Custom Number Systems to Minimize Energy Consumption

FPTalks 2021

Theodore Omtzigt
Stillwater Supercomputing, Inc.
https://linkedin.com/in/theodoreomtzigt

Digital Transformation

Source: 2015 ITRS 2.0 Executive Report

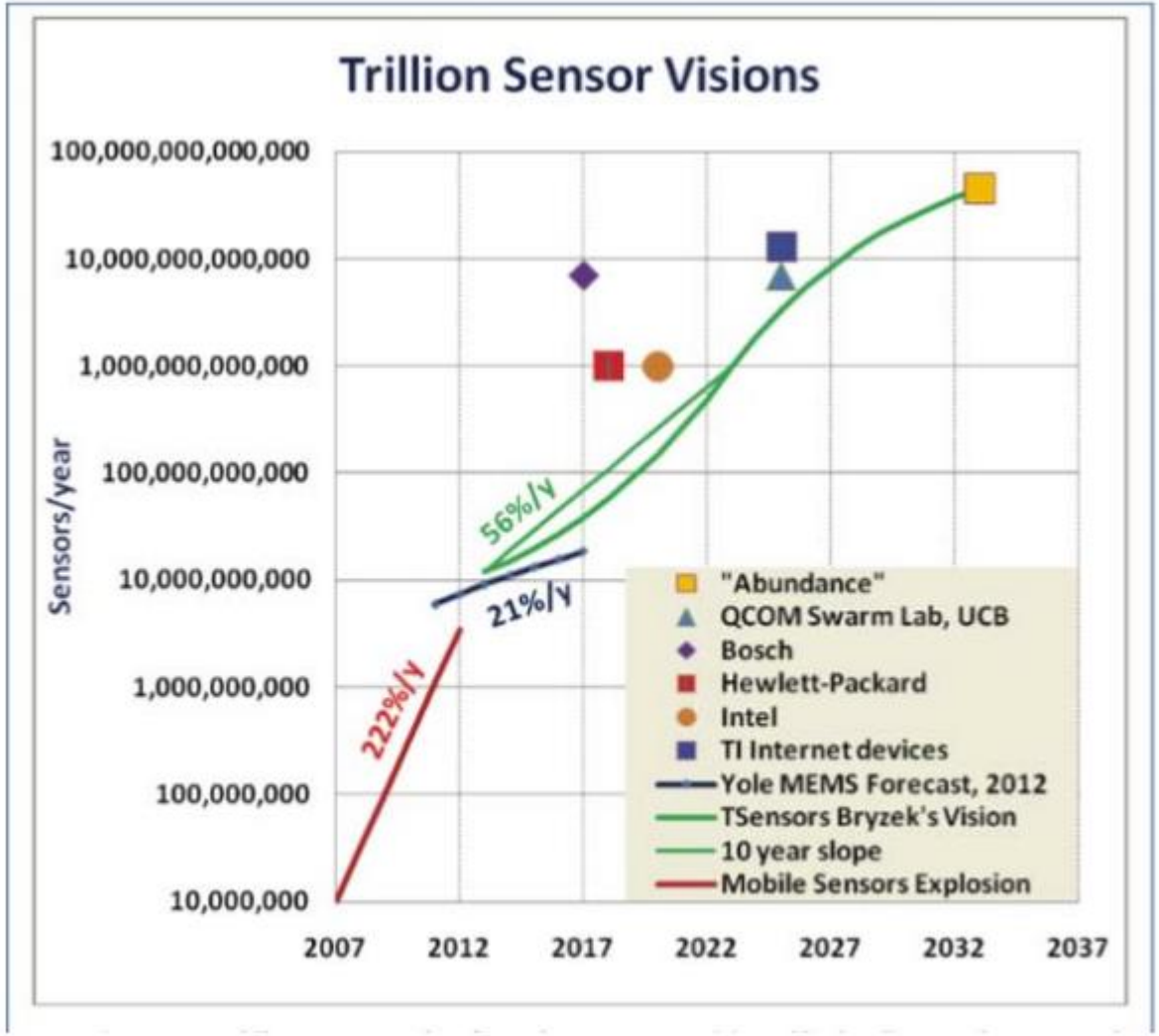
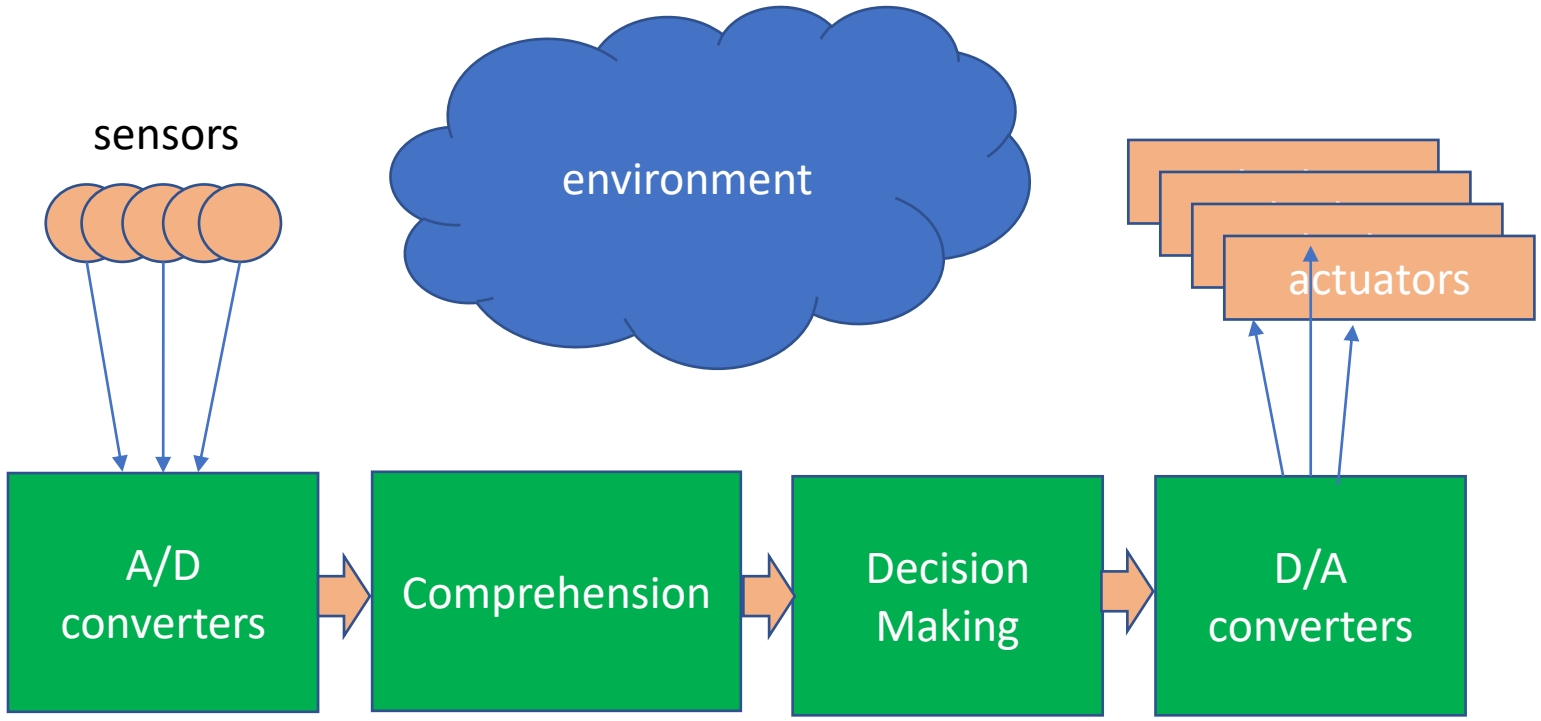


Fig. 5.2 Sensors will populate the world of the IoE

Embedded Intelligence



Requirements tailored to environment, sensors, and actuator dynamics.

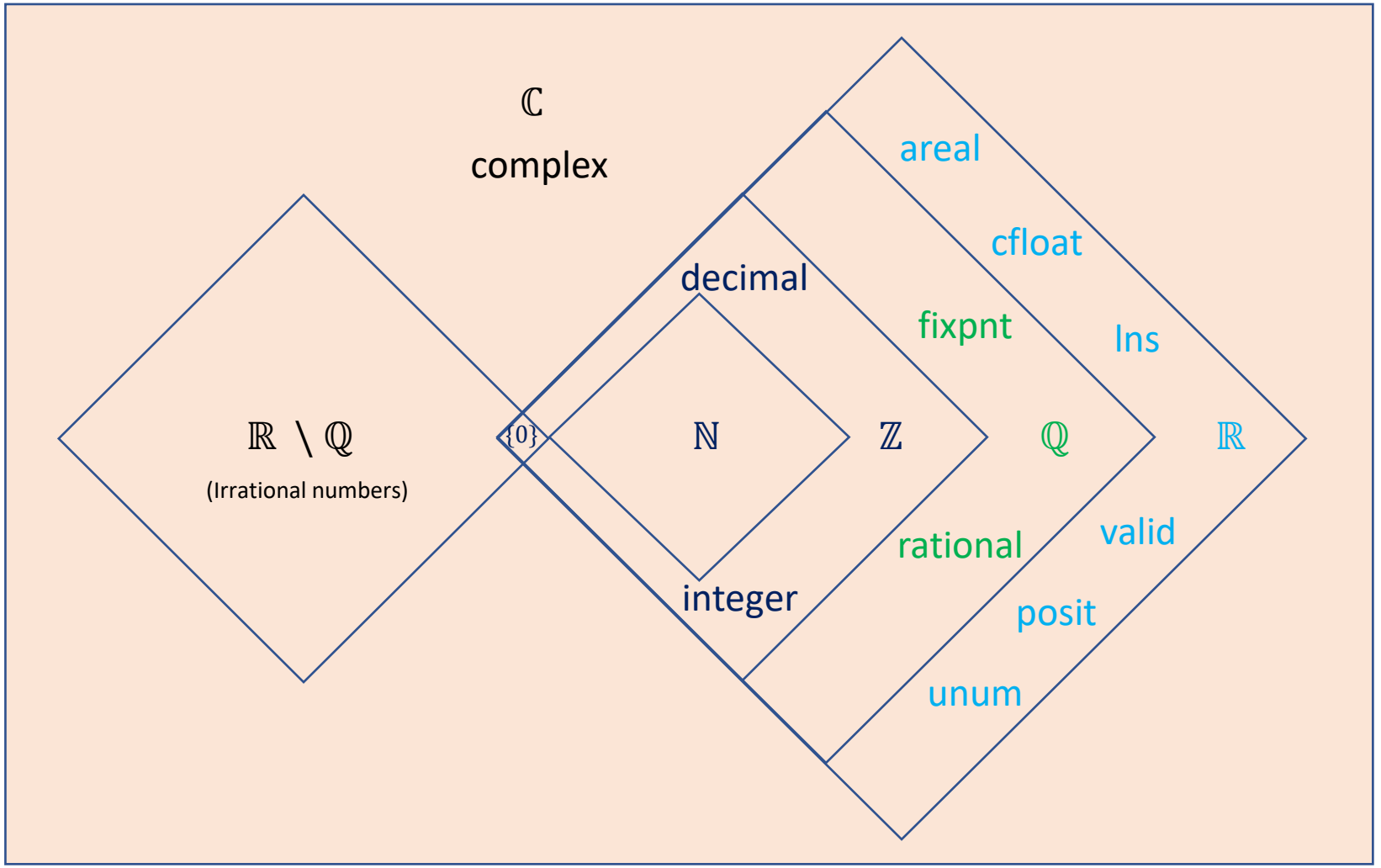
How to maximize efficiency and lower cost?

- Tailor data structure and compute to application
- Avoid unnecessary off-chip accesses
- Custom number systems with minimum precision and dynamic range to get the job done

Universal V3

A versatile **Number Systems** Library

Mathematical Number Sets and associated *Universal* Number Systems



Infinite Precision
- decimal
- adaptiveint
- adaptivefloat
- adaptiveposit

$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$

$\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$

- Parameterize algorithms along data type
- Explore refinement
  - Precision
  - Dynamic range
  - Arithmetic
  - Sampling Profile (linear, exponential, logarithmic, custom, etc.)
- Leverage *Fused Dot Product*
  - Theoretical max benefit is a 2x reduction in operand bandwidth
- Measure numerical error
  - Forward error propagation
  - Theoretical predictions tend to be too conservative

Number System:

decimal

```
class decimal;
```

- Exploratory/Test number system
- Mathematical experiments on representation and precision
- Arbitrary precision and dynamic range
- Implemented as a string of digits [0,9]

Number System:

integer

```
template<size_t _nbits, typename BlockType = uint8_t>
class integer;
```

- Arbitrary Precision, fixed-size integer of *nbits*
- 2’s complement signed integer
- User-directed block size
  - Minimum set of blocks to contain the integer
  - Optimal memory layout for linear-algebra
  - Default block size is 8bits
- Smallest integer is integer<2>
- Largest integer is limited by memory
  - We have tested nbits = 1,048,576

Number System:

fixpnt

```
template<size_t _nbits, size_t _rbits, bool arithmetic = Modulo,
typename bt = uint8_t>
class fixpnt;
```

- Arbitrary Precision, fixed-point number of *nbits* with radix point at *rbits*
- 2's complement signed integer with *rbits* normalizing shift
- Either Saturating or Modulo arithmetic
- User-directed block size
  - Minimum set of blocks to contain the fixpnt
  - Optimal memory layout for linear-algebra
  - Support for *Fused Dot Product*
  - Default block size is 8bits
- Smallest fixpnt is fixpnt<2,2>
- Largest fixpnt is limited by memory

Number System:

cfloat

```
template<size_t _nbits, size_t _es, typename bt = uint8_t,
bool hasSubnormals, bool hasSupernormals,
bool isSaturating>
class cfloat;
```

- Arbitrary Precision, fixed-size classic float of *nbits* and an exponent field of size *es* bits
- Selectable gradual underflow (subnormals), gradual overflow (supernormals), and saturation
- 1 bit Signalling/Quiet NaN, 1bit +/-infinite
- + and - zero
- User-directed block size
  - Minimum set of blocks to contain the cfloat
  - Optimal memory layout for linear-algebra
  - Support for *Fused Dot Product*
  - Default block size is 8bits
- Smallest cfloat is cfloat<3,1>
- Largest *es* is currently 11

Number System:

areal

```
template<size_t _nbits, size_t _es, typename bt = uint8_t>
class areal;
```

- Arbitrary Precision, fixed-size linear float of *nbits* and an exponent field of size *es* bits
- Gradual underflow (subnormals) and gradual overflow (es == all 1's)
- Uncertainty bit to model (v, v+ULP) interval
- User-directed block size
  - Minimum set of blocks to contain the areal
  - Optimal memory layout for linear-algebra
  - Default block size is 8bits
- Smallest areal is areal<4,1>
- Largest *es* is currently 11

Number System:

posit

```
template<size_t _nbits, size_t _es, typename bt = uint8_t>
class posit;
```

- Arbitrary Precision, fixed-size tapered float of *nbits* and *es* exponent bits
- Exponential regimes of *useed* = $2^{2^{es}}$
- Saturating arithmetic to *minpos*/*maxpos*
- 1 code for NaR, 1 code for 0
- Maximum precision at 1.0 at (*nbits – es – 1)* fraction bits
- User-directed block size
  - Minimum set of blocks to contain the posit
  - Optimal memory layout for linear-algebra
  - Default block size is 8bits
- Smallest posit is posit<2,0>
- Largest posit is posit<256,5>

Number System:

lns

```
template<size_t nbits, typename bt = uint8_t>
class lns;
```

- Arbitrary Precision, fixed-size logarithmic number system of *nbits*
- Values represent $\log_b(|x|)$
- Represented as 2's complement number
- User-directed block size
  - Minimum set of blocks to contain the lns
  - Optimal memory layout for linear-algebra
  - Default block size is 8bits

Number System:

valid

```
template<size_t _nbits, size_t _es, typename bt = uint8_t>
class valid;
```

- Arbitrary Precision, fixed-sized valid of *nbits* and *es* exponent bits
- An interval type with posits as lower/upper bound values plus an uncertainty bit
- User-directed block size
  - Minimum set of blocks to contain the valid
  - Optimal memory layout for linear-algebra
  - Default block size is 8bits

Number System:

unum

```
template<size_t esizesize, size_t fsizesize, typename bt = uint8_t>
class unum;
```

- Variable Precision floating-point of maximum *esizesize* exponent bits and maximum fsizesize fraction bits
- Experimentation type to capture precision of complex computations
- User-directed block size
  - Minimum set of blocks to contain the unum
  - Optimal memory layout for linear-algebra
  - Default block size is 8bits

Application Examples

- Application Integrations:
  - G+SMO: Iso Geometric Analysis Package
  - FDBB: Fluid Dynamics Building Blocks
  - MTL4 and MTL5 linear algebra engines
  - AutoDiff: Automatic Differentiation engine
  - ODEint: Boost Library for ODE solvers
  - LibKet: Quantum Computer simulator
- Posits with FDP
  - Chebyshev Polynomials for Approximation
  - Reversible FFTs
  - Perfect Matrix Inverses
  - Krylov IDR(s)
  - AI/DL training with 8-bit posits
- Integers
  - Factorization
  - Irrational number approximation
  - Quantum Safe Crypto
- Fixpnt and Posit
  - Quantum Expression Template Library
  - DSP and Spectral Analysis
- Bfloat vs Posit
  - FPGA hardware efficiency research
- Bfloat/areal & posit/valid
  - Numerical error analysis research

Conclusions

- Energy efficiency is differentiating embedded intelligence
- Tailor the compute to the application to maximize efficiency
- *Universal* presents an SDK to develop and optimize mixed-precision algorithms