Los Alamos
NATIONAL LABORATORY

Number Representations and Their Applications to Hardware Devices

Matthew E. Broussard[1] Andrew C. Alexander[2]

[1]Department of Mathematics, North Carolina State University

[2]Department of Mathematics, Princeton University

SULI Presentation Session
July 29, 2021

LA-UR-21-27242

Introduction

Purpose

- Numbers can be represented in many different ways
  - write numbers in decimal and code in binary
- The way a computer represents numbers can affect:
  - storage efficiency
  - energy requirements
  - performance
- Different number representations suit different devices

Our goal: *Map hardware to number systems to make computers and projects more efficient.*

Introduction

Example

15

2^3 2^2 2^1 2^0

1 1 1 1

3^3 3^2 3^1 3^0

1 -1 -1 0

$(-3+i)^3$ $(-3+i)^2$ $(-3+i)^1$ $(-3+i)^0$

1 5 4 5

mod: 7 5 3 2

1 0 0 1

p = 0.15

000000010000010000000000000010100…

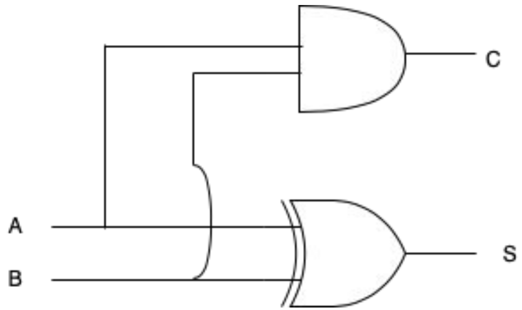
Introduction

Present Technology and Issues

- Prevailing system on CMOS machines: *2's complement binary*
  - → widespread use on existing and new devices
- But can we do better?
- As the end of Moore's Law draws closer, binary representation may not be the most resource efficient system for *all* new architectures

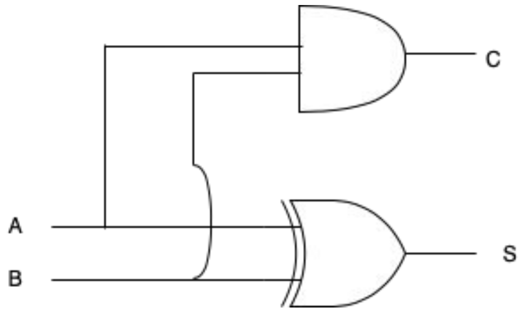
Introduction

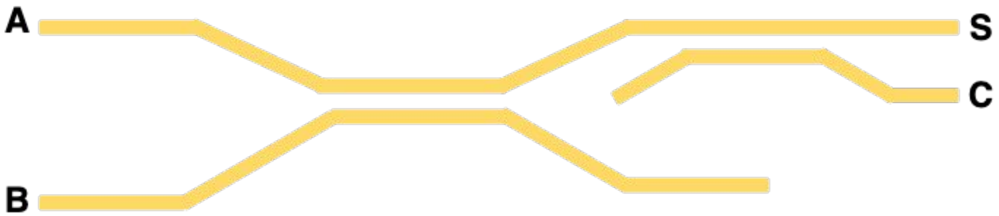
Hardware Devices



Introduction

Hardware Devices

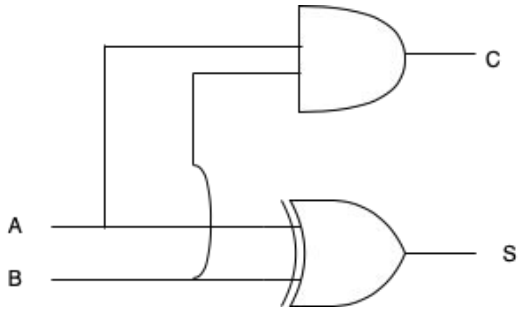


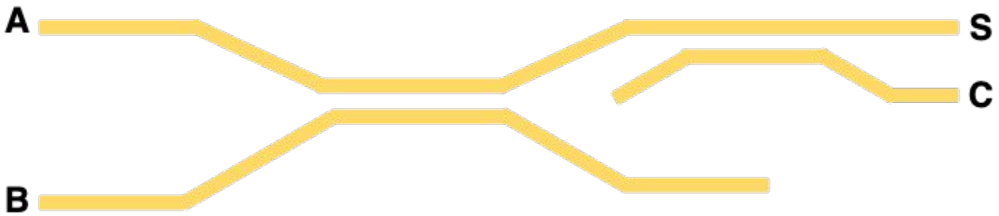


[1]

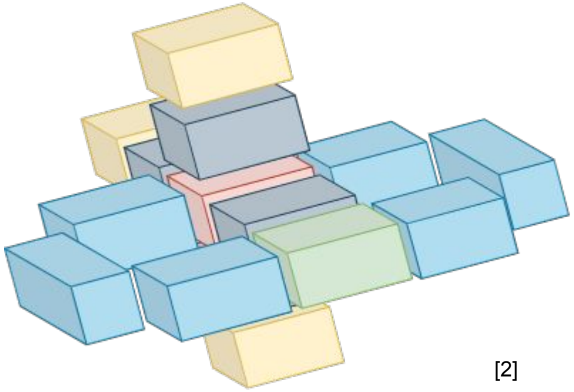
Introduction

Hardware Devices





[1]



[2]

Introduction

New Contributions and Benefits

- Compiling results about representations and hardware
  - ensure new projects are as resource-efficient as possible
  - easy resource for upfront knowledge about hardware-rep mappings
- Reconsidering old ideas in the context of current technology
  - new uses for antiquated number systems
- Establishing some framework for future communication
  - mathematical research in tandem with hardware development

Representations

Representations

Overview

Familiar →		Strange →		Wacky

	Integers
(blue)	Binary
(blue)	2's Complement
(blue)	Excess Binary
(light blue)	Binary Signed Digit (BSDR)
(light blue)	Balanced Ternary
(light blue)	Multivalue Logic
(magenta)	Phinary
(magenta)	Complex Base
(yellow)	Residue Number System (RNS)
(yellow)	Gray Code

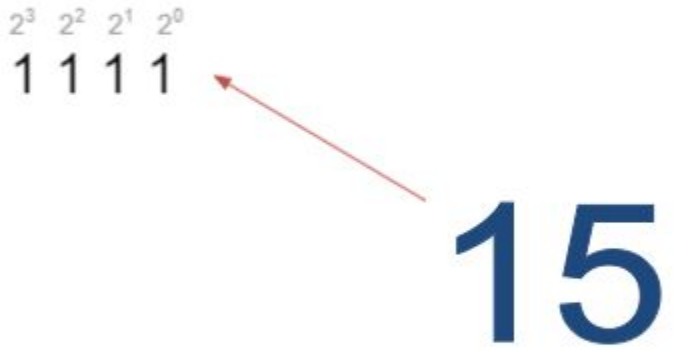
	Reals
(light blue)	Floating Point
(magenta)	Sign Logarithm
(magenta)	Posits
(black)	Stochastic Computing

- Attributes:
  - negatives in Red
  - positives in Blue

Representations

Binary

- Numbers stored as sums of powers of
- Easy to store values with binary
  - value is either on or off
  - large difference between states
- Arithmetic with multiples of 2 is simple
  - multiplication, shift bits to the left
  - division, shift bits to the right
  - mod, mask bits
- Representations attempt to solve negative problem



Representations

Binary: 1’s Complement

- Negative integers stored as compliments
  - complement all bits of positive to represent negative
  - most significant bit stores sign
- Very easy to calculate negative numbers
- Representation for 0 is all 0’s
  - there is another representation for 0
  - “negative 0” is represented by all 1’s
  - occurs in addition, causes carry delays
- Complicated Arithmetic
  - addition requires wrap around calculation
  - multiplication requires shifting partial sums

Representations

Binary: 2's Complement

- Very familiar system
- Negatives are complement of pos. representation + 1
  - solves ±0 issue
  - simplifies addition
- Minor issues
  - magnitude of negatives is difficult to determine
  - negative integers are more complicated to compute
    - than other binary systems
- Predominately used on most CMOS computers today

Representations

Binary: Excess

- No signed bit, string of 0’s represents smallest value
  - excess 8 has 0000 = -8, 1000 = 0
- Difficult computations
  - repeated addition requires modifying answer[3]
  - multiplication representation changes based on inputs
- Easy to store values
  - magnitude is easy to tell for all integers
- Used in IEEE floating point

Representations

Binary: Binary Signed Digit (BSDR)

- Integers are sums or differences of powers of 2
  - stores 1,0,-1 values
  - non-unique representations
- Difficulty in storing 3 distinct values
- Efficient calculations
  - reduced carry operations
  - representations with many 0's
  - specific carry free algorithm[4]
- Unique non-adjacent form for all numbers
  - maximum number of 0's[5]

Representations

Balanced Ternary

- Uses sums and differences of powers of 3
  - stores values 1,0,-1
  - unique representations for all integers
- Requires storing a third distinct value
- Arithmetic Benefits
  - multiplication, division and modding by 3
  - rounding to nearest bit
- Most storage-efficient radix to store values with
  - closest integer to euler's number[6]
- Ternary is also used in Logic
  - applications in TCAM searches[7]

Representations

Multivalue Logic

- Integers are stored using sums of powers of 4 and greater
  - unique representations for all positive integers
- Fewer powers needed to store numbers
  - results in less memory cells used
- More distinct values need to be stored on each cell
  - lower noise margins
  - 3 is the most storage-efficient, 2 and 4 same storage-efficiency
  - for 5 and greater, storage-efficiency decreases[6]

Representations

Redundant Logic

- Broad term encompassing all non-unique systems
  - BSDR is a specific example
- Requires storing several more values
  - difficulty of representing many values
  - may require storing almost twice as many values
- Benefits for systems with negative and positive weights
  - increased cancellations
  - carry free algorithm

Representations

Negative Base

- Uses sums of powers of negative integers
  - -2 is the most common base
  - can be used with any integer
- No signed bit or negative weights needed
  - alternating bits have alternating signs
- Addition is made difficult
  - overflows now result in 2 carry bits
  - carry bits may result in infinite sum
  - this process can be terminated within 2 calculations

Representations

Phinary

- Non-integer base, sums of the golden ratio
  - uses values 0,1
  - unusual relations between powers
  - unique finite standard form for all positive integers
- Several trade-offs result from this
  - Benefits
    - standard form simplifies multiplication
    - efficient irrational number representations
  - Disadvantages
    - addition becomes more complicated
    - non-integer rational numbers only have infinite representations
    - no signed bit systems[8]

Representations

Complex Base

- Uses sums of powers of (-n+i)
  - unique representation for all Gaussian integers
  - weights are $0,1,\ldots,n^2$
  - most common form is complex binary, -1+i
- Benefits
  - more efficient complex calculation
  - special applications to electronic and harmonic projects
- Disadvantages
  - multiple carry bits
  - possible runaway calculations
  - termination algorithm dependent on base[9]

Representations

Residue Number System (RNS)

- Represents an integer *n* using multimodular arithmetic[10]
  - each digit represents the value of *n* via some modulus
  - moduli are coprime, often of form $2^n \pm 1$
- Benefits
  - addition, multiplication work naively
  - because arithmetic is modular, no carries
  - computation done in parallel
- Problems:
  - determining magnitudes, division
- Applications to digital signal processing, convolutional neural nets

Representations

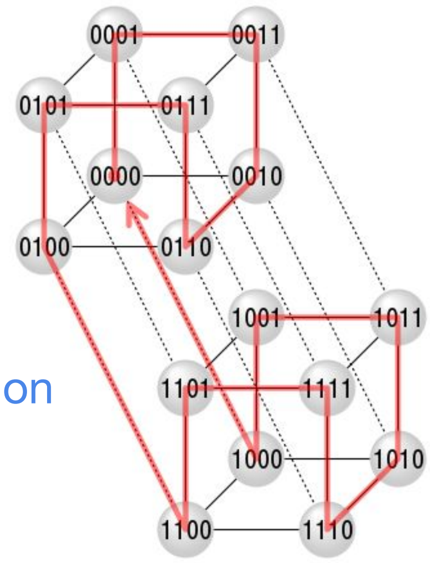
Zeckondorf System

- Numbers are sums of Fibonacci numbers
  - unique Zeckondorf standard form
  - non-adjacent representation
- Error resilient properties
  - adjacent 1’s only at the end
  - code is stored in reverse, msb goes first
  - useful for encoding data with only 0’s and 1’s
- Difficult operations
  - only one relation for carry bits
  - multiplication is extremely difficult
  - quicker to convert to binary and compute

Representations

Gray Code

- Values are stored depending on the previous value
  - adjacent integers have one bit difference
  - representations change for storage sizes
  - first and last values also one bit difference
  - useful in data measurement, analog-to-digital conversion
- Can be extended to real numbers as well[11]
  - approximate representation
  - precision depends on storage size
- Computations possible, but inefficient

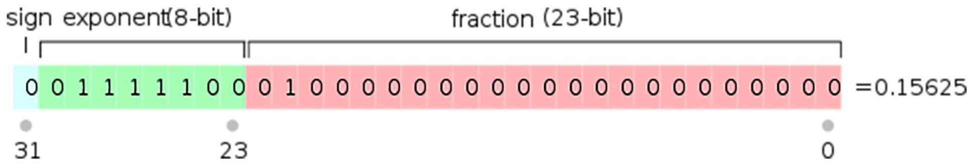


Source: WikiCommons

Representations

Floating Point

- Sign bit, exponent bits, and significand bits
- IEEE standard: base 2, biased exponent (-127 on 32-bit)[12]
- Fixed precision
  - makes math easier
- Standard for decades
  - error detection/correction well studied
  - adders, multipliers, etc. optimized



Source: Microcontrollertips.com

Representations

Sign/Logarithm

- Sign bit and (scaled) logarithm of absolute value
- Avoids slowness of multiplication of traditional binary systems
  - and magnitude issues of residue system
- Addition requires a lookup table[13]
  - with $O(n \cdot 2^n)$ bits of ROM for n-bit addition
- Special purpose processing
  - pattern recognition
  - image enhancement
  - radar processing

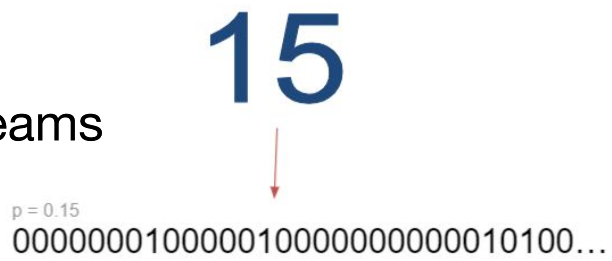
Representations

Posits

- Exponent and significand fields are of *variable bit length*
- Saves space: fits all 64-bit floats into 32 bits[14]
  - no built-in NaNs, like floats
- Tapered precision[15]
  - greater range than floats
  - values around 1 have greater precision than floats
  - precision drops off dramatically at extreme values
- Relatively infant in terms of theory
  - variable precision means harder math

Representations

Stochastic Computing



- Represent numbers probabilistically as bit streams
  - versions with higher-base bit streams
- Multiplication = ANDing two bit streams[16]
- *Progressive precision*
  - good for approximate computations (but bad for math)
- Operation unit AND less costly than FA
- Getting pseudorandom bit streams costly
  - issues of correlation
- Applications to neural nets[17], Low-Density Parity Check (LDPC) codes[18]

Hardware

Hardware

Recommendations

- Not all representations mentioned above are equal
- The most important number systems are
  - Binary/2's Complement
  - BSDR
  - Ternary/Balanced Ternary
  - Multivalue Logic
- While many of the representations above are very interesting
  - the focus is pairing devices with representations efficiently
- Representation recommendation(s) are in **Yellow**

Hardware

Overview

Device	Recommendation(s)
Optical	
Neuromorphic	
Reversible	
Nanomagnetic	
RSFQ	
MAGIC	
MRL	
Magnonics	

Hardware

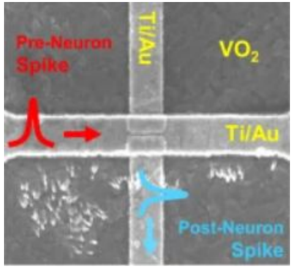
Fourier Optics

- Using lenses and masks to perform Fourier transforms
  - reducing $O(n^2)$ multiplication to $O(n)$ convolution[19]
- Fourier transform happens for free with light
- Biggest problem:
  - carryless convolution accumulates greater-than-base values
  - → increased error, decreased resolution
- Latter could be improved by choosing a number system with fewer carries
  - with **BSDR**, fewer carries → less error and/or higher resolution

Hardware

Neuromorphic Computing

- Mimicking neurons and synapses in hardware
  - many different implementations
- Neurobiological processes use analog (chemical) signals
  - *key feature:* electronic analog components or models of analog
- Leaky integrate-and-fire
  - neuron membrane builds up charge
  - releases potential spike once past a threshold
  - necessitates a **2-state system** in most cases
    - e.g. VO_2 insulator-to-metal transition[20]
- **Gray codes** use as analog-digital converter has potential



Hardware

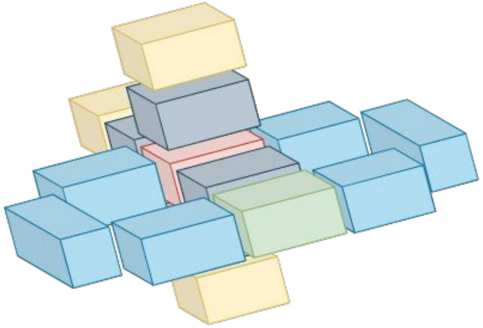
Reversible Computing

- Inputs can be recovered after computation
  - theoretically no heat lost
  - each step is reversible, not just total calculation
  - not a hardware device, but a concept
- Basic gates are used to design complicated systems
  - binary gates have been designed on boolean gate level[21,22]
  - high level designs for ternary gates exist
  - quantum cost for ternary much higher than binary[23,24]
- **Binary** is the current recommendation

Hardware

Nanomagnetic Computing

- Uses nanometer sized magnets for computing
  - magnets start in indeterminate state
  - values are stored in one of two field directions
  - neighbors influence undecided magnets
- Binary is the optimal choice
  - indeterminate state is unstable
  - eventually defaults to an up/down orientation
- 3-d oriented system
  - majority gates are very efficient
  - 3-d needed to minimize errors and maximize space[2]



Hardware

Rapid Single Flux Quantum (RSFQ)

- Superconductor junctions perform logic on voltage pulses
  - devices need to be cooled to 4.2K
- Device space is severely limited
  - chips comparable to 1998 intel chips[26]
- Extremely fast computation[25]
- Current recommendation is **2’s complement**
  - 3 state systems were proposed in 1998[27]
  - ternary is more efficient, only with ternary logic
  - challenging barriers to implement different logic

Hardware

Memristor Aided Logic (MAGIC)

- Memristors are dynamic non-volatile resistors
  - ± voltage adjusts resistance[28]
- MAGIC uses resistance value in storage and logic
  - voltages still needed to “run” calculations[29]
- Current recommendation is **binary**
  - small errors could compound with lower precision states
  - easier to control changing resistance values
  - higher base systems viable if technology improves

Hardware

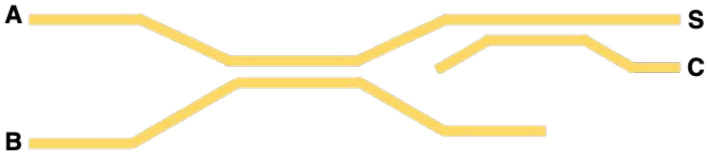
Memristor Ratioed Logic (MRL)

- Uses resistance value as storage
  - but uses voltage value in logic
  - can be used in hybrid CMOS circuits
- Multivalue logic in development
  - AND gate same for binary and ternary
  - utilizes ternary logic
  - early in development[30]
- Storage cells can be used for many systems[31]
  - quaternary cell, binary CMOS system
- **Multivalue/ternary** is the optimal choice here
  - binary circuits see improvements too[28]

Hardware

Magnonics

- Values are stored in magnetic alloy electron spin waves
  - theoretically both extremely small and fast
  - still early in development, slow calculations for now[32]
- Values can be stored in the amplitude and phase
  - perfect for representing the sign and magnitude[33]
  - **BSDR** is perfectly suited to this system
  - also unsigned multivalued logic[32]
- Computing is done via wave interference
  - unsigned systems require more overhead[1]



Conclusions

Hardware Recommendations

Device	Recommendation(s)
Optical	BSDR
Neuromorphic	Gray code, Binary
Reversible	Binary
Nanomagnetic	Binary
RSFQ	2’s Complement
MAGIC	Binary
MRL	Ternary
Magnonics	BSDR, Multivalue

Conclusions

Future Work

- Further investigation into hardware devices
  - this was a brief investigation into this work
  - better results will come from more indepth research
  - including direct work with hardware engineers
- Optimize number systems
  - develop signed bit systems for certain representations
  - explore in more detail specific system benefits
  - combine multiple systems (RNS, sign/log, gray)

Conclusions

Questions?

Andrew Alexander

andrewca@princeton.edu

Matthew Broussard

mbrouss@ncsu.edu

Acknowledgements

Thanks to our mentor Laura Monroe for her guidance, and to Vanessa Job, Nathan Kodama, and Jose Ortiz for their insightful discussions this summer! Thank you to the SULI program coordinators for the opportunity to be at LANL.

Conclusions

References

[1]: https://arxiv.org/pdf/1902.02855.pdf

[2]: https://link.springer.com/content/pdf/10.1007/s10825-019-01304-8.pdf

[3]: https://www.cs.auckland.ac.nz/~patrice/210-2006/210%20LN04_2.pdf

[4]: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6861586

[5]: https://arxiv.org/pdf/2103.05810.pdf

[6]: http://ijcsit.com/docs/Volume%205/vol5issue04/ijcsit2014050473.pdf

[7]: https://ieeexplore.ieee.org/document/34085

[8]: http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fibonacci/phigits.html

[9]: https://www.math.uwaterloo.ca/~wgilbert/Research/ArithCxBases.pdf

[10]: https://link.springer.com/book/10.1007%2F978-3-319-41385-3

[11]: https://arxiv.org/pdf/1904.12763.pdf

[12]: https://dl.acm.org/doi/10.1145/103162.103163

[13]: https://ieeexplore.ieee.org/document/1672765

[14]: http://www.johngustafson.net/pdfs/BeatingFloatingPoint.pdf

[15]: https://www.cs.cornell.edu/courses/cs6120/2019fa/blog/posits/

[16]: https://arxiv.org/abs/1805.06262

[17]: https://ieeexplore.ieee.org/document/9378809

Conclusions

References Continued

[18]: https://digital-library.theiet.org/content/journals/10.1049/el_20030217

[19]: https://ieeexplore.ieee.org/document/8638618

[20]: https://www.nature.com/articles/s41598-020-61176-y

[21]: https://www.sciencedirect.com/science/article/pii/S1877050915019523

[22]: ttps://link.springer.com/chapter/10.1007%2F978-981-13-8821-7_1

[23]:https://www.researchgate.net/profile/Vitaly-Deibuk/publication/329268553_DESIGN_OF_TERNARY_REVERSIBLEQUANTUM_SEQUENTIAL_ELEMENTS/links/5bffa33b299bf1a3c155bb50/DESIGN-OF-TERNARY-REVERSIBLE-QUANTUM-SEQUENTIAL-ELEMENTS.pdf

[24]: https://www.researchgate.net/publication/282246612_Optimized_Design_of_the_Universal_Ternary_Gates_for_QuantumReversible_Computing

[25]: https://iopscience.iop.org/article/10.1088/1742-6596/1559/1/012002/pdf

[26]: http://www.physics.sunysb.edu/Physics/RSFQ/Projects/WhatIs/rsfqre2m.html

[27]: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=679270

[28]: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8676033

[29]: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9125983

[30]: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9214881

[31]: https://www.sciencedirect.com/science/article/pii/S0026269215002530

[32]: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6880494

[33]: https://aip.scitation.org/doi/pdf/10.1063/1.5042417