inesc id lisboa

TECH NOLOGY FROM SEED

UNCONVENTIONAL COMPUTER ARITHMETIC FOR EMERGING APPLICATIONS AND TECHNOLOGIES

IEEE COMPUTER SOCIETY DISTINGUISHED VISITORS PROGRAM (DVP)

https://www.computer.org/web/chapters/dvp

Leonel Sousa

Webinar, July 9, 2020

TÉCNICO LISBOA

IEEE COMPUTER SOCIETY DISTINGUISHED VISITOR

UNCONVENTIONAL COMPUTER ARITHMETIC FOR EMERGING APPLICATIONS AND TECHNOLOGIES

IEEE COMPUTER SOCIETY DISTINGUISHED VISITORS PROGRAM (DVP)

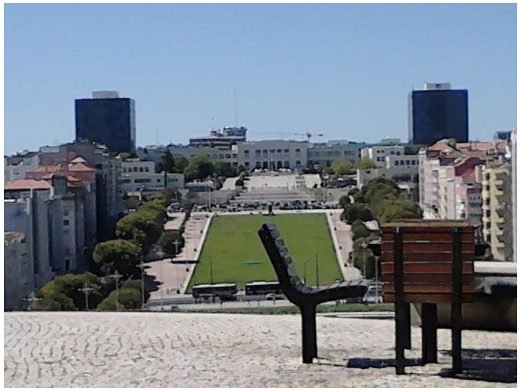
https://www.computer.org/web/chapters/dvp

Leonel Sousa

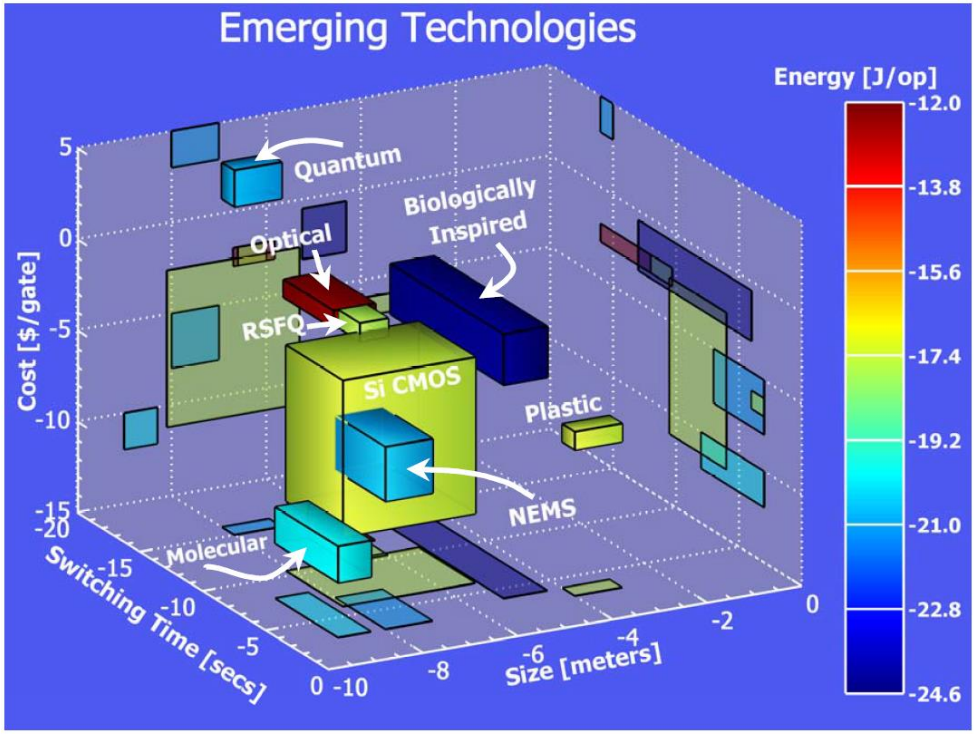
Webinar, July 9, 2020

From Lisbon

- IST (Head of the ECE Dept)
  - Faculty of Engineering University of Lisbon
  - ~9000 / ~55000 students
- INESC-ID
  - Research institute
    - 200 PhD researchers and
    - 300 Graduate Students
  - Main research areas
    - Spoken Language Systems
    - Information and Decision Support Systems
    - Interactive Virtual Environments
    - Embedded Electronic Systems
    - Communication Networks and Mobility



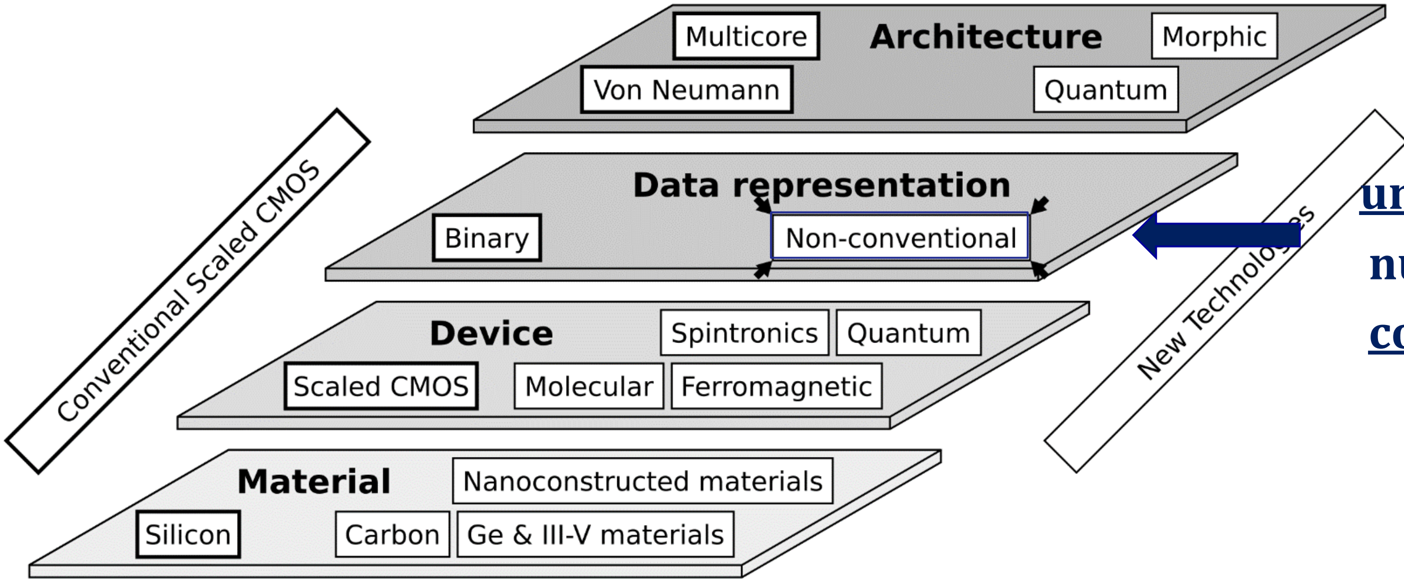
Motivation



Cost, speed, size, and energy/operation encoded by color [ITRS03]

- No solution with few major drawbacks as CMOS along *all* axes
  - spin transistors, superconducting electronics, molecular electronics, resonant tunneling devices, QCA, and optical switches

Motivation

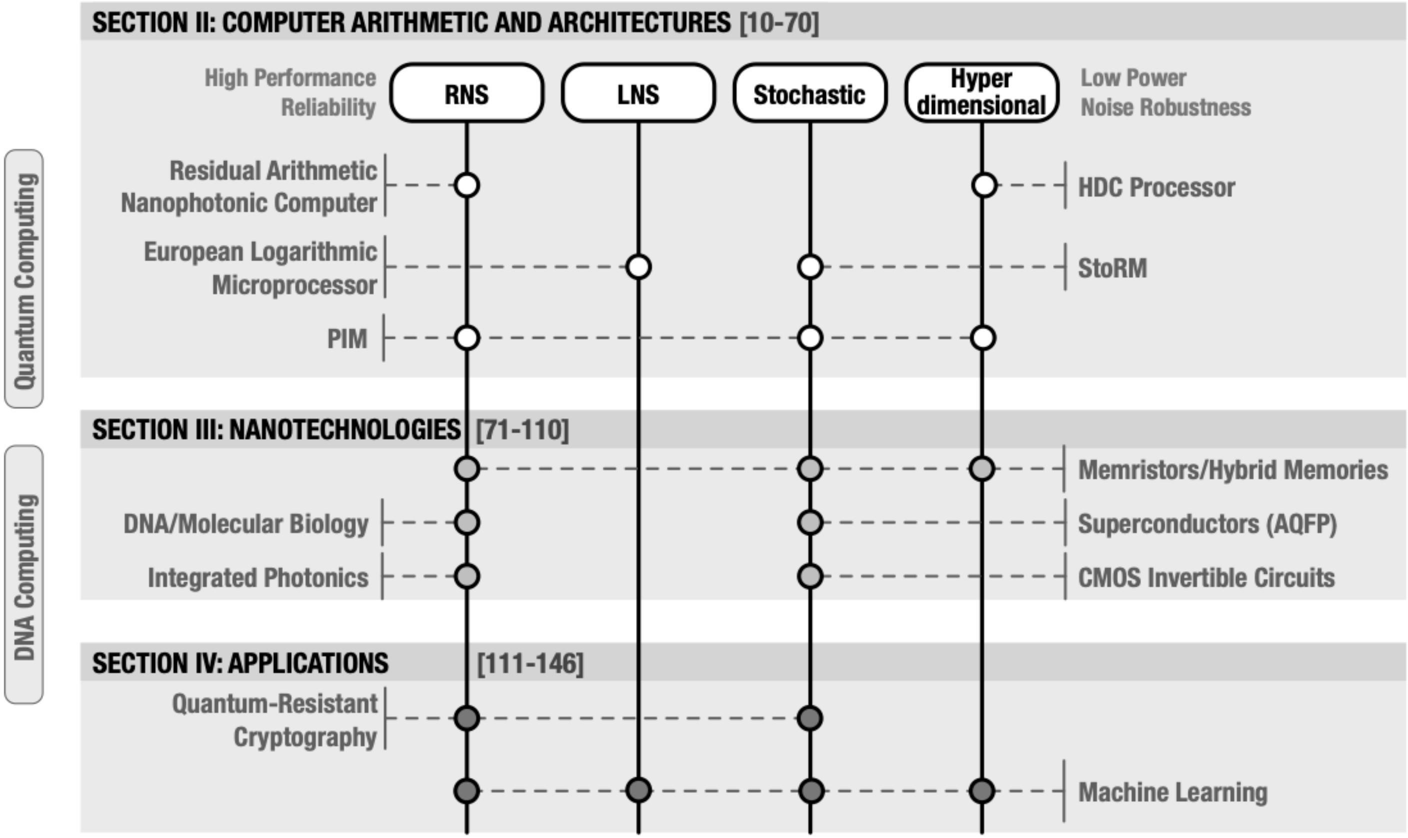


unconvetional number systems & computer arithmetic

Nano processing technologies [ITRS15-Beyond CMOS]

- scaled CMOS
  - FinFET non-planar transistor dominant gate design for current 7 nm
- emerging nano-technologies require

Unconventional Computer Arithmetic [IEEE Journal]



- Confluence of non-conventional computer arithmetic, new computing paradigms, emergent technologies and applications
  - jigsaw puzzle: connecting pieces in the right way to get the whole picture

Outline

1. Logarithmic Residue Number Systems (LNS)
2. Residue Number Systems (RNS)
3. Stochastic Computing (SC)
4. Hyper-Dimensional Computing (HDC)
5. DNA Computing
6. Quantum Computing
7. Applications:
   A. Lattice-based Post-Quantum Cryptography
   B. Machine Learning
8. Conclusions

LNS

1. Logarithmic Number Systems (LNS)
2. Residue Number Systems (RNS)
3. Stochastic Computing (SC)
4. Hyper-Dimensional Computing (HDC)
5. DNA Computing
6. Quantum Computing
7. Applications:
   A. Lattice-based Post-Quantum Cryptography
   B. Machine Learning
8. Conclusions

inesc id lisboa

LNS

S_{FP} (sign bit)	E_{FP} (exponent): 8 bits	F_{FP} (mantissa): 23 bits
S_{LNS} (sign bit)	IT_{LNS} (integer): 8 bits	F_{LNS} (fractional): 23 bits

$$P = -1^{S}_{FP} \times 1.F_{FP} \times 2^{E_{FP}-127}$$

$$p = -1^{S}_{LNS} \times 2^{IT_{LNS}.F_{LNS}}$$

Absolute values	minimum	maximum
P (FP)	$1.0 \times 2^{-126} \approx 1.2 \times 10^{-38}$	$2 \times 2^{+127} \approx 3.4 \times 10^{+38}$
p (LNS)	$1.0 \times 2^{-128} \approx 2.9 \times 10^{-39}$	$2 \times 2^{+127} \approx 3.4 \times 10^{+38}$

- Simple logarithmic operations come at the cost of more complex +,-

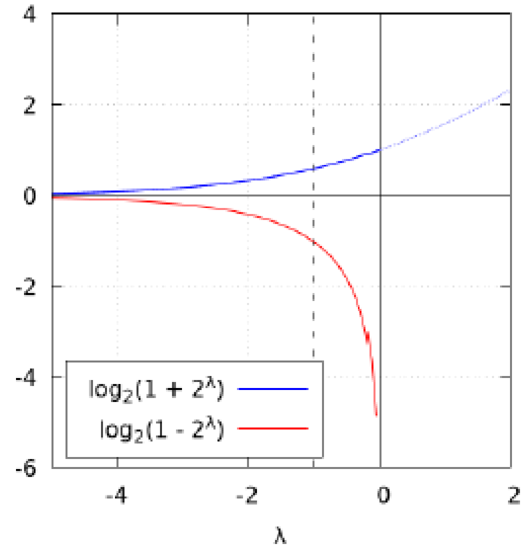
$$\begin{aligned}
\log_b(P * Q) &= p + q \ ; \\
\log_b(P/Q) &= p - q \ ; \\
\log_b(P^2) &= 2 * p = BitShift(p, 1) \ ; \\
\log_b(\sqrt{P}) &= \frac{1}{2} * p = BitShift(p, -1) \ .
\end{aligned}$$

inesc id lisboa

LNS

- Addition/subtraction in LSN apply Gaussiam Logarithms

$$G = \log_2(1 \pm 2^{\lambda}) \quad , \quad \lambda = -|q - p|$$



- For high number of bits (32,64) piecewise polynomial approximation or digit-serial iterative methods are applied
- For subtracting in the LNS domain, co-transformations have to be applied in the critical region λ [-1, 0]

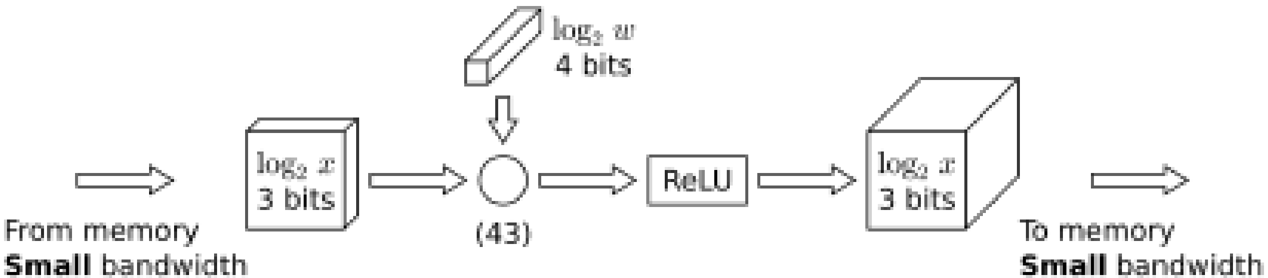
inesc id lisboa

European Logarithmic Microprocessor (ELM)

- 32-bit scalar microprocessor, Register-Memory ISA
  - 16 general-purpose registers, 8 kB L1data cache
  - two real adders/subtractors operating in 3 clock cycles
  - four combined multiplier/divider/sqrt/integer units operating in 1 clock cycle
  - vector operations use in parallel 4 functional units
- **Fixed-point LNS-based AU**
  - **Sign bit and 23 bits fractional component**
  - **Taylor interpolation for addition and subtraction**
- Fabricated with 0.18μm CMOS running at 125MHz, is evaluated against the TMS320C6711 contemporary DSP
  - addition marginally better **multiplications 3.4x faster**
  - **division and square root several times faster**

LNS: Convolution Neural Networks

- Application of LNS on CNNs allows activation and weights with only 3bits
  - with almost no loss in classification performance



$$conv = \sum_i 2^{\tilde{x}_i + \tilde{w}_i} = \sum_i BitShift(1, \tilde{x}_i + \tilde{w}_i)$$

- Accumulation can be done also in the log domain with the approximation

$$\log_2(1+x) \approx x \text{ for } 0 \leq x < 1$$

RNS

1. Logarithmic Number Systems (LNS)
2. Residue Number Systems (RNS)
3. Stochastic Computing (SC)
4. Hyper-Dimensional Computing (HDC)
5. DNA Computing
6. Quantum Computing
7. Applications
   A. Lattice-based Post-Quantum Cryptography
   B. Machine Learning
8. Conclusions

RNS

- RNS based on a set of relatively prime moduli: moduli set

$$P = \langle m_1, m_2, \cdots, m_N \rangle$$

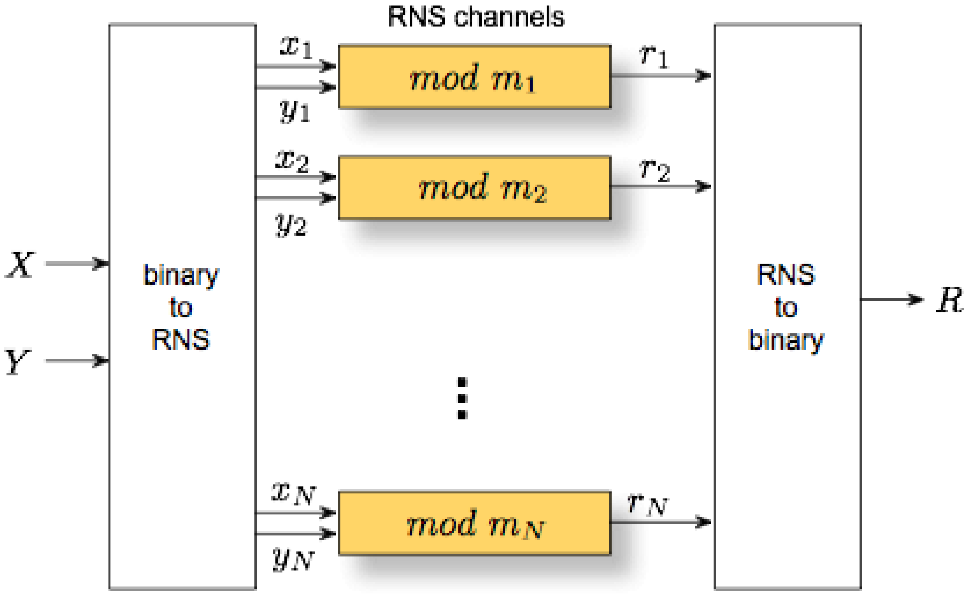
- The dynamic range M is given by:

$$M = m_1 \times m_2 \times \cdots \times m_N$$

- Integer *X* represented as:

$$X \longrightarrow \{x_1, x_2, \cdots, x_N\}$$

$$x_i = X \ mod \ m_i$$

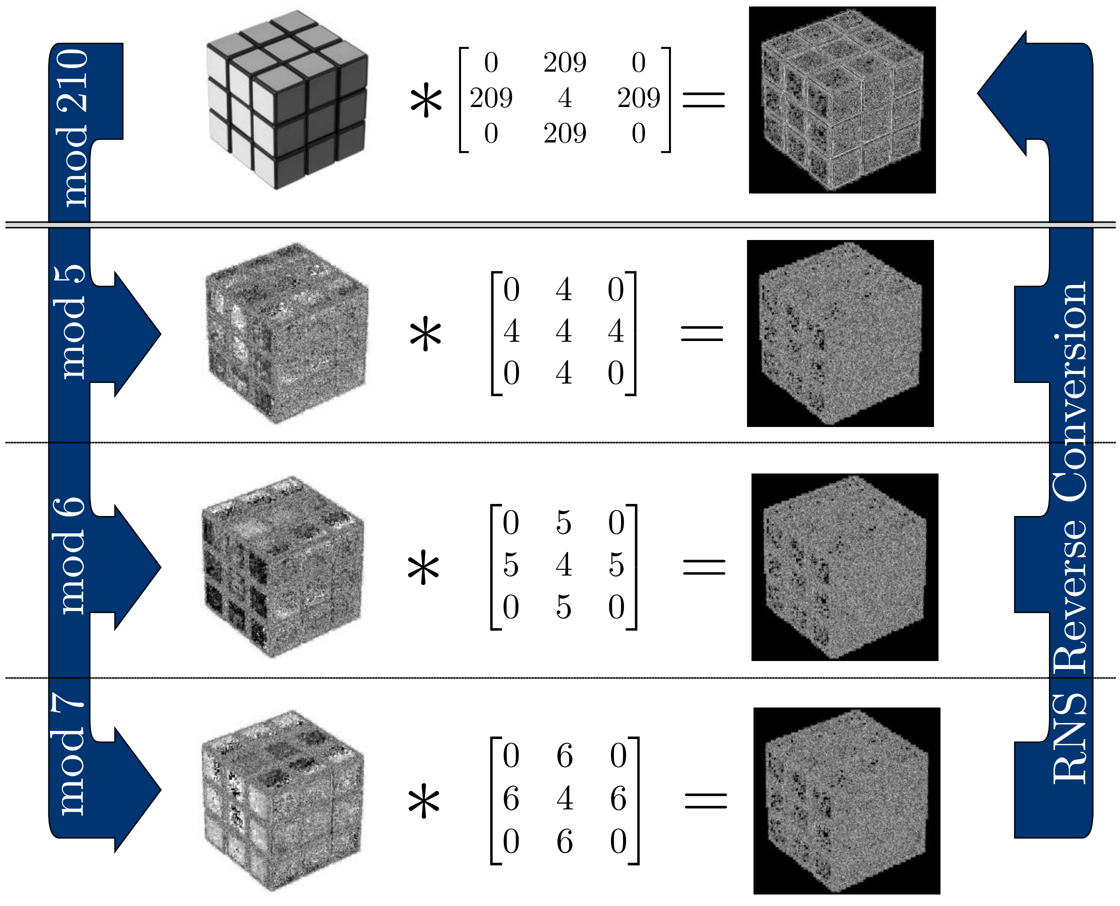


$$R = X \circ Y$$

Arithmetic operations (+,-,x,/):

$$\{r_1, r_2, \cdots r_N\} = \{(x_1 \circ y_1) \ mod \ m_1, (x_2 \circ y_2) \ mod \ m_2, \cdots, (x_N \circ y_N) \ mod \ m_N\}$$

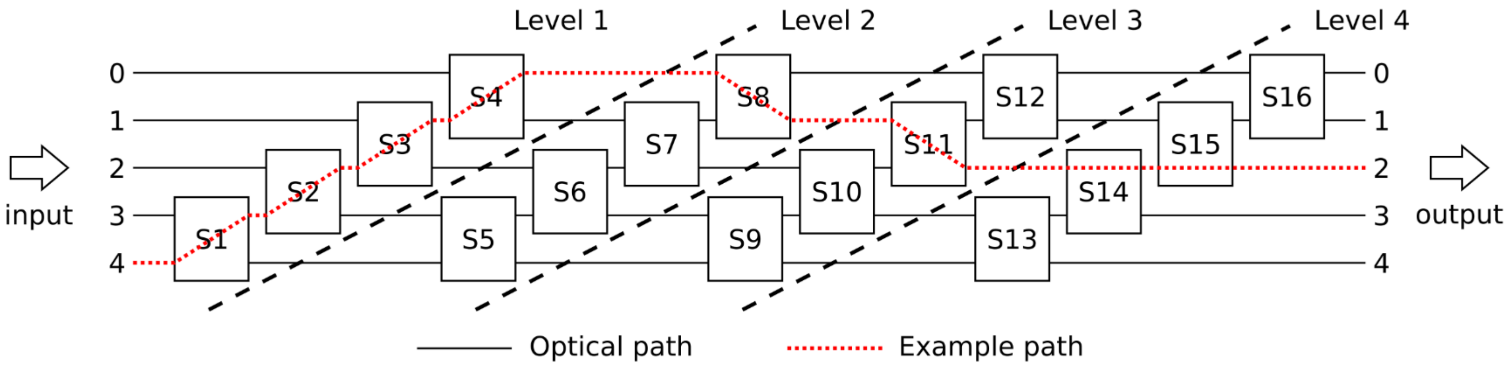
RNS


mod 210
mod 5
mod 6
mod 7
* [0 209 0; 209 4 209; 0 209 0] =
* [0 4 0; 4 4 4; 0 4 0] =
* [0 5 0; 5 4 5; 0 5 0] =
* [0 6 0; 6 4 6; 0 6 0] =
RNS Reverse Conversion

RNS: Photonics

- 2 x2 Hybrid Photonic-Plasmonic (HPP) integrated switches
  - fabricated by using Indium Tin Oxide as index modulation material
  - voltage signal controls guidance of light (may operate at 400 GHz), speed is defined by modulators, photodetectors and electronics

- RNS Parallelism (# switches grows with N^2) and energy efficiency of integrated phontonics → high-speed RNS units



RNS

- R(edundant)RNS is used for error detection/ correction
  - residues are independent, by introducing redundant moduli, the range of the legitimate moduli is extended to an illegitimate one

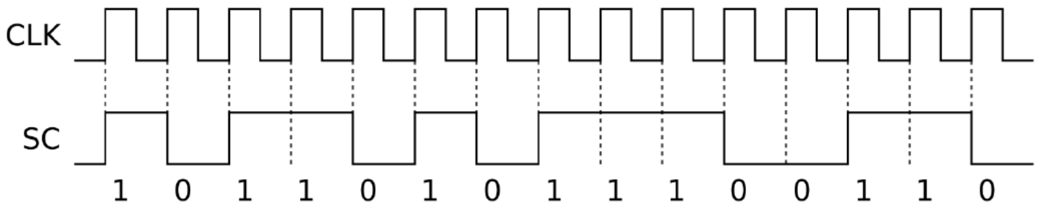
- The *Processing for Y'all* (CREEPY) [2018] core microarchitecture and ISA integrates RRNS centered algorithms and techniques to efficiently assure computational error correction.
  - significant improvements over a non-error correcting binary core
  - novel schemes proposed also for RNS based memory access, extend low power and energy efficient RRNS based architectures

SC

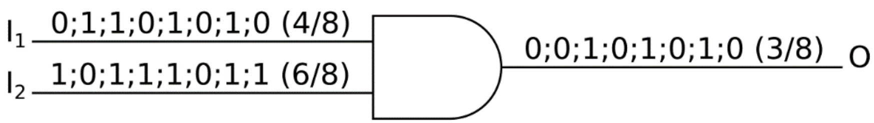
1. Logarithmic Number Systems (LNS)
2. Residue Number Systems (RNS)
3. Stochastic Computing (SC)
4. Hyper-Dimensional Computing (HDC)
5. DNA Computing
6. Quantum Computing
7. Applications
   A. Lattice-based Post-Quantum Cryptography
   B. Machine Learning
8. Conclusions

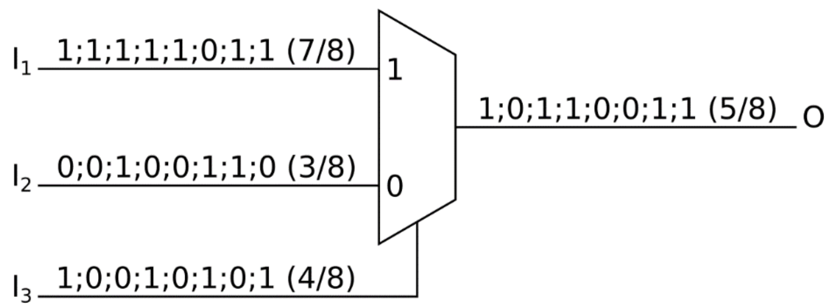
SC

- From a continuous-time stochastic process, the value of a bitstream is the #‘1’ bits over the total #bits (9/15=0.6)



- Multiplication addition

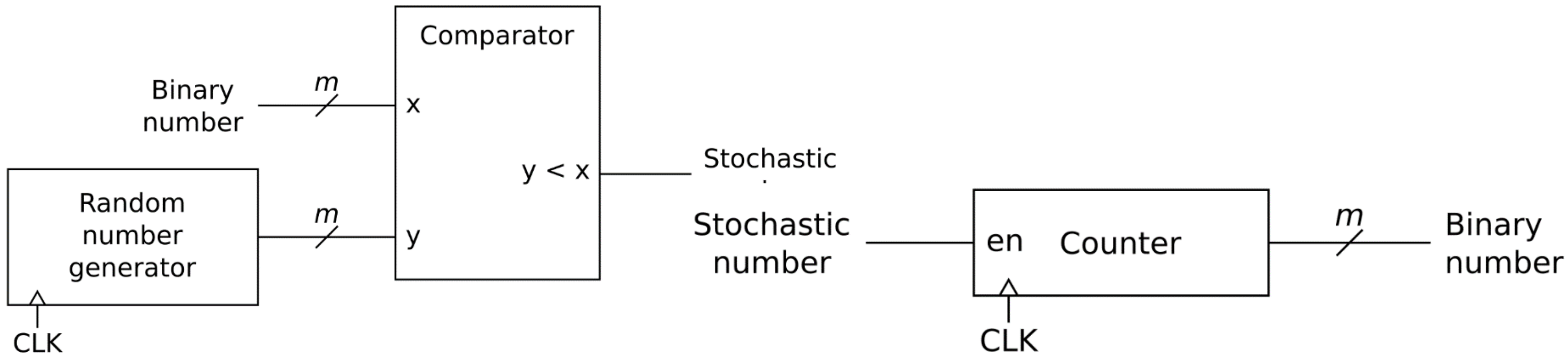




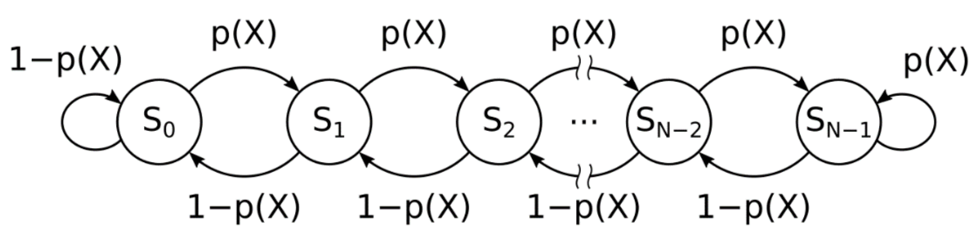
- Correctness impacted by correlation between bitstreams
  - e.g. the same stream at the 2 inputs of the * produces the same stream at the output, instead of the square

SC

- Converters SC->Binary Binary->SC



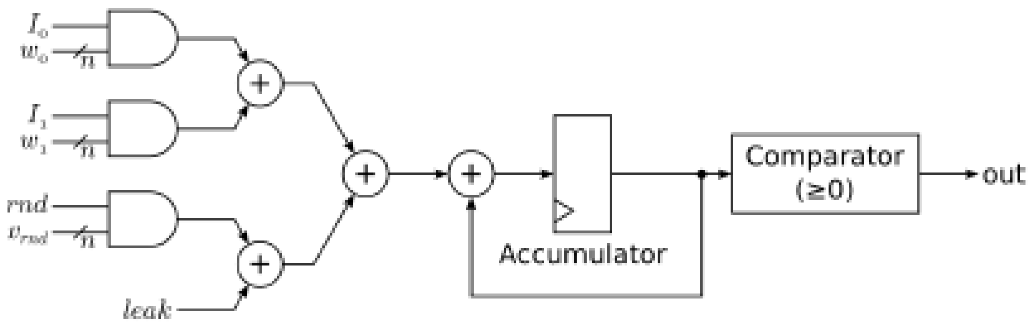
- Highly non-linear functions (e.g. tanh and max functions in ANNs) require FSM-based SC elements



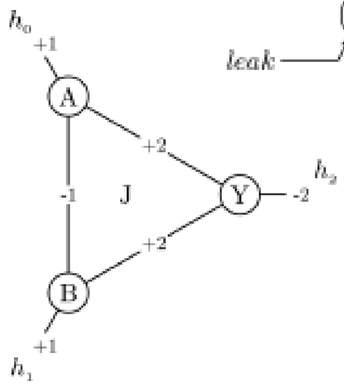
SC-based CMOS Invertible Logic

- Boltzman Machine

$$m_i(t+\Delta t) = sgn(rnd(-1,+1) + \tanh(I_i(t+\Delta t)))$$
$$I_i(t+\Delta t) = I_0(h_i + \sum J_{ij} m_j(t)) \quad (36)$$



AND gate

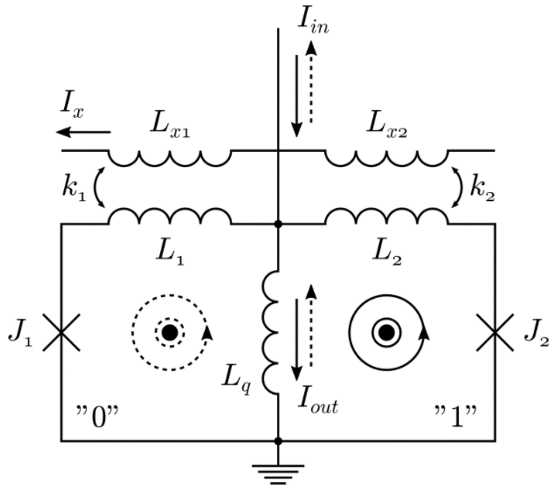


$$h = \begin{bmatrix} +1 & +1 & -2 \end{bmatrix}$$
$$J = \begin{bmatrix} 0 & -1 & +2 \\ -1 & 0 & +2 \\ +2 & +2 & 0 \end{bmatrix}$$

- 5 by 5-bit */divider/factorizer 13x less area than binary for the TSMC 65nm technology

SC: Superconducting Quantum Device

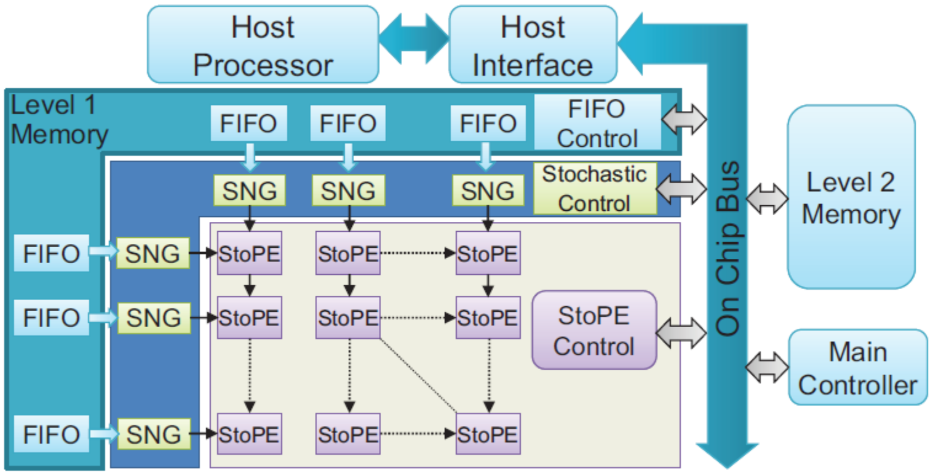
- Adiabatic Quantum-Flux-Parametron logic
  - energy efficiency: ALU RISC V 10x lower energy than CMOS 12nm



- Two characteristics AQFP suitable to implement SC
  - deep pipelining: gate is connected with AC clock signal requiring a clock phase, difficult to avoid RAW hazards with binary computing
  - The opportunity of true RNG using simple buffers

SC: Processor

- Stochastic Recognition and Mining (StoRM) Processor



- 2D array of Stochastic PE (typically 15x15)
- Binary-to-stochastic units shared across rows/columns
- Implementation on TSMC 65nm: one order of magnitude less circuit area and power consumption

Hyper-Dimensional Computing (HDC)

1. Logarithmic Number Systems (LNS)
2. Residue Number Systems (RNS)
3. Stochastic Computing (SC)
4. Hyper-Dimensional Computing (HDC)
5. DNA Computing
6. Quantum Computing
7. Applications
   A. Lattice-based Post-Quantum Cryptography
   B. Machine Learning
8. Conclusions

HDC

- HDC inspired in brain-like operation
  - supported on random high-dimensional vectors, in the order of thousands of bits (10,000-bit vector)
  - alternative to SVM and CNN for supervised classification
  - Associate Memories (AM): pattern X is stored using pattern A as the address, latter X can be retrieved from A or A’ similar to A

- The high number of bits does not improve resolution
  - tolerant to errors and component failure, many patterns equivalent
  - highly structured information, like in the brain, deals with arbitrariness of the neural code

HDC: Arithmetic

- Componentwise **Addition** of a set: sum represents a set individual vectors
- **Multiplication implements** with bitwise logic XNOR
  - bipolar representation, (0, 1) -> (1, -1), X*Y=X *xnor* Y
  - Multiplication maps points: X*M maps X into X_M that is as far from X as the number of 1s in M;

    M a random vector => multiplication randomizes X

$$d(X_M, X) = \| X_M * X \| = \| M * X * X \| = \| M \|$$

  - multiplication is distributive over addition, and implements a mapping that preserves distance

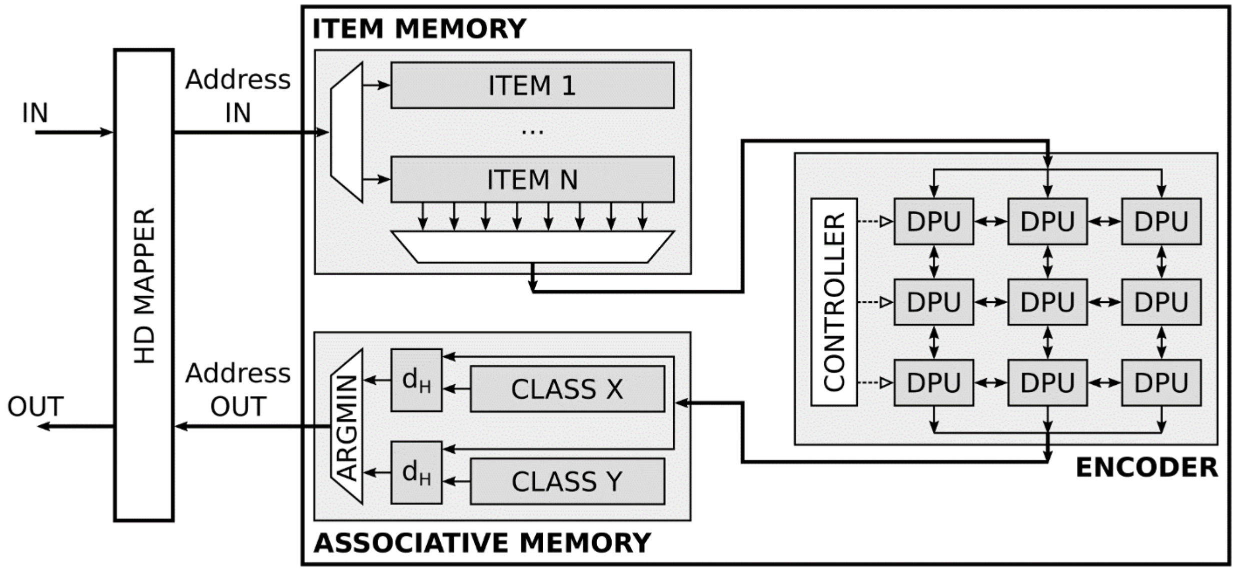
$$d(X_M, Y_M) = d(X, Y)$$

HDC: Nanosystem

- End-to-end brain-inspired HDC nanosystem, using heterogeneous integration of multiple emerging nanotechnologies

  - Monolithic 3D integration of Carbon, Nanotube Field-Effect Transistors (CNFETs) and Resistive Random-Access Memory (RRAM)
  - fine-grained and dense vertical connections between computation and storage layers
  - Integrating RRAM and CNFETs allows to create area-and energy-efficient circuits

HDC: Processor



- IM stores a large collection of random hyper-vectors (items)
  - maps symbols to items in the inference phase as trained
- DPUs combine hyper-vectors sequence according to the algorithm
  - to compose a single hyper-vector per each class.
- AM stores the trained class hyper-vectors
  - deliver the best prediction according to the Hamming distance (d_h).

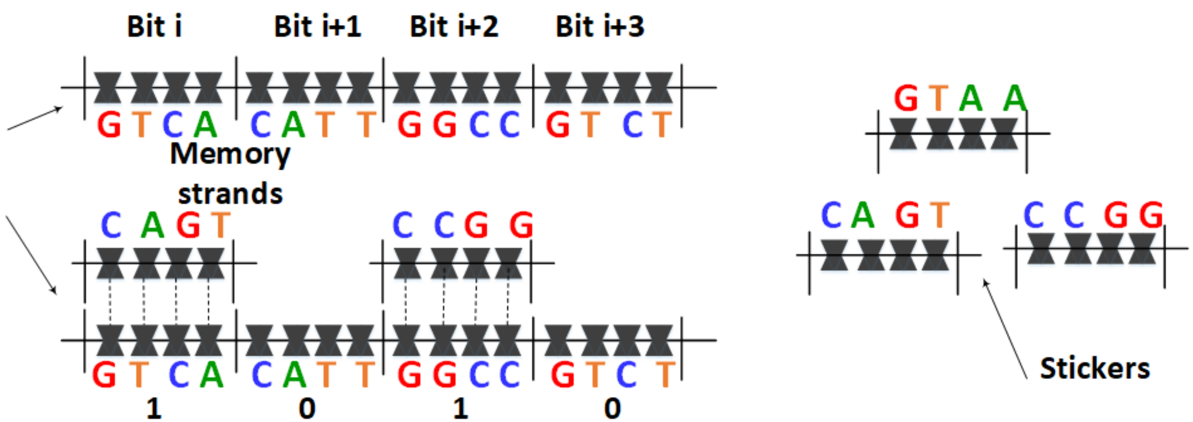
DNA-based Computing

1. Logarithmic Number Systems (LNS)
2. Residue Number Systems (RNS)
3. Stochastic Computing (SC)
4. Hyper-Dimensional Computing (HDC)
5. **DNA Computing**
6. Quantum Computing
7. Applications
   A. Lattice-based Post-Quantum Cryptography
   B. Machine Learning
8. Conclusions

inesc id lisboa

DNA-based Computing

- With the DNA sticker model, a binary number represented through two groups of single-stranded DNA molecules
  - the memory strand, a long DNA molecule subdivided into non-overlapping segments
  - set of stickers, short DNA molecules, each with the length of a segment, a sticker is complementary to one of those segments



DNA-based Computing

- Example of the bitwise AND operation of 2 n-bit vectors

Algorithm 1 AND(T_{s1}, T_{s2}, n:in; T_d:out)

Require: Pour blank strand of n bits $(0 \ldots 0)$ in T_d
Ensure: bit_stream_in_T_d =bit_stream_in_T_{s1} $\wedge$ bit_stream_in_T_{s2}

Combine(T_a, T_{s1}, T_{s2}) {T_a: auxiliary Tube}
for all bit $0 \leq i < n$ **do**
  Separate($T_a, i, B_{[1]}, B_{[0]}$)
  if $B_{[0]}$ is empty **then**
    Set(T_d, i)
  end if
  Combine($T_a, B_{[1]}, B_{[0]}$)
end for

- DNA ALU was constructed:
  - with 1-bit FA, AND, OR and NAND, decoding and controlling logic

RRNS DNA-based Computing

- RRNS has been applied for overcoming the negative effects caused by the defects and instability of the biochemical reactions and errors in hybridizations
  - applying the RRNS 3-moduli set $\{2^{n-1}, 2^{n+1}, 2^{n+1}\}$ to the DNA model leads to one-digit error detection
  - the parallel RRNS-based DNA arithmetic improves the reliability of DNA computing while at the same time simplifies the DNA encoding scheme

1. Logarithmic Number Systems (LNS)
2. Residue Number Systems (RNS)
3. Stochastic Computing (SC)
4. Hyper-Dimensional Computing (HDC)
5. DNA Computing
6. **Quantum Computing**
7. Applications
   A. Lattice-based Post-Quantum Cryptography
   B. Machine Learning
8. Conclusions

Quantum Computing

- A quantum bit (*qubit*), a microscopy unit, such as an atom or a nuclear spin, is a superposition of orthogonal basis states, |0> and |1>

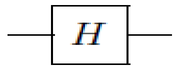
$$|x\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \ ; \ |\alpha|^2 + |\beta|^2 = 1$$

- Generalizing, the state of an *n-qubit* system

$$\Upsilon = \sum_{b=\in 0,1^n} c_b\,|b\rangle \ ; \sum_b |c_b|^2 = 1$$

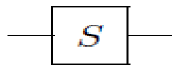
Quantum Computing

- Single *qubit* gates and respective unitary matrices



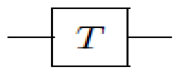
$$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

(a) Hadamard gate



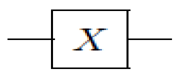
$$\begin{bmatrix}1 & 0\\ 0 & j\end{bmatrix}$$

(b) Phase gate



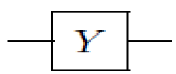
$$\begin{bmatrix}1 & 0\\ 0 & \exp^{j\pi/4}\end{bmatrix}$$

(c) $\pi/8$ gate



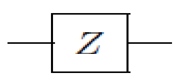
$$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$$

(d) Pauli-X gate



$$\begin{bmatrix}0 & -j\\ j & 0\end{bmatrix}$$

(e) Pauli-Y gate

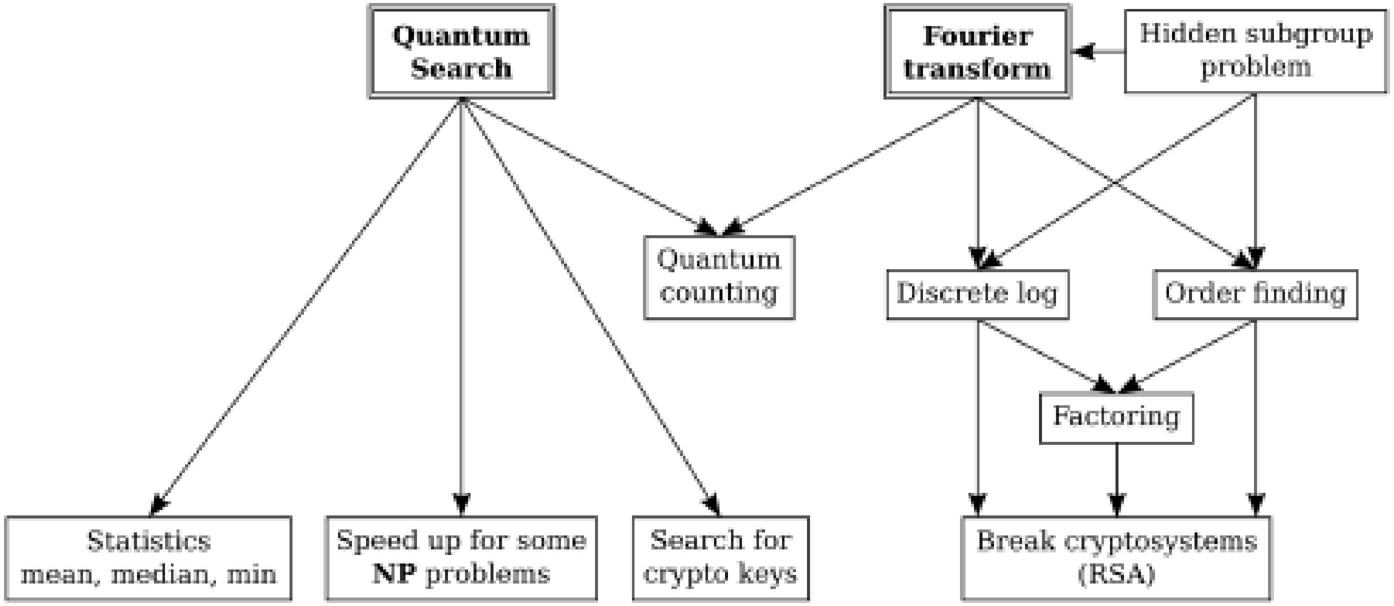


$$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$$

(f) Pauli-Z gate

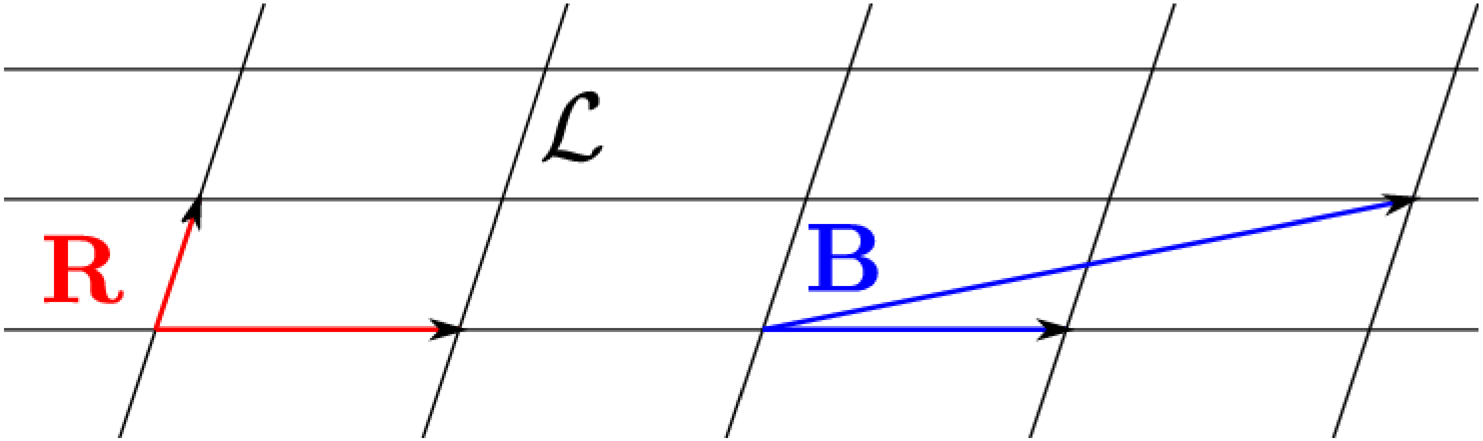
Quantum Computing

- Quantum algorithms



1. Logarithmic Number Systems (LNS)
2. Residue Number Systems (RNS)
3. Stochastic Computing (SC)
4. Hyper-Dimensional Computing (HDC)
5. DNA Computing
6. Quantum Computing
7. Applications
   A. Lattice-based Post-Quantum Criptography
   B. Machine Learning
8. Conclusions

Lattice-based Cryptography

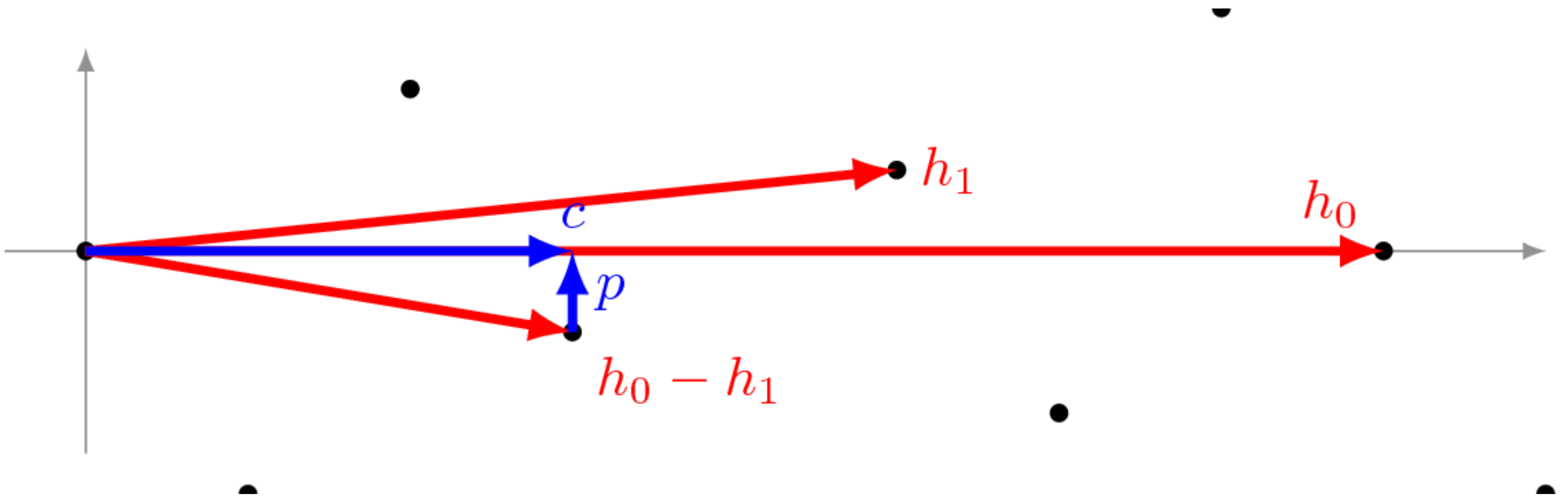


- Matrix $R = (r_1, \ldots, r_l)^T$: a basis of $\mathcal{L}$
  - $\mathcal{L} = r_1\mathbb{Z} \oplus \ldots \oplus r_l\mathbb{Z}$

- For $n \geq 2$, there are infinite basis

Lattice-based Cryptography

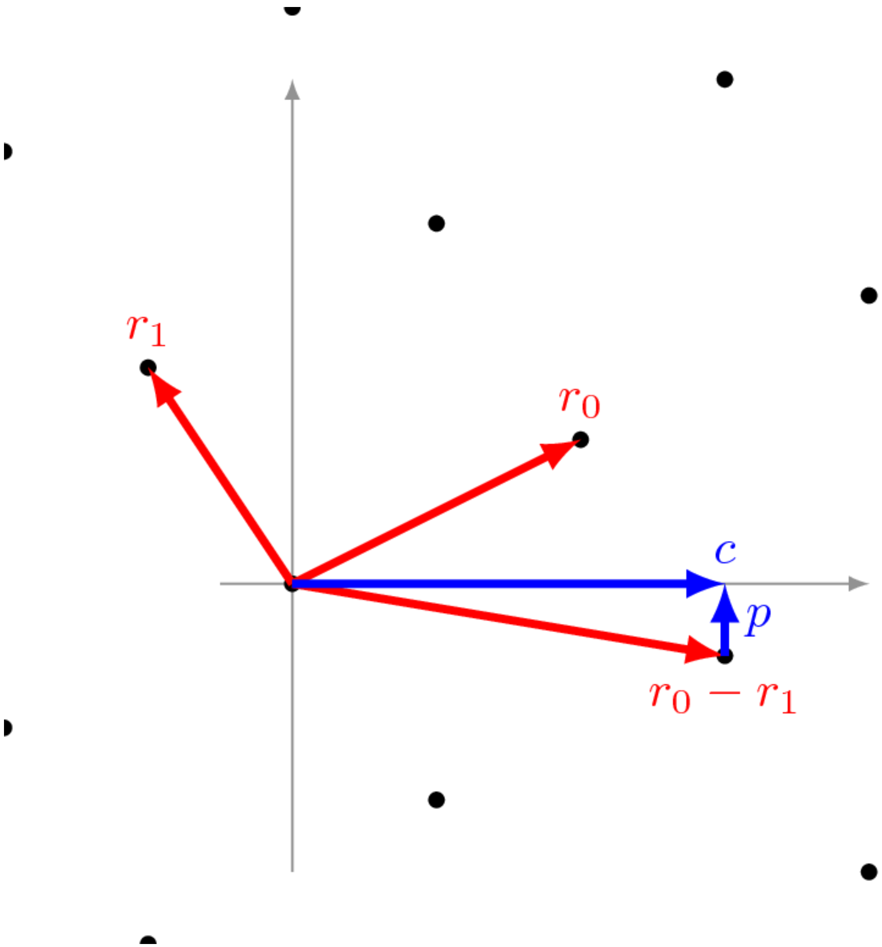
- Encryption corresponds to adding a perturbation p to a lattice point
- (h_0, h_1) is a "bad" lattice base



Lattice-based Cryptography

- Decryption corresponds to finding the closest lattice vector u to c and outputting $p = c - u$

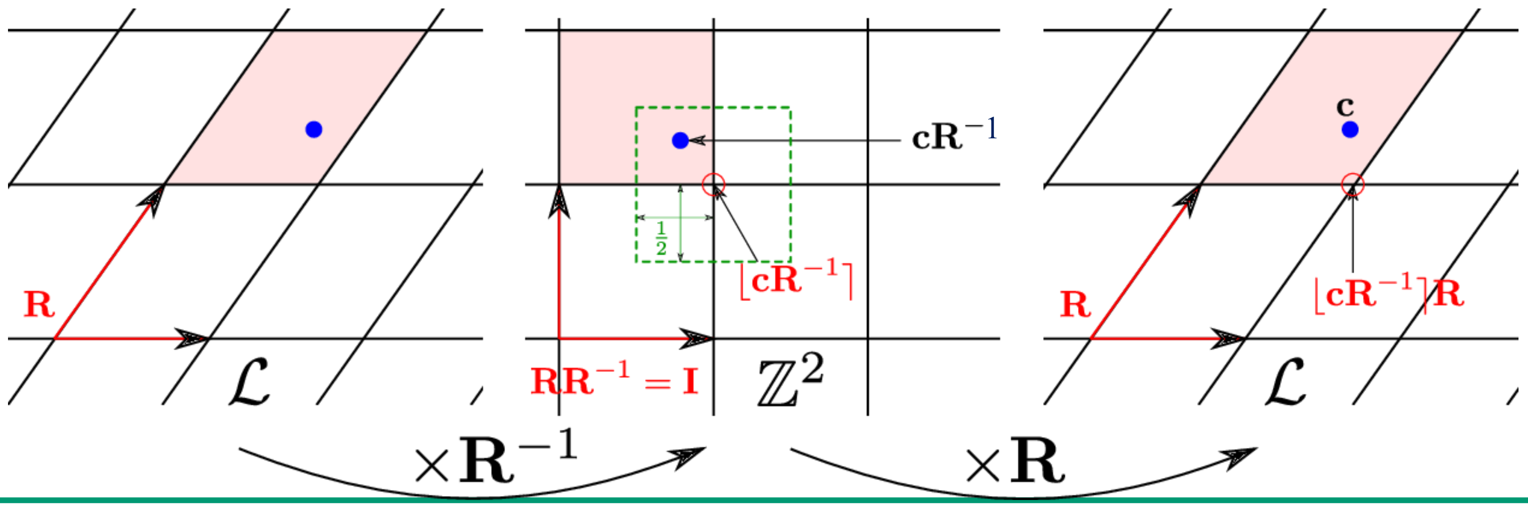
- (r_0, r_1) is a “good” lattice base



Lattice-based Cryptography

Babai's Round-off Algorithm

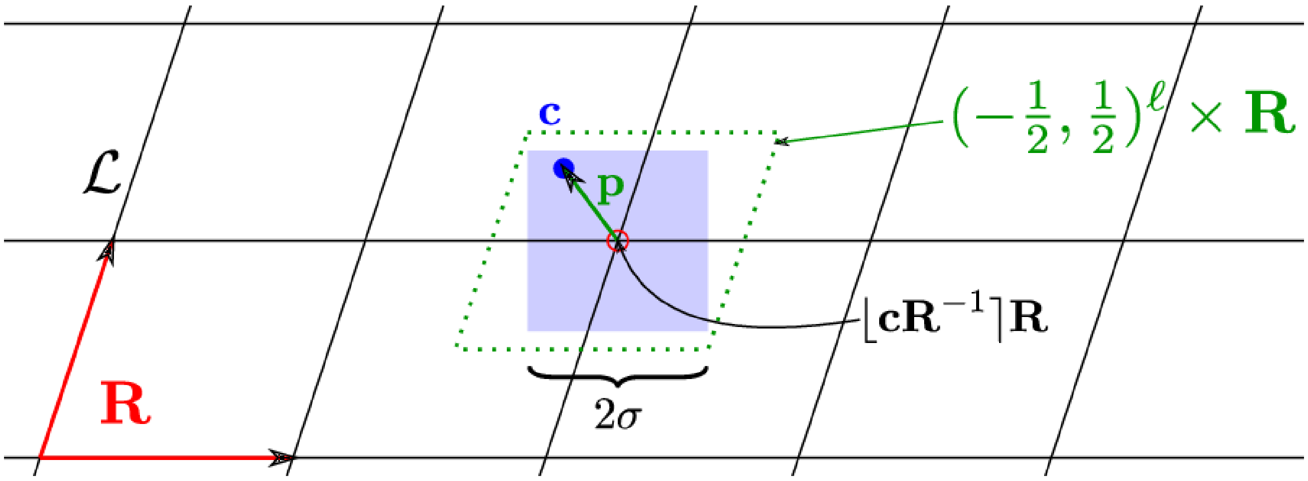
change of basis	rounding components	back to canonical basis
$c \times R^{-1}$	$\lfloor c \times R^{-1} \rceil$	$\lfloor c \times R^{-1} \rceil \times R$



Lattice-based Cryptography

Common Simplification Step

- Use special case of CVP: Bounded Distance Decoding Problem (BDD)
- Babai's Round-off gives the closest vector for a rotated nearly-orthogonal basis *R* of a lattice



$$p = c - \lfloor cR^{-1} \rceil R \text{ modc } m_\sigma \text{ for } m_\sigma \geq 2\sigma + 1$$

Lattice-based Cryptography

- Babai's algorithm rewritten with integer arithmetic:

  - $u = \lfloor cR^{-1} \rceil R = \left\lfloor cR^{-1} + \frac{1}{2} \right\rfloor R = \left\lfloor \frac{dcR^{-1}}{d} + \frac{1}{2} \right\rfloor R =$

$$\frac{2cdR^{-1} + d - \boxed{(2cdR^{-1} + d \bmod (2d))}}{2d} R$$

where $d = \det(R)$

Use RNS Montgomery's reduction

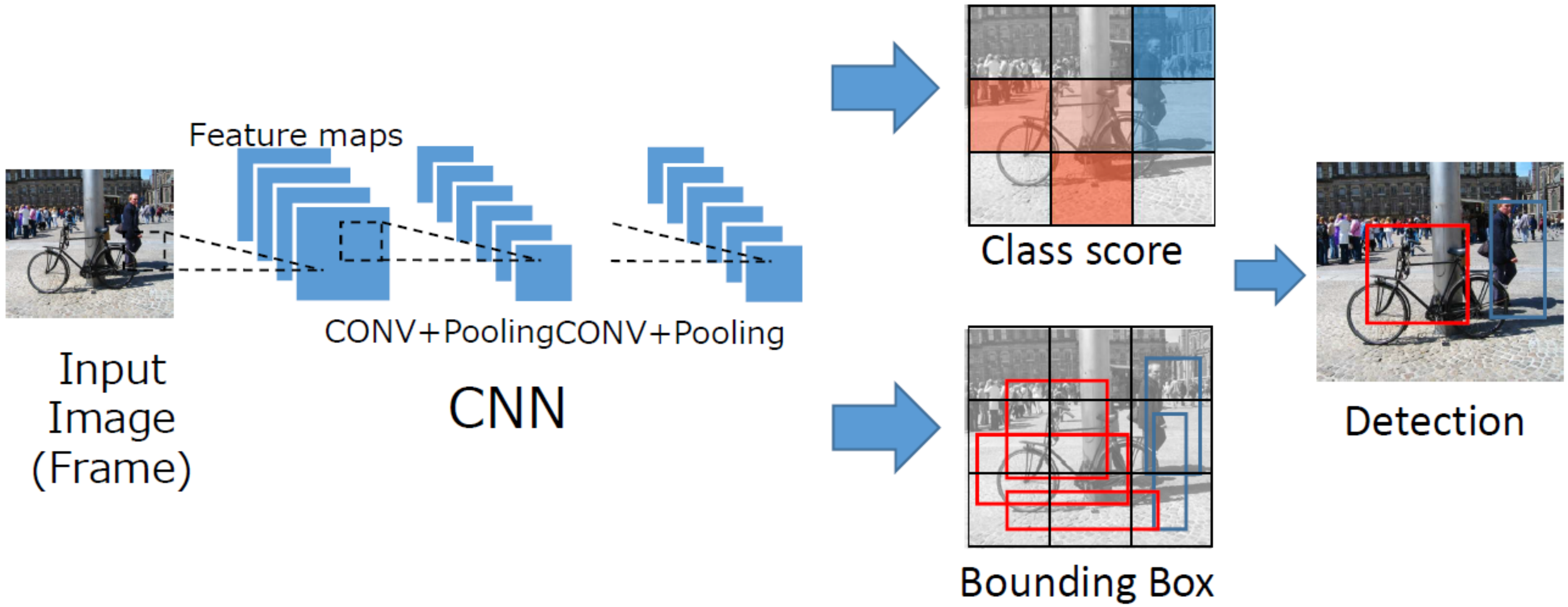
RNS based LBC decryption

Results for LBC decryption in CPUs/GPUs

Execution Times [$\times 10^6$ clock cycles] (Speed-up)				
Method	$n = 400$	$n = 600$	$n = 800$	$n = 1000$
Sequential (i7 4770K)	97.51	283.8	619.4	1222
RNS-GPU (K40c)	22.97 (4.2)	283.8 (3.6)	248.9 (2.5)	512.4 (2.4)
RNS-GPU (GTX 780 Ti)	16.55 (5.9)	59.73 (4.8)	148.2 (4.2)	349.6 (3.5)
4-core RNS-CPU (i7 4770K)	21.05 (4.6)	75.48 (3.8)	189.9 (3.3)	369.7 (3.3)
4-core RNS-CPU (with AVX2) (i7 4770K)	8.668 (11.2)	29.05 (9.8)	74.79 (8.3)	148.5 (8.2)

YOLOv2
(You Only Look Once version 2)

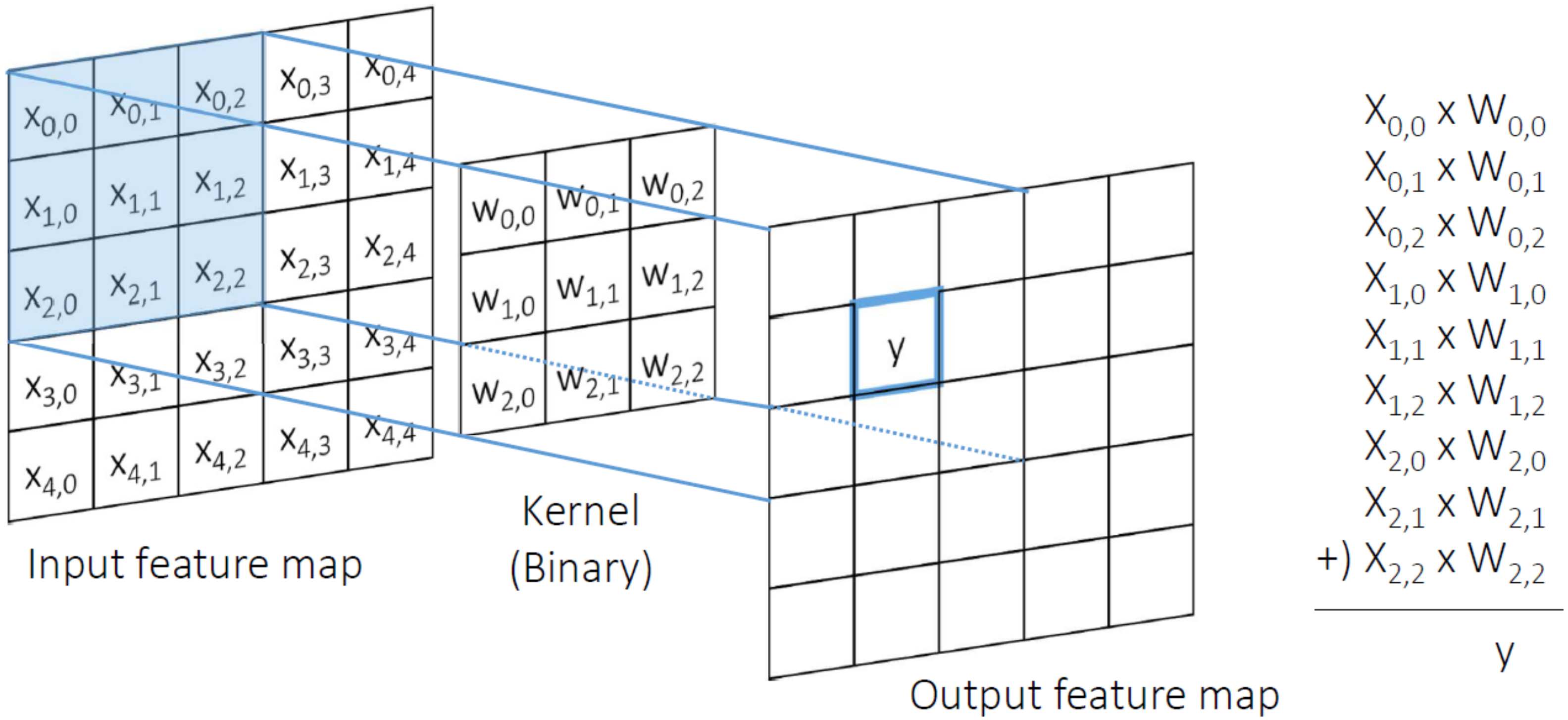
- Single CNN (One-shot) object detector
  - Both a classification and a BBox estimation for each grid



J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *arXiv preprint arXiv:1612.08242*, 2016.

2D Convolutional Operation

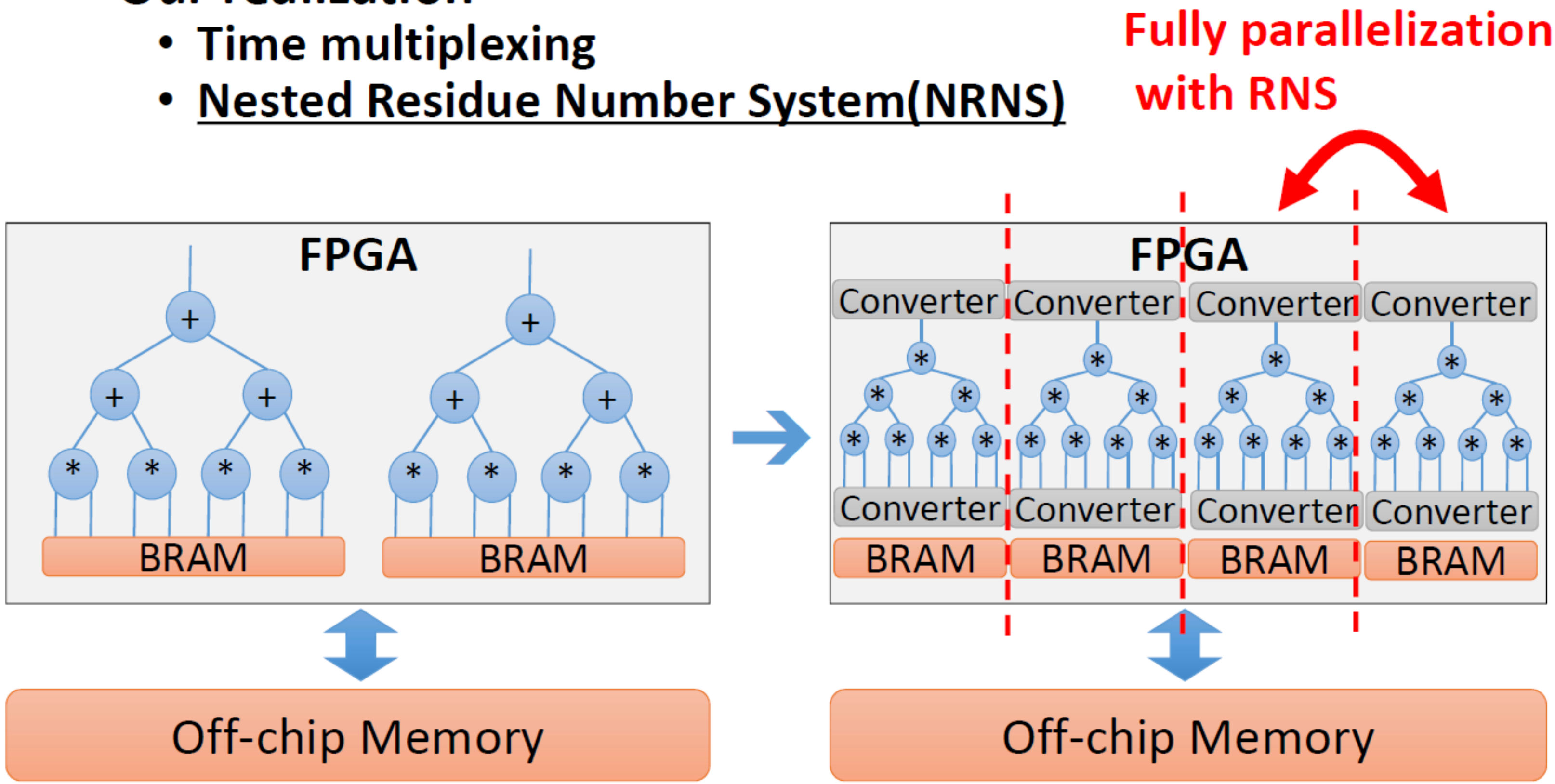
- **Computational intensive part of the YOLOv2**



ML: CNNs

Realization of 2D Convolutional Layer

- Requires more than billion MACs
- Our realization
  - Time multiplexing
  - <u>Nested Residue Number System(NRNS)</u>



ML: Nested RNS

Nested RNS

- $(Z_1, Z_2, \ldots, Z_i, \ldots, Z_L) \rightarrow (Z_1, Z_2, \ldots, (Z_{i1}, Z_{i2}, \ldots, Z_{ij}), \ldots, Z_L)$
- Ex: $\langle 7, \underline{11}, \underline{13} \rangle \times \langle 7, 11, 13 \rangle$

Original modulus

$$\langle 7, \langle 5,6,7 \rangle_{11}, \langle 5,6,7 \rangle_{13} \rangle \times \langle 7, \langle 5,6,7 \rangle_{11}, \langle 5,6,7 \rangle_{13} \rangle$$

1. Reuse the same moduli set
2. Decompose a large modulo into smaller ones

Example of Nested RNS

- $19 \times 22 (= 418)$ on $<7, <5,6,7>_{11}, <5,6,7>_{13}>$

19×22

$= <5,8,6> \times <1,0,9>$

Binary2NRNS Conversion

$= <5, <3,2,1>_{11}, <1,0,6>_{13}> \times <1, <0,0,0>_{11}, <4,3,2>_{13}>$

Modulo Multiplication

$= <5, <0,0,0>_{11}, <4,0,5>_{13}>$

Bin2RNS on NRNS

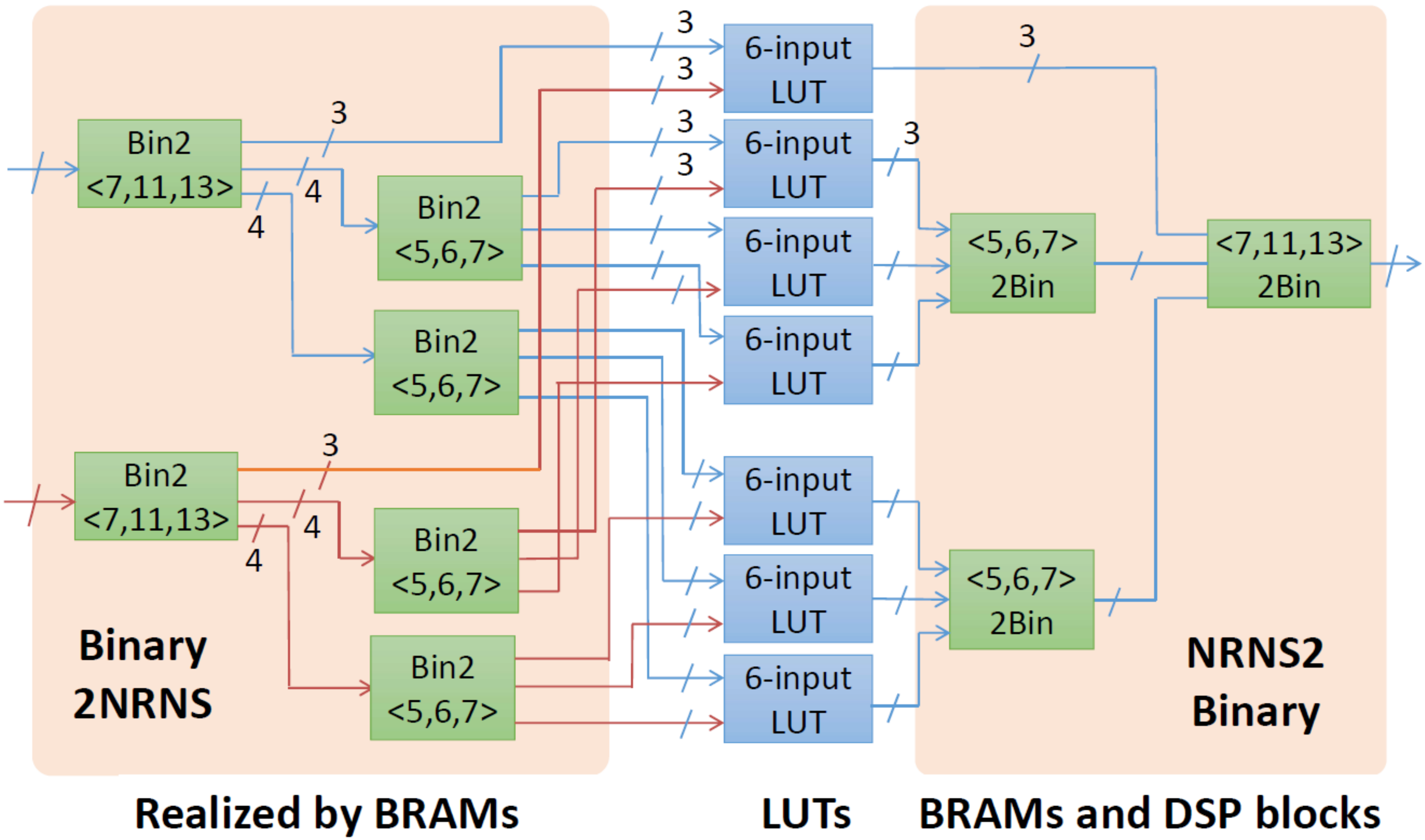
$= <5,0,2>$

RNS2Bin

$= 418$

ML: Nested RNS

Realization of Nested RNS



ML: NRNS based YOLOv2

NRNS based YOLOv2

- Framework: Chainer 1.24.0
- CNN: Tiny YOLOv2
- Benchmark: KITTI vision benchmark
- mAP: 69.1 %

Layer	# In. Fmaps	# Out. F Size
(Feature Extraction)		
Conv1	3	128×128
Conv2	128	128×128
Max Pool	128	64×64
Conv3	128	64×64
Conv4	128	64×64
Conv5	128	64×64
Max Pool	128	32×32
Conv6	128	32×32
Conv7	128	32×32
Conv8	128	32×32
Max Pool	128	16×16
(Localization+Classification)		
Conv9	128	16×16
Conv10	128	16×16
Conv11	128	$5^2 \times 3 + (5 \times 5)$
Accuracy (mAP)	69.1	

ML:Implementation

Implementation

- FPGA board: NetFPGA-SUME
  - FPGA: Virtex7 VC690T
  - LUT: 427,014 / 433,200
  - 18Kb BRAM: 1,235 / 2,940
  - DSP48E: 0 / 3,600
- Realized the pre-trained NRNS-based YOLOv2
  - 9 bit fixed precision (dynamic range: 30 bit)
- Synthesis tool: Xilinx Vivado2017.2
  - Timing constrain: 300MHz
  - 3.84 FPS@3.5W → 1.097 FPS/W

ML: Evaluation

Comparison

	NVivia Pascal GTX1080Ti	NetFPGA-SUME
Speed [FPS]	20.64	3.84
Power [W]	60.0	3.5
Efficiency [FPS/W]	0.344	1.097

inesc id lisboa

Conclusions

- Unconventional data representation and arithmetic fundamental for computing on emerging technologies, such as
  - **RNS**: DNA computing; **SC**: quantum devices (AQFP); **HDC**: CNFET,RRAM

- New applications using unconventional arithmetic, namely
  - **LNS**: ML/CNN; **RNS**: Post-Quantum cryptography; SC: homomorphic encryption

- For the investigation on non-conventional arithmetic all dimensions of the systems should be considered
  - including not only computer arithmetic theory, but also advances in technology and the demands of emergent applications.

Thank You
for your attention!

technology
from seed

inesc id lisboa 10 anos

Instituto de Engenharia de Sistemas e Computadores
Investigação e Desenvolvimento em Lisboa