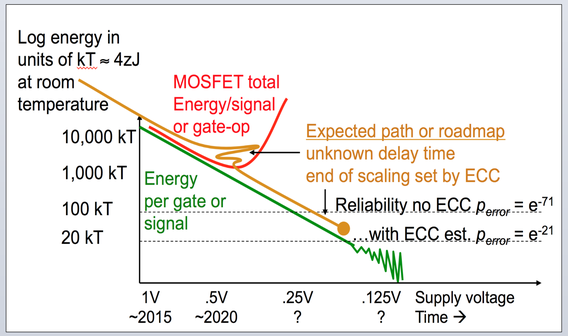
Fixed-point Arithmetic of Redundant Residue Number System

Bobin Deng, Thomas M. Conte
School of Computer Science
Georgia Institute of Technology

Erik Debenedictis
Zettaflops

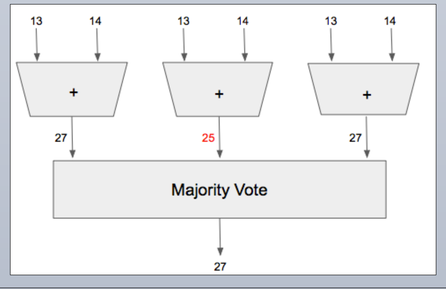
Jeanine Cook
Sandia National Laboratories

- Dennard Scaling has ended. Leverage "marginal devices".
  - Post-CMOS/millivolt switches become unreliable with lower voltages.
- Achieve lower energy in high throughput exascale HPC systems by lowering V_{DD} while correcting errors.
- Develop an error-tolerant microarchitecture for these "marginal devices".



Error rate grows as its signal energy is reduced.

Traditional Computational Reliability: Triple Modular Redundancy (TMR)



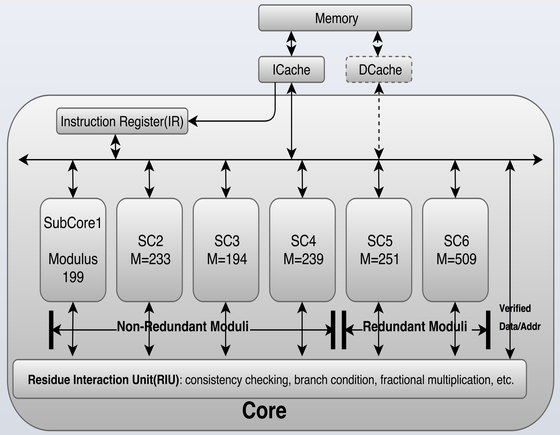
200% energy overhead; unacceptable

Redundant Residual Number System (RRNS)

- Each residue is independent under addition, subtraction and multiplication (no carries!).
- Using two redundant residues, can correct a single error in any of the residues or detect two.

Decimal	mod 3	5	2	7	11	13
6	0	1	0	6	6	6
9	0	4	1	2	9	9
6 + 9 = 15	(0 + 0) mod 3 = 0	0	1	1	4	2
6 x 9 = 54	(0 x 0) mod 3 = 0	4	0	5	10	2

- Moduli are relatively prime.
- Range = Product of non-redundant moduli
  - 3 x 5 x 2 x 7 = 210
  - 199 x 233 x 194 x 239 ~= 32 bits
- Redundant moduli: (11, 13) or (251, 509) respectively

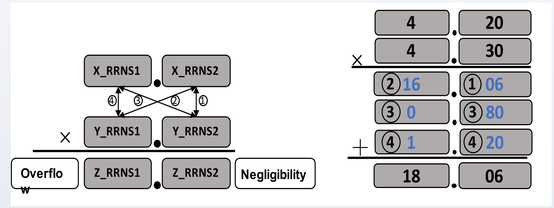


RRNS Fixed-Point Arithmetic

- The IEEE 754 floating-point arithmetic works efficiently for the binary number system thanks to its efficient bit-shift operations (for normalize/denormalize).
- Bit-shifting can be implemented as multiplication and optimized scaling algorithms. However, they are more time-consuming and energy-inefficient.
- To address these problems, we propose two alternatives of RRNS fixed-point arithmetic for RRNS based microarchitectures.

1) 2-RRNS Concat Representation

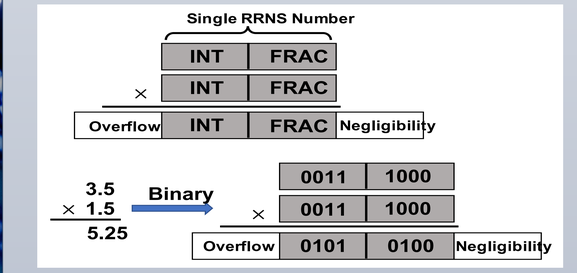
- 2-RRNS Concat proposes using two RRNS integer values to represent the integer and fractional segments of a fixed-point number separately.
- Multiplication: In order to maintain bit-width consistency, the highest (overflow) and lowest (negligibility) segments of the result are cut.



- Steps 1-4 compute the intermediate results.
- This multiplication requires 4 RRNS integer multiplications, 3 RRNS fractional multiplications, 4 RRNS additions, and 2 overflow detections.

2) RRNS Logical Partition Representation

- The key idea of RRNS Logical Partition is to use only one RRNS integer number to represent a fixed-point number for both integer and fractional parts.
- This multiplication requires 2 RRNS scaling down and 1 RRNS multiplication.



References

[1] Deng, B., Srikanth, S., Hein, E. R., Conte, T. M., Debenedictis, E., Cook, J., & Frank, M. P. (2018). Extending Moore's law via computationally error-tolerant computing. ACM Transactions on Architecture and Code Optimization (TACO), 15(1), 1-27