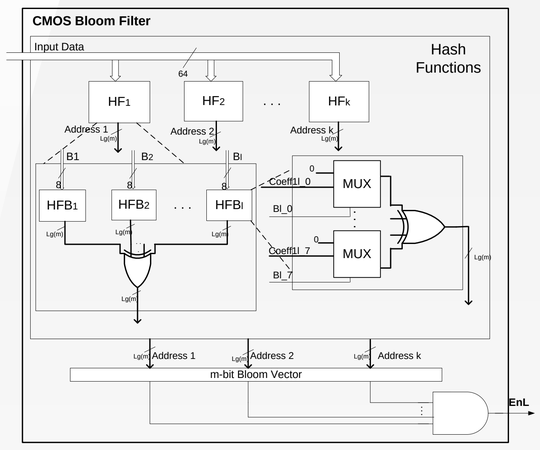
Tolerable error rate of image processing applications

Application	ADD	MUL	MAC	SQRT	PSNR (min, max, avg) (dB)
Sobel Filter	0.001	0.0001	0.001	0.001	(26.4,32.9,28.0)
Sharpen	–	0.01	0.01	0.01	(24.7,36.8,28.12)
Roberts	–	0.001	0.001	0.001	(25.4,32.5,27.9)
Prewitt	–	0.01	0.001	0.01	(25.2,33.3,27.1)
Scharr	–	0.001	0.0001	0.001	(26.1,31.1.27.2)

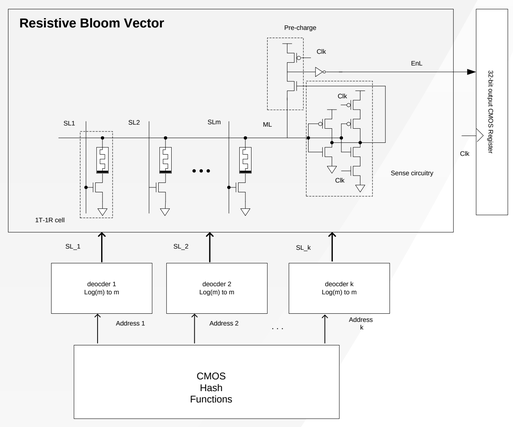
Optimum ReBF configuration for different applications

FU	Sobel Filter					Roberts					Sharpen					Prewitt					Scharr				
	#Out	#In	HR%	BV	#Fn	#Out	#In	HR%	BV	#Fn	#Out	#In	HR%	BV	#Fn	#Out	#In	HR%	BV	#Fn	#Out	#In	HR%	BV	#Fn
ADD	2	12	29.5	256	4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
MUL	2	16	29.6	1024	3	6	26	41.6	512	4	6	15	42.2	256	2	8	30	59.4	512	2	4	12	27.7	256	4
MAC	4	12	25.4	256	4	10	12	29.5	256	4	6	15	30.2	256	2	6	15	32.6	1024	2	8	18	37	512	4
SQRT	16	16	20.8	512	3	10	10	29.5	256	4	–	–	–	–	–	14	14	82	256	2	18	18	10	512	3

Resistive Bloom Filter (ReBF) vs CMOS Bloom Filter

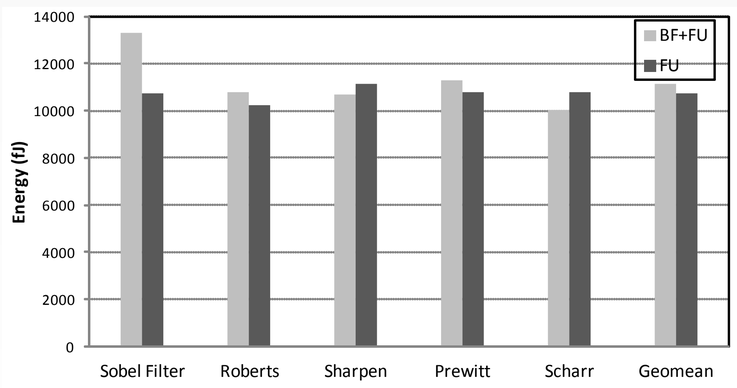


CMOS implementation of Bloom filter

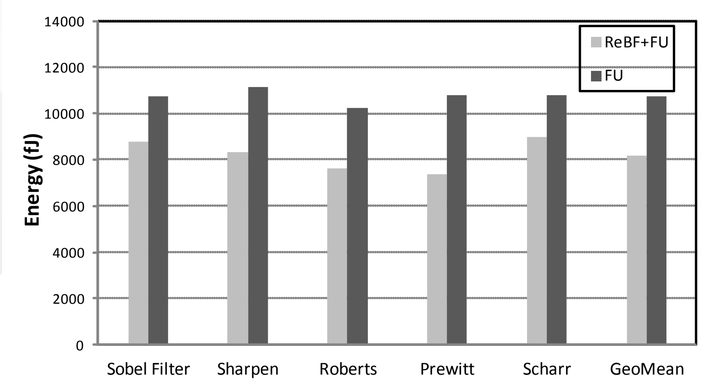


1T-1R implementation of ReBF

Experimental Results



Energy consumption comparison of the proposed architecture using CMOS BFs and conventional FU



Energy comparison of the proposed architecture using ReBFs and conventional FUs.

Energy consumption comparison of Resistive and CMOS Bloom vector

size (bits)	2048	1024	512	256	128	64
Vdd (V)	1	0.71	0.6	0.54	0.51	0.47
RBV (fJ)	19.3	17.12	8.04	5.1	3.77	1.62
CMOS BV (fJ)	12188.4	6662.4	3992.4	2603.1	1837.35	1649.55

Function approximation using ReBF for five image processing kernels running on the AMD Southern Islands GPU yields on average 24.1% energy saving in 45 nm technology compared to the exact computation.
ReBF represents on average 38.42% of the functionality of FUs in five different kernels running on GPUs, while guaranteeing the acceptable outputs with PSNR of greater than 27 dB.