Saint-Malo, September 13th, 2015
Cryptographic Hardware and Embedded Systems

Highly Efficient $GF(2^8)$ Inversion Circuit Based on Redundant GF Arithmetic and Its Application to AES Design

Rei Ueno[1], Naofumi Homma[1], Yukihiro Sugawara[1],
Yasuyuki Nogami[2], and Takafumi Aoki[1]

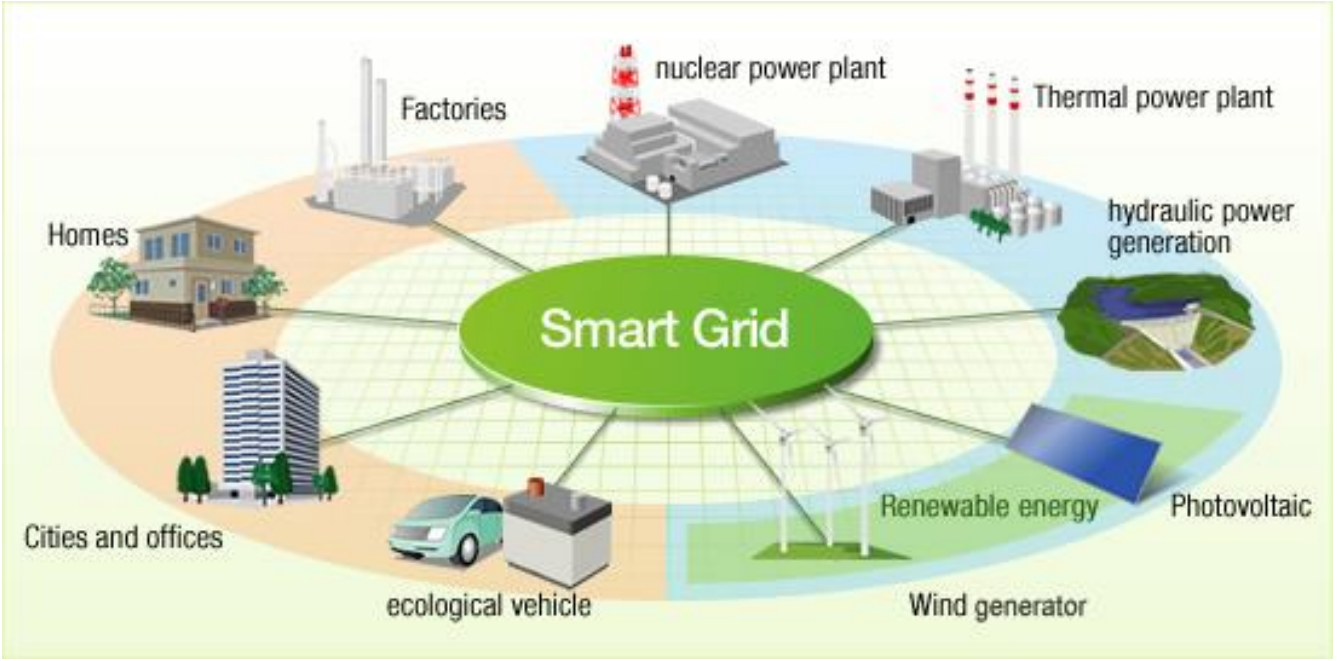
Joint work with
[1] Tohoku University and [2] Okayama University

Outline

- Introduction
- Redundant GF arithmetic
- $GF(2^8)$ inversion circuit
- AES encryption S-Box
- Concluding remarks

Background

- Demands for compact and efficient crypto. HW
  - Applications to resource-limited devices in IoT



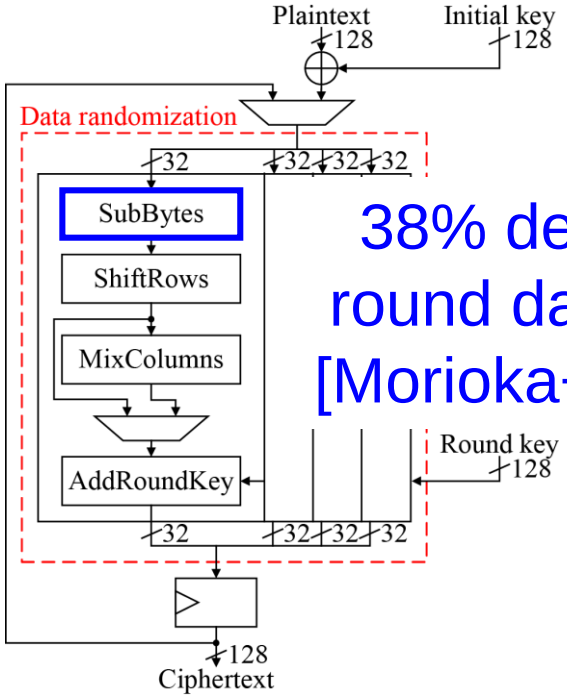
www.hitachi.com

- Light-weight AES implementation
  - Connectivity of existing systems and protocols
  - Influence on other ciphers (e.g., Camellia, SNOW 3G)

AES processors

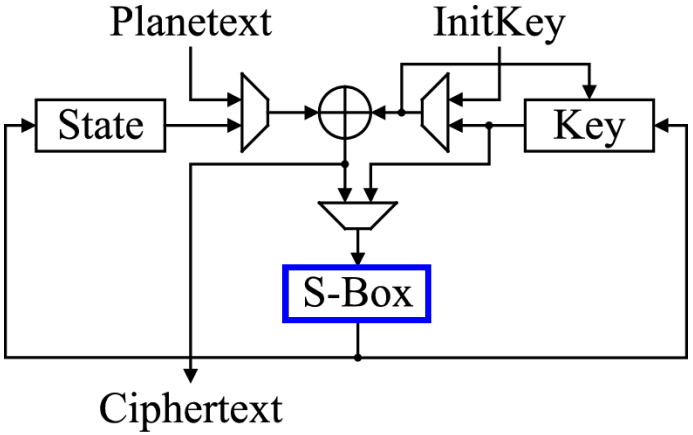
- $GF(2^8)$ inversion is critical in AES processors
  - Major part of SubBytes

Round-based architecture



38% delay of round datapath [Morioka+ 2004]

Byte-serial architecture

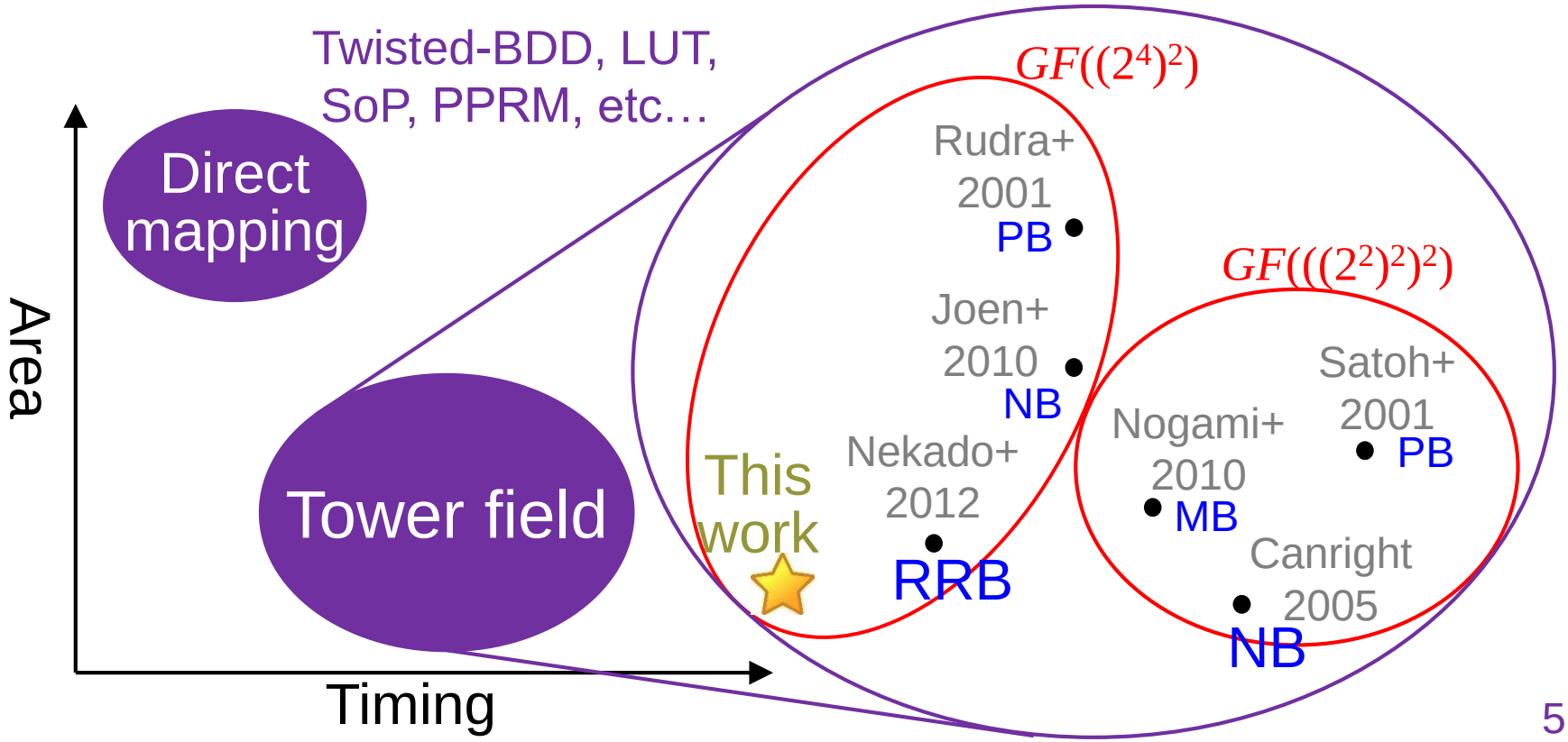


28% area of combinational block [Moradi+ 2011]

Compact and efficient $GF(2^8)$ inversion circuit is desirable

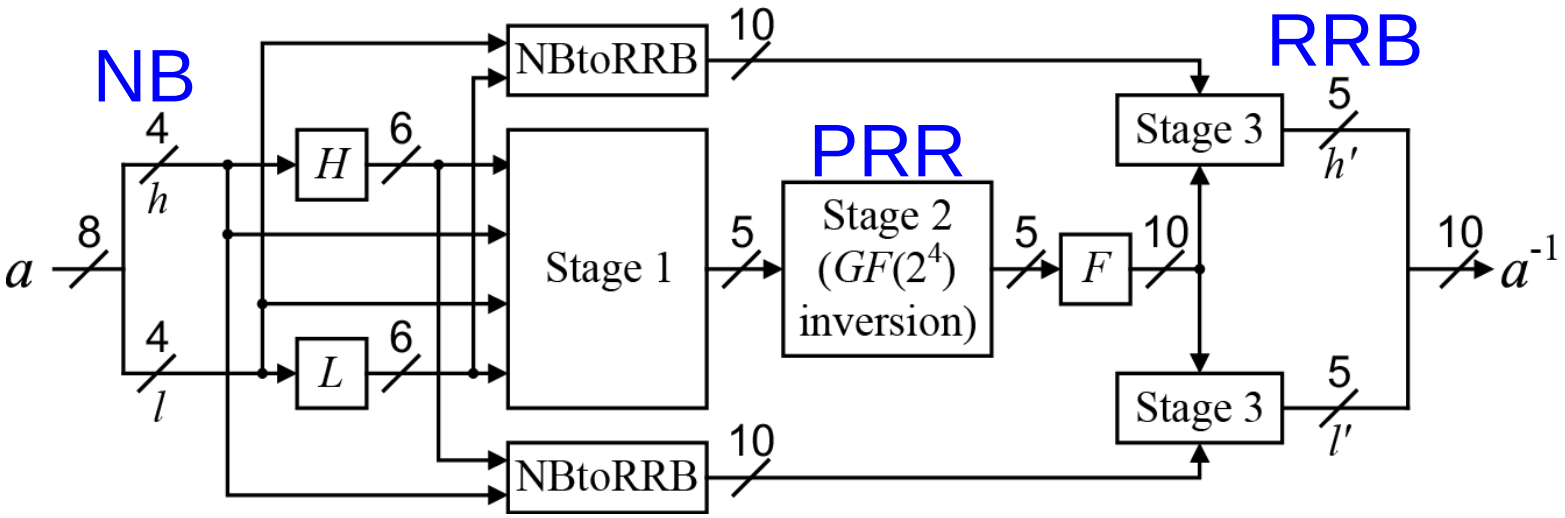
Design of $GF(2^8)$ inversion circuit

- Arithmetic approach for AES S-box design
  - Field towering and GF representation make a difference
    - Tower field: $GF(((2^2)^2)^2)$, $GF((2^4)^2)$
    - GF representation: PB, NB, MB, RRB…



Key trick

- Combination of three GF representations
  - One non-redundant representation: Normal Basis (NB)
  - Two redundant representations:
    - Polynomial Ring Representation (PRR)
    - Redundantly Represented Basis (RRB)



Proposed circuit architecture

Results

- Highly efficient $GF(2^8)$ inversion circuit
  - Redundant GF arithmetic makes difference
  - 38% faster than the conventional smallest one w/o area overhead
- Application to AES encryption S-box
  - Isomorphic mappings optimized for efficiency
  - 17% more efficient than state-of-the-art S-boxes

Synthesis result of $GF(2^8)$ inversion circuits with TSMC 65 nm

	Field	Area [GE]	Timing [ns]	AT product
[Canright 2005]	$GF(((2^2)^2)^2)$	237.33	2.92	693.00
[Nekado 2012]	$GF((2^4)^2)$	272.67	1.89	515.35
This work	$GF((2^4)^2)$	229.67	1.81	415.70

Outline

- Introduction
- Redundant GF arithmetic
- $GF(2^8)$ inversion circuit
- AES encryption S-box
- Concluding remarks

What's redundant GF arithmetic?

- Represent $GF(2^m)$ element by n bits ($n > m$)
  - Modular polynomial: n-th degree reducible polynomial
- Polynomial Ring Representation (PRR)
  - Equal to Cyclic Redundancy Code (CRC)
    - Don't-care inputs (explained by code theory)
    - Efficient for non-linear operations e.g., inversion
- Redundantly Represented Basis (RRB)
  - Linear combination of linear dependent elements of $GF(2^m)$
    - Each element is NOT represented uniquely
    - Efficient for multiplication

Why redundant GF arithmetic?

- Modular polynomial determines performance of GF arithmetic circuit
  - Binomial $x^n + 1$ is optimal but reducible
  - Redundant GF can exploit binomial
    - $x^5 + 1$ is available for redundant $GF(2^4)$

Critical factors of GF arithmetic algorithm

Rep.	Modular polynomial	Squaring	Multiplication	Inversion
PB	Irreducible	XOR-gate array	Mastrovito	ITA
NB	Irreducible	Bit-wise permutation	Massey-Omura	ITA
PRR	Binomial	Bit-wise permutation	CVMA	Mapping
RRB	Binomial	Bit-wise permutation	Reduced CVMA	ITA

Why redundant GF arithmetic?

- Modular polynomial determines performance of GF arithmetic circuit
  - Binomial $x^n + 1$ is optimal but reducible
  - Redundant GF can exploit binomial
    - $x^5 + 1$ is available for redundant $GF(2^4)$

Critical factors of GF arithmetic algorithm

Rep.	Modular polynomial	Squaring	Multiplication	Inversion
PB	Irreducible	Bad	OK	OK
NB	Irreducible	Good	Bad	OK
PRR	Binomial	Good	Good	Good
RRB	Binomial	Good	Very good	OK

Outline

- Introduction
- Redundant GF arithmetic
- $GF(2^8)$ inversion circuit
- AES encryption S-Box
- Concluding remarks

Tower field inversion: Itoh-Tsujii Algorithm (ITA)

- $GF(q^m)$ inversion based on ITA is given by

$$a^{-1} = (a^{q^1} \times a^{q^2} \times \cdots \times a^{q^{m-1}}) \times (a^{q^0} \times a^{q^1} \times \cdots \times a^{q^{m-1}})^{-1}$$

  - q-th power over $GF(q^m)$ is Frobenius mapping
    - Performed by cyclic shift in NB
  - Usage of norm of input a
    - Considered as subfield ($GF(q)$) element
    - Inversion in *rhs* is $GF(q)$ inversion

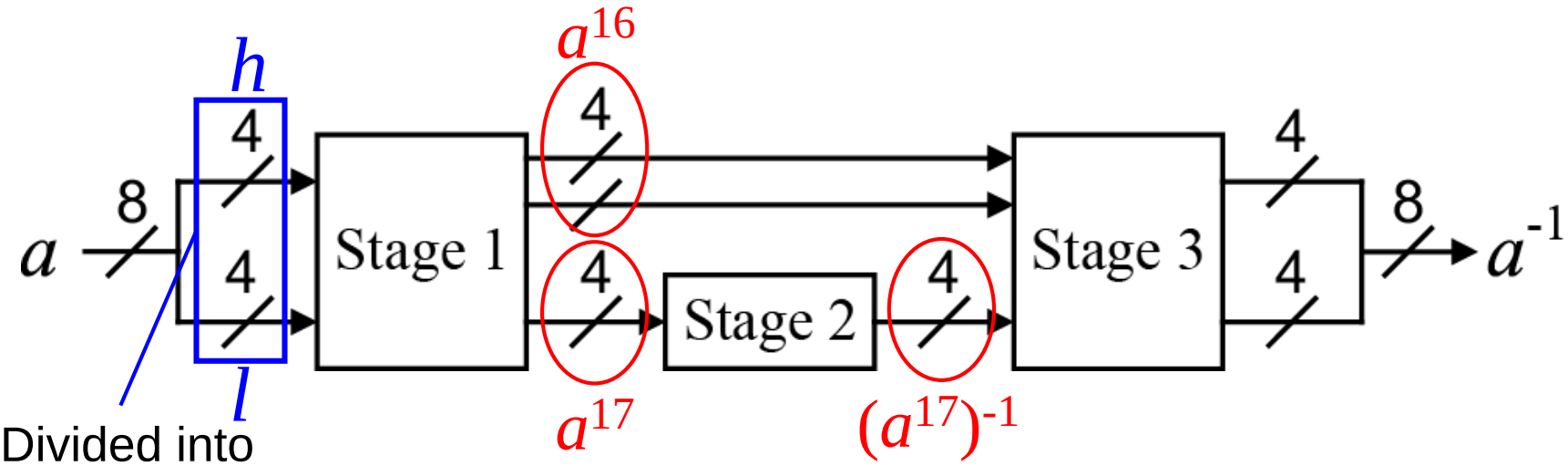
- ITA for $GF((2^4)^2)$ and $GF(((2^2)^2)^2)$, i.e., $q = 16$, $m = 2$

$$a^{-1} = a^{16} \times (a \times a^{16})^{-1}$$

  - a^{16} calculated by only twisting wires
  - $a \times a^{16}$ is $GF(2^4)$ element

ITA-based tower field inversion circuit

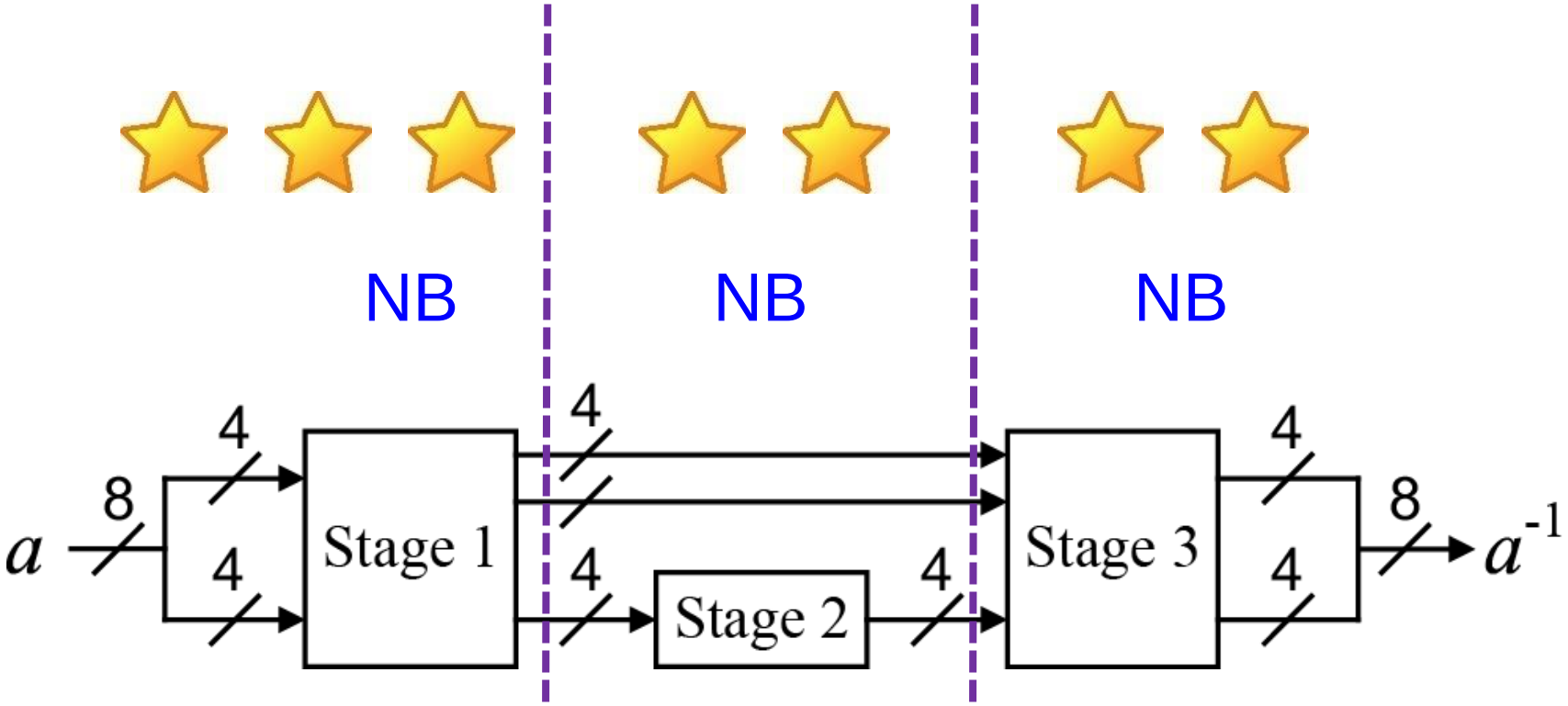
- Consists of 3 stages:
  - Stage 1: 16th and 17th power $a^{-1} = a^{16} \times (a^{17})^{-1}$
  - Stage 2: $GF(2^4)$ inversion $a^{-1} = a^{16} \times (a^{17})^{-1}$
  - Stage 3: final multiplication $a^{-1} = a^{16} \times (a^{17})^{-1}$



Divided into $GF(2^4)$ datapath

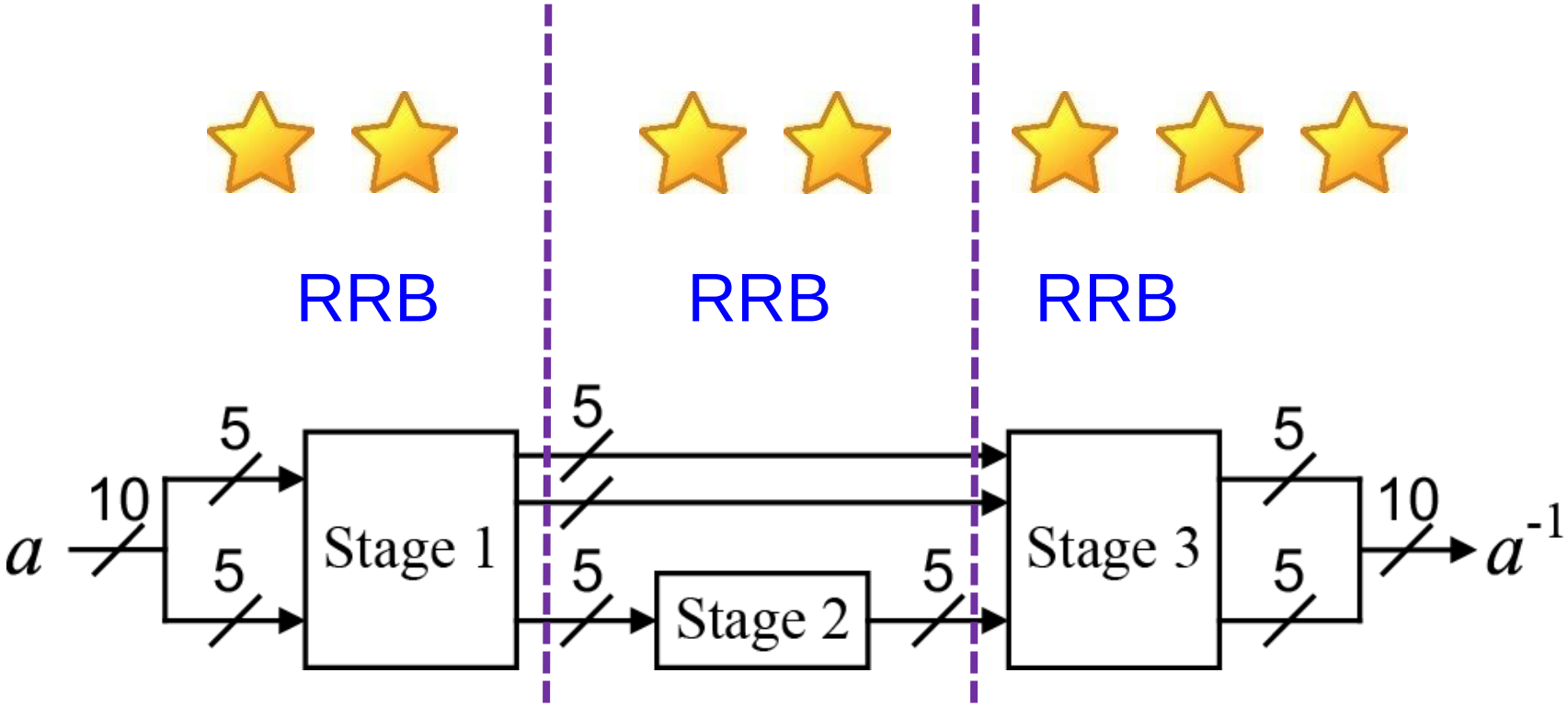
Area-Time efficiency evaluation

NB-based $GF(((2^2)^2)^2)$ inversion [Canright, 2005]



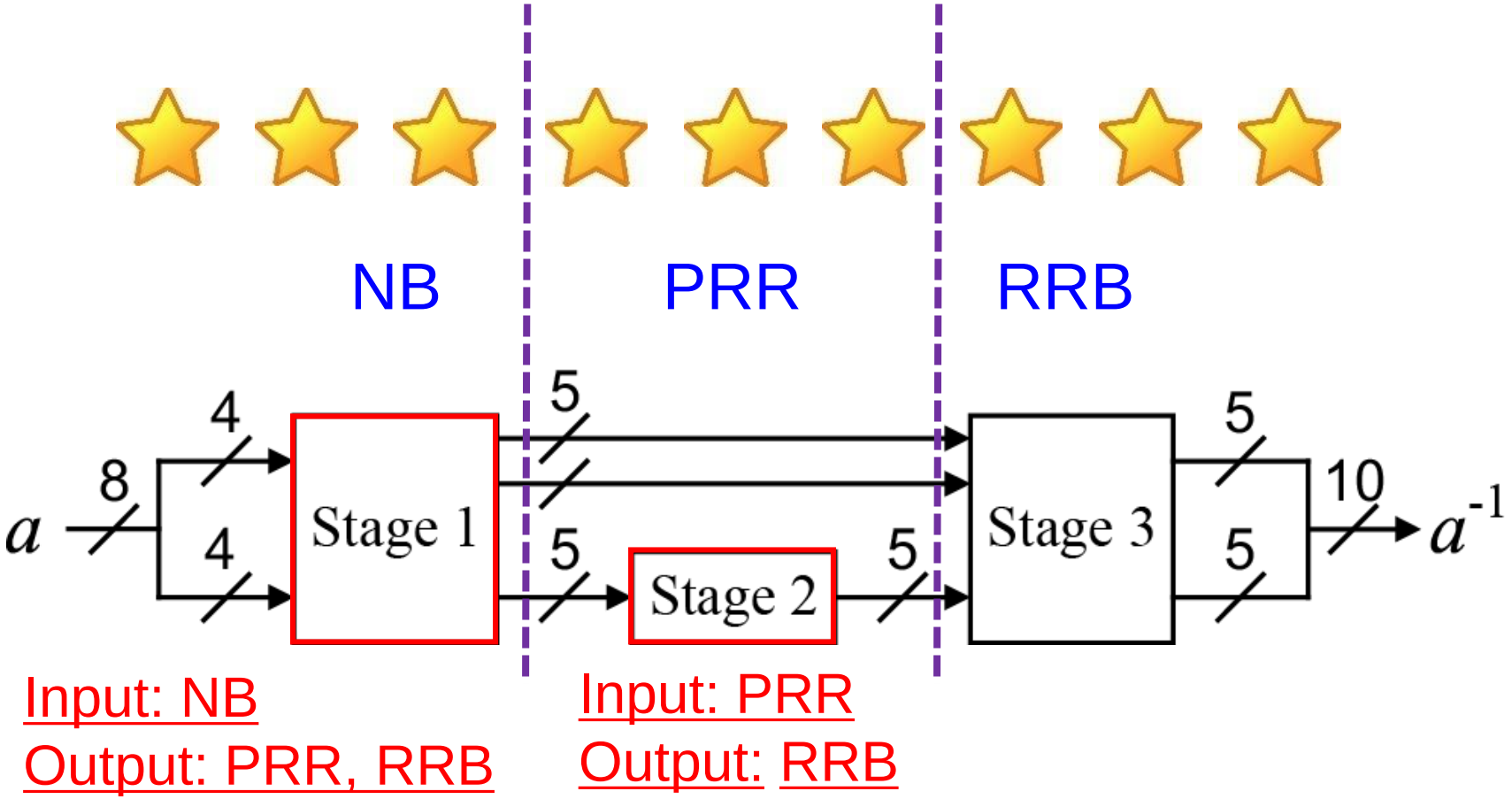
Area-Time efficiency evaluation

RRB-based $GF((2^4)^2)$ inversion [Nekado, 2012]



Proposed concept

- Use the best representation for each stage



To avoid additional gates for conversion

- Mapping from NB to PRR is isomorphism
  - Performed by applying linear mapping Φ to a^{17}
- Merging Φ and constant multiplications in a^{17}
  - Stage 1 output d (a^{17} in PRR) given by

$$d = \Phi(a^{17}) = \Phi(hl\mu^2) + \Phi((h+l)^2\nu)$$
$$= \Phi'(hl) + \Phi''((h+l)^2)$$

    - Φ', Φ'': merged linear mapping
- Symmetric property of $GF(2^4)$ NB for h and l can further reduce Stage 1 delay

Straight-forward mapping	Asymmetric NB	Symmetric NB
$T_A + 5T_X$	$T_A + 4T_X$	$T_A + 3T_X$

T_A, T_X : delay of AND and XOR gate

Effect of PRR in Stage 2

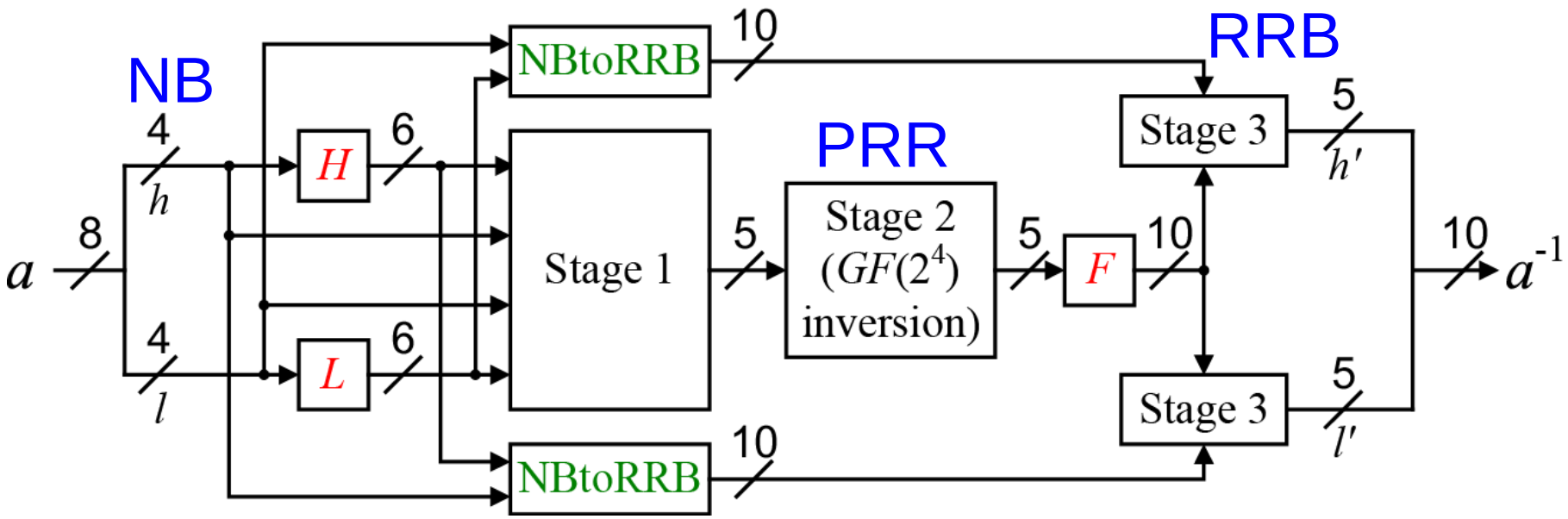
- Don't-care condition of PRR is useful for $GF(2^4)$ inversion function

Field	Representation	Critical delay
$GF((2^2)^2)$	PB	$2T_A + 7T_X$
$GF((2^2)^2)$	NB	$2T_A + 5T_X$
$GF(2^4)$	PB	$2T_A + 2T_X$
$GF(2^4)$	NB	$2T_A + 2T_X$
$GF(2^4)$	RRB	$2T_A + 2T_X$
$GF(2^4)$	PRR	$T_A + T_O + T_X$

T_A, T_O, T_X : delay of AND, OR, and XOR gate

- Conversion from PRR to RRB can also be performed without logic gates

Proposed circuit



- Inputs to stage 1 and 3 should be shared
  - H, L, and F are shared XOR-gate array
  - To save 22 XOR gates
- NBtoRRB converts element from NB to RRB
  - Performed by only wiring

Performance evaluation

	Tower Field	Represen-tation	Gate count (AND, OR, XOR, XNOR, NOT, NAND,NOR)	Critical delay path
Satoh et al.	$GF(((2^2)^2)^2)$	PB	(30, 0, 96, 0, 0, 6, 0)	$4T_A + 17T_X$
Canright	$GF(((2^2)^2)^2)$	NB	(0, 0, 56, 0, 0, 34, 6)	$4T_A + 15T_X$
Nogami et al.	$GF(((2^2)^2)^2)$	PB, NB	(36, 0, 95, 0, 0, 0, 0)	$4T_A + 14T_X$
Rudra et al.	$GF((2^4)^2)$	PB	(60, 0, 72, 0, 0, 0, 0)	$4T_A + 10T_X$
Jeon et al.	$GF((2^4)^2)$	PB	(58, 2, 67, 0, 0, 0, 0)	$4T_A + 10T_X$
Nekado et al.	$GF((2^4)^2)$	RRB	(42, 0, 68, 2, 0, 0, 0)	$4T_A + 7T_X$
This work	$GF((2^4)^2)$	NB, PRR, RRB	(38, 16, 51, 0, 4, 0, 0)	$3T_A + T_O + 6T_X$

T_A, T_O, T_X : Delay of AND, OR, and XOR gate

- Shortest critical delay path
- Gate count comparable with the conventional smallest

Synthesis result

- Synthesis with area optimization
  - Logic synthesis: Design Compiler, Synopsys
  - Cell Library: Standard 65 nm, TSMC

	Tower Field	Representation	Area [GE]	Timing [ns]	AT product
Canright*	$GF(((2^2)^2)^2)$	NB	237.33	2.92	693.00
Nekado et al.**	$GF((2^4)^2)$	RRB	272.67	1.89	515.35
This work	$GF((2^4)^2)$	NB, PRR, RRB	229.67	1.81	415.70

*HDL code was obtained from Canright's website
**HDL code was described by ourselves according to the paper

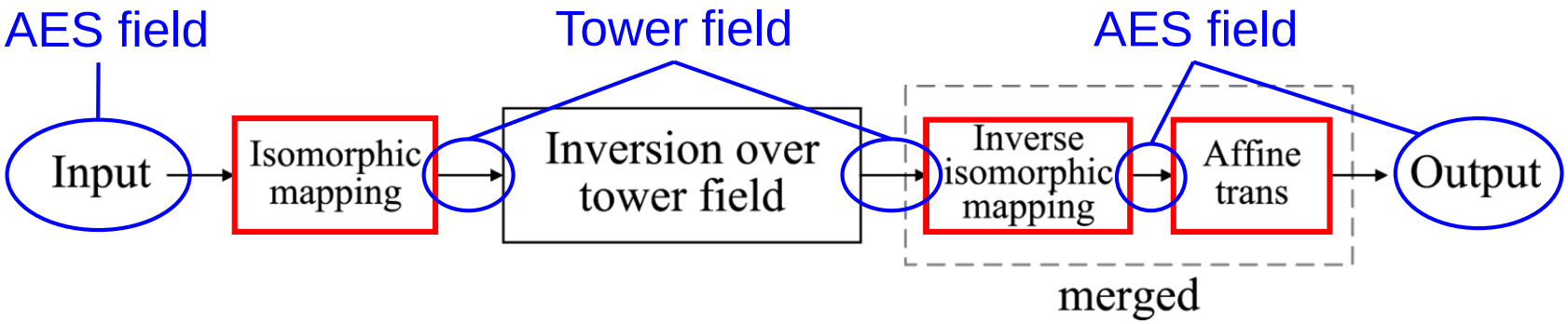
- Our inversion circuit achieved the best efficiency (i.e. AT product) and area

Outline

- Introduction
- Redundant GF arithmetic
- $GF(2^8)$ inversion circuit
- AES encryption S-Box
- Concluding remarks

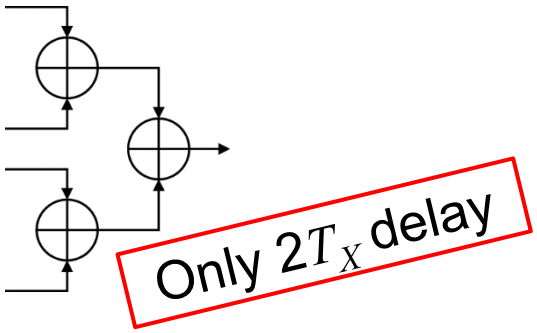
AES encryption S-box

- Require isomorphic mappings and affine trans
  - Later matrix operations should be merged

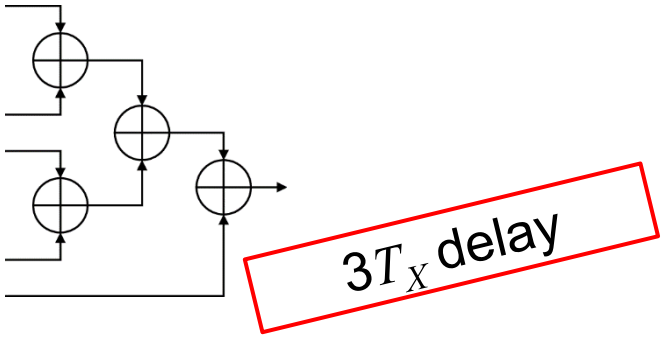


- Conversion matrices optimization for efficiency
  - Hamming weight of each row should be less than 4

Hamming weight = 4



Hamming weight = 5



Synthesis result

	Critical delay path			Area [GE]	Timing [ns]	AT product
	Iso.	Inversion	Iso.$^{-1}$ +Affine			
Canright	$3T_X$	$4T_A + 15T_X$	$3T_X$	315.67	4.30	1,357.38
Nekado et al.	$2T_X$	$4T_A + 7T_X$	$3T_X$	386.00	3.29	1,269.94
This work	$2T_X$	$3T_A + T_O + 6T_X$	$3T_X$	332.00	3.17	1,052.44

- Our S-Box achieved the highest efficiency
  - Synthesis with area-optimization option
- Optimization of conversion matrix operations
  - Canrights' are optimized for low-area
  - Nekados' and ours are optimized for efficiency
    - Low-area optimization of our S-box is a future work

Concluding remarks

- Highly efficient $GF(2^8)$ inversion circuit
  - 38% faster than the conventional one w/o area overhead
- AES encryption S-Box with isomorphism optimization for efficiency
  - Achieved the lowest Area-Time product
- Future work
  - Further optimization of conversion matrices
    - Lower-area or/and higher efficiency
    - Both encryption and decryption S-box
  - Design of AES datapath with the proposed S-box