A Novel Permutation-based Hash Mode of Operation FP
and
The Hash Function SAMOSA

Souradyuti Paul, Univ. of Waterloo, Canada and K.U.Leuven, Belgium

Ekawat Homsirikamol, George Mason University, USA

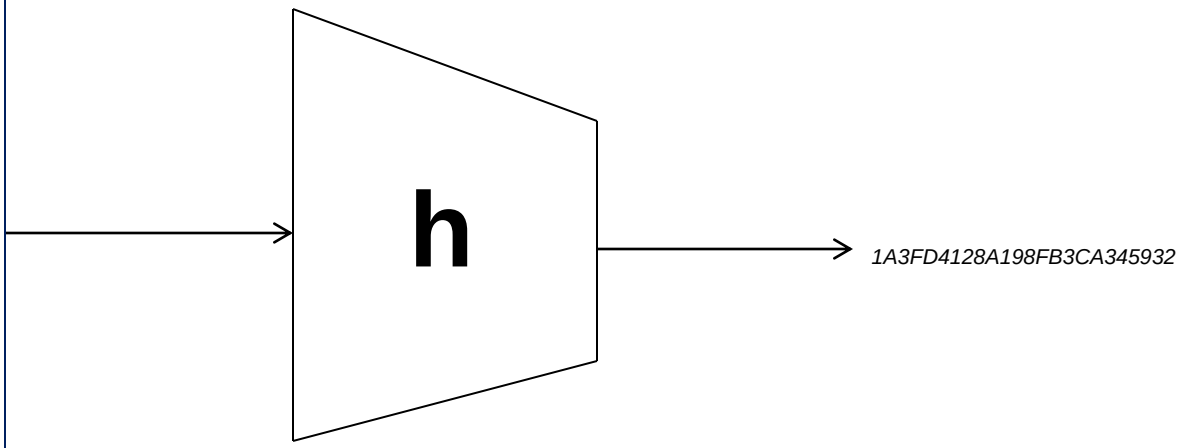
Kris Gaj, George Mason University, USA

12/12/12

What is a hash function?

- $h:\{0,1\}^* \rightarrow \{0,1\}^n$

The UK-based bank was alleged to have helped launder money belonging to drug cartels and states under US sanctions.
Earlier this year HSBC admitted having poor money laundering controls following a US Senate investigation.
Last month announced it had set aside $1.5bn to cover the costs of any settlement or fines.
The deal could be announced as early as Tuesday, the Wall Street Journal reports.
It follows the announcement of a similar but much smaller settlement with UK-based Standard Chartered bank, which will pay $300m in fines for violating US sanction rules.



Uses of Hash Function

- digital signatures
- data authentication
- password-protection
- micro-payments
- Commitment protocol
- pseudo-random string generation/key derivation
- entropy extraction
- MACs, enciphering schemes
- 2005: 800 uses of MD5 in Microsoft Windows

Properties of a Hash Function

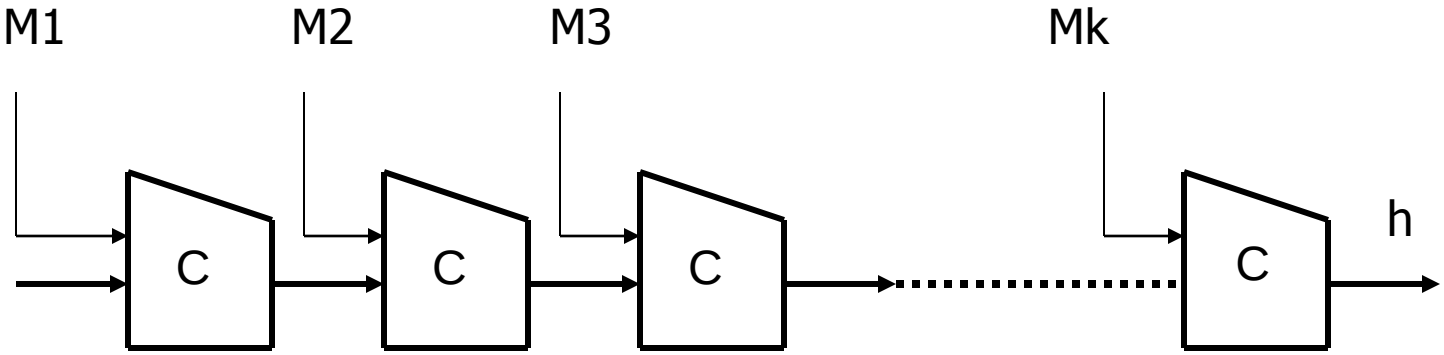
- Pre-image resistance
- 2nd Pre-image resistance
- Collision resistance
- Length extension attack resistance
- Indifferentiability attack resistance

Basic Recipe to Design an Iterative Hash Function

- Mode of Operation
- An iterative Function/Compression function
- Padding Rule
- Constants if any in the above three

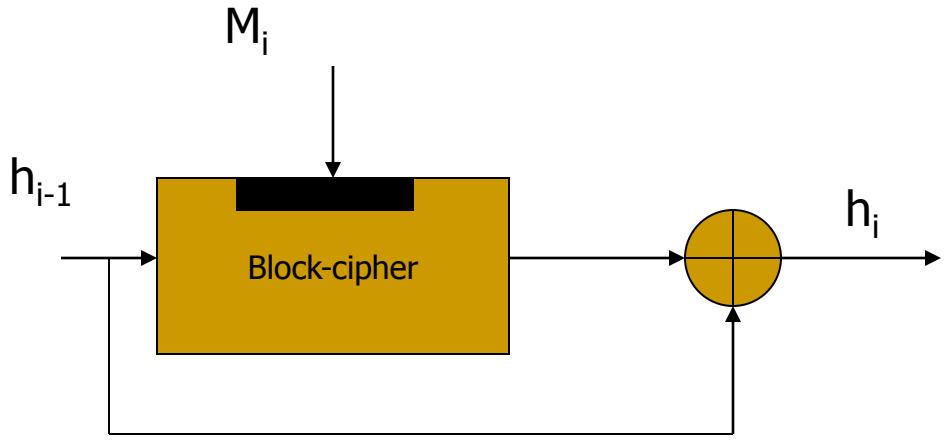
Example: Merkle-Damgard Mode

Pad(M)=$M_1M_2M_3$----M_K



Compression Function using a block cipher

- Davis-Meyer Construction



- More: Matyas-Meyer-Oseas, Miyaguchi-Preneel, etc.

Hash functions using Merkle-Damgard and Block Cipher

- SHA-0/1/2 (NIST/NSA)
- MD-4/5 (Rivest)
- RIPEMD (Dobbertin, Bosselaers and Preneel)
- WHIRLPOOL (Barreto and Rijmen)
- TIGER (Anderson and Biham)
- ..
- ..

Merkle-Damgard Mode is Damaged

- Length-extension attacks
- Joux multi-collision attacks
- Kelsey-Schneier 2^{nd} preimage attack
- Kelsey-Kohno Herding attack
- …

NIST SHA-3 Hash Function Competition

None of the 64 submitted algorithms was based on Merkle-Damgard Mode!!!

Block cipher may not be the best alternative

- Key schedule often turns out to be weak
- Implementing a wide block cipher requires relatively larger memory
- Ideal cipher assumption is a stronger assumption

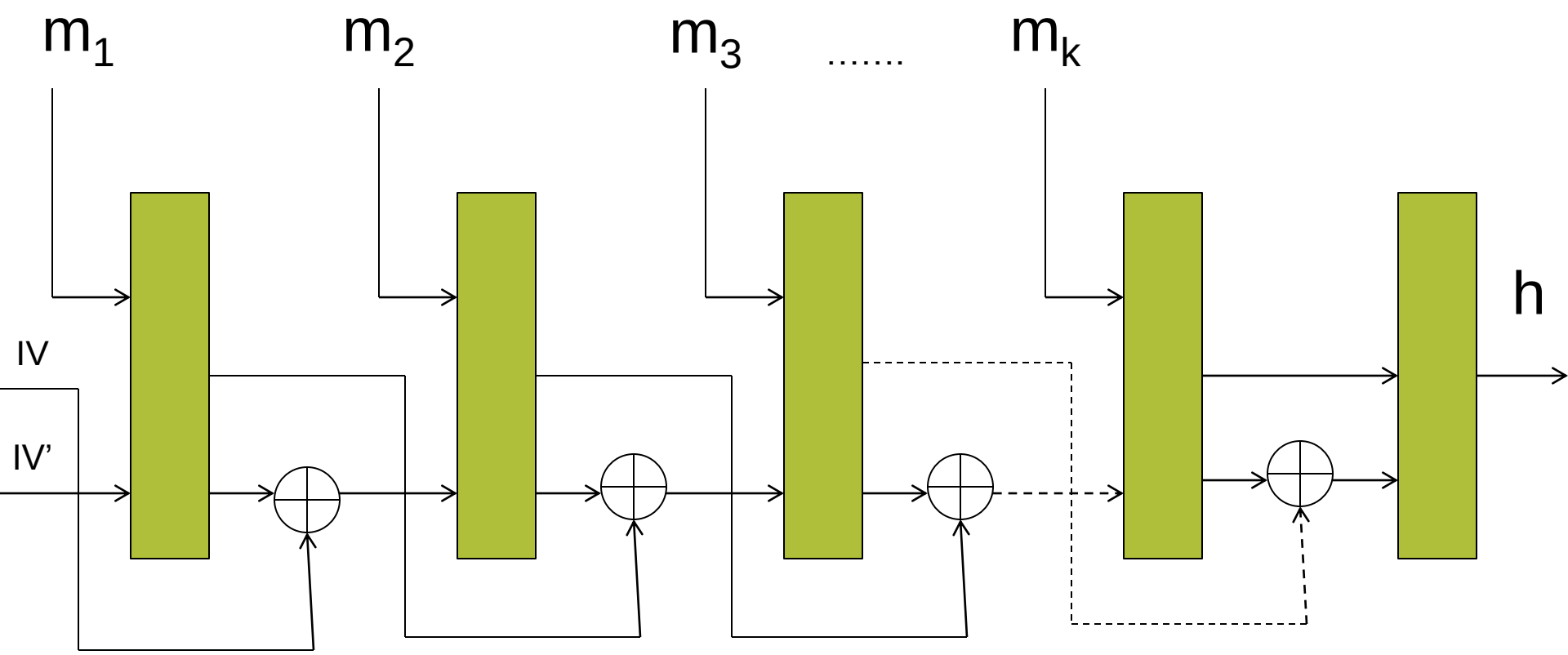
Block-cipher-based Hash Functions On the Decline

- Most of the hash functions in the recently concluded NIST SHA 3 hash function competition were not block-cipher-based
- 9 out 14 semi-finalists are permutation-based
- 3 out of 5 finalists are permutation-based
- The SHA-3 winner is permutation-based

Permutation-based Hash Functions

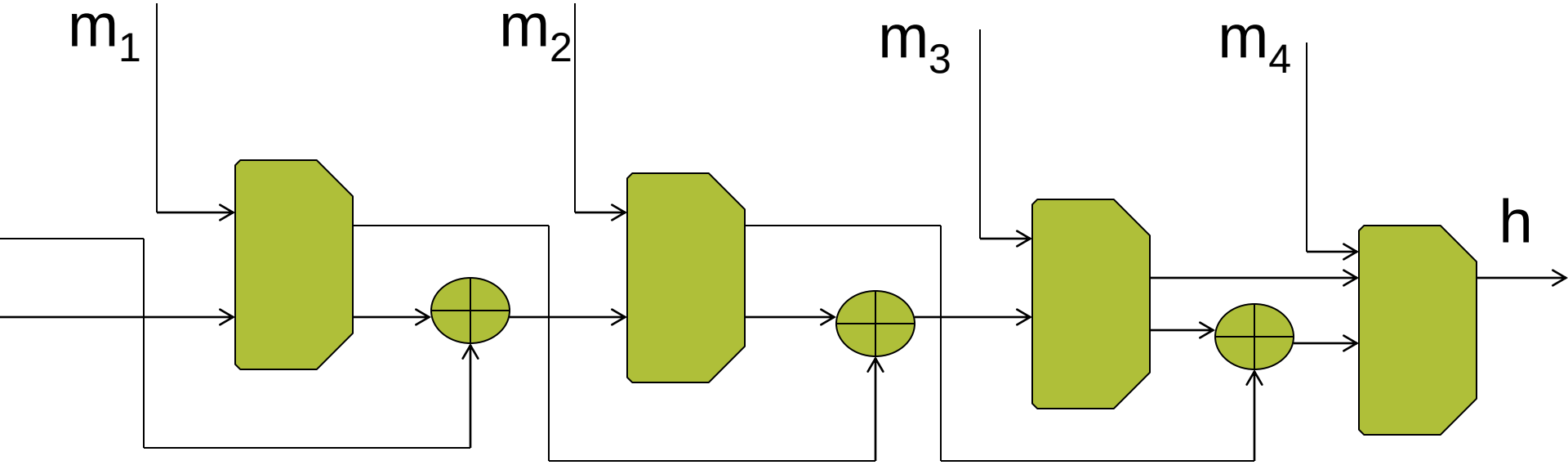
- Grøstl
- JH
- Sponge
- MD6
- Luffa
- HAMSI
- Parazoa
- ..

A Novel Permutation-based Mode: FP



All wires are n bits each

Origin of FP: The FWP Mode (2010)



FWP: rate=0.5; presently holds the record of achieving the best indifferentiability bound of 2n/3 bits (up to an additive constant)

How FP Mode compares With Other Permutation-based Modes

Mode of operation	Msg-blk (ℓ)	Size of π (a)	Rate (ℓ/a)	Indiff. bound		# of independent permutations
				lower	upper	
Hamsi [24]	$n/8$	$2n$	0.07	$n/2$	n	1
Luffa [5]	$n/3$	n	0.33	$n/4$	n	3
Sponge [4]	n	$3n$	0.33	n	n	1
Sponge [4]	n	$2n$	0.5	$n/2$	$n/2$	1
JH [29]	n	$2n$	0.5	$n/2$	$n(1-\epsilon)$	1
Grøstl [17]	n	$2n$	0.5	$n/2$	n	2
FP	$\boldsymbol{n}$	$\boldsymbol{2n}$	**0.5**	$\boldsymbol{n/2}$	$\boldsymbol{n}$	**1**
MD6 [14]	$6n$	$8n$	0.75	n	n	1
$\mathbf{FP}^{\mathrm{Ext1}}$	$\boldsymbol{6n}$	$\boldsymbol{7n}$	**0.85**	$\boldsymbol{n/2}$	$\boldsymbol{n}$	**1**

Proof of Indifferentiability

- We proved indifferentiability of FP up to n/2 bits. Full proof in eprint archive
- We conjecture that the bound could be improved up to 2n/3 bits

(We do not explain the proof here. But ….)

Small Cash Rewards

- Indifferentiability attack on FP mode with work n/2-bit or less (US$ 100)
- Indifferentiability attack on FP mode with work between n/2 and 2n/3-bit (US$ 50)
- Improvement of bound to 3n/4-bit or more (US$ 100)
- (Serious)Bug in the proof (US$99)

SAMOSA From Grøstl

Grøstl

SAMOSA

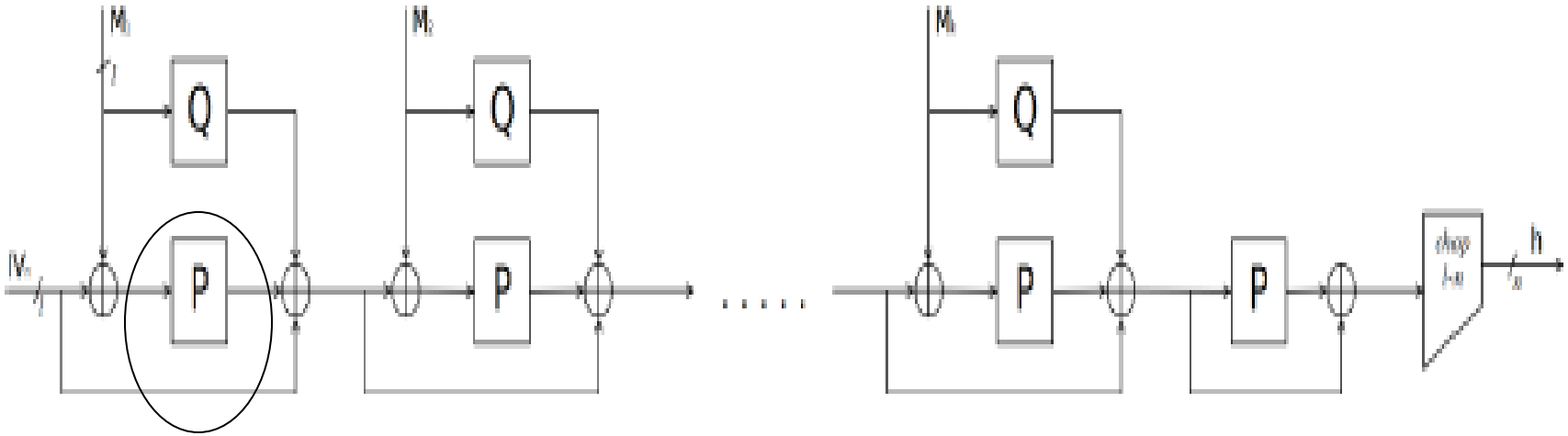
SAMOSA: A Hash Function Based on FP Mode

SAMOSA Hash Function

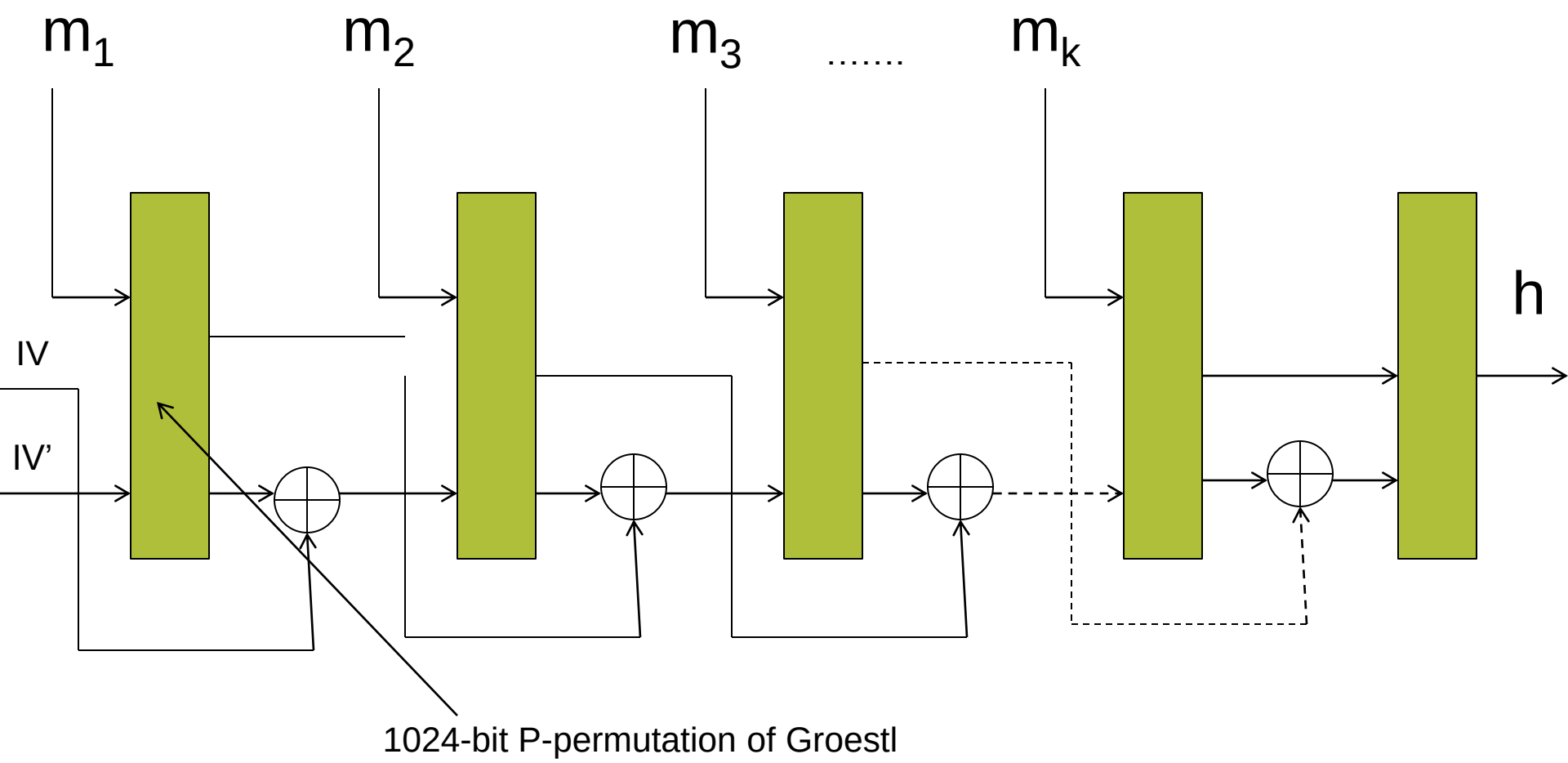
=

FP Mode + P-permutation of Grøstl

Grøstl Hash Function



SAMOSA Hash Function



Security of SAMOSA Hash Function

- Security of the FP Mode
  - Conjectured to be indifferentiable secure up to n bits
- Security of the 1024-bit P-permutation of Grøstl
  - Grøstl is one of the most heavily analyzed hash functions

SAMOSA: Hardware Implementation Motivation

- Performance in software and hardware decides whether an algorithm can be adopted as a standard when security levels of competing algorithms are comparable

- Grøstl is used as a reference point as it shares a very significant building block, permutation P, with SAMOSA

Grøstl: Hardware Implementation Previous Work

Grøstl-256, without padding unit, in Xilinx Virtex 5 FPGA

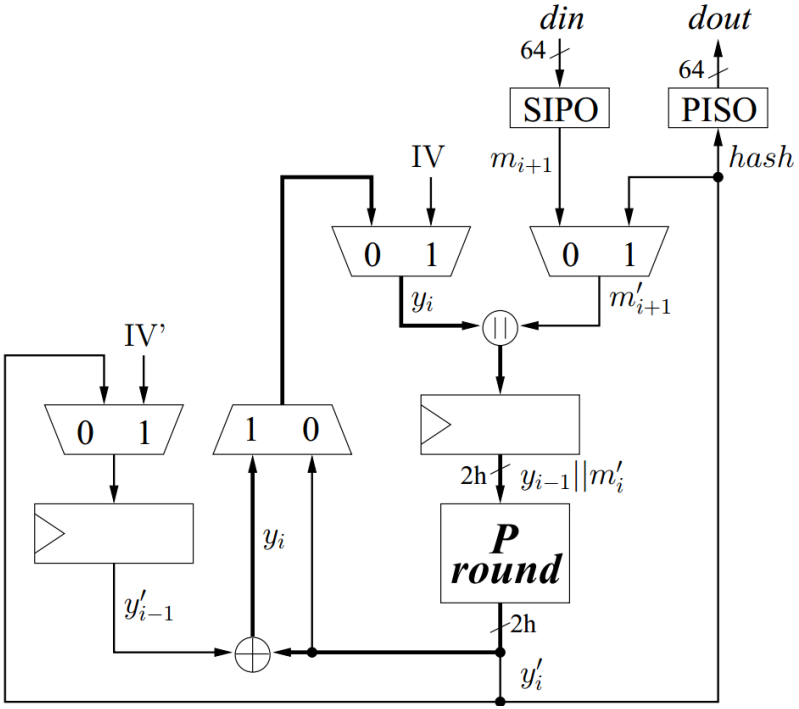
Source	Architecture Variant	Throughput [Mbit/s]	Area [CLB slices]	Thr/Area [(Mbit/s)/ CLB_slices]
Latif et al.	x1 (P/Q)	6200	1419	4.37
Gaj et al.*	**x1 (P/Q)**	**6117**	**1795**	**3.41**
Homsirikamol et al.	x1 (P/Q)	6072	1912	3.18
Gaj et al.	/2(v) (P/Q)	3721	1195	3.11
Homsirikamol et al.	/2(v) (P+Q)	4014	1598	2.51
Gaj et al.	x1 (P+Q)	7213	2906	2.48
Baldwin et al.	x1 (P+Q)	7709	3137	2.46
Guo et al.	x1 (P+Q)	5027	3798	1.32

**Implementation by Latif et al. is not portable among FPGA families, and its source code has not been published.*

Therefore, the implementation by Gaj et al. is used for comparison.

SAMOSA: Hardware Implementation Top-Level Comparison

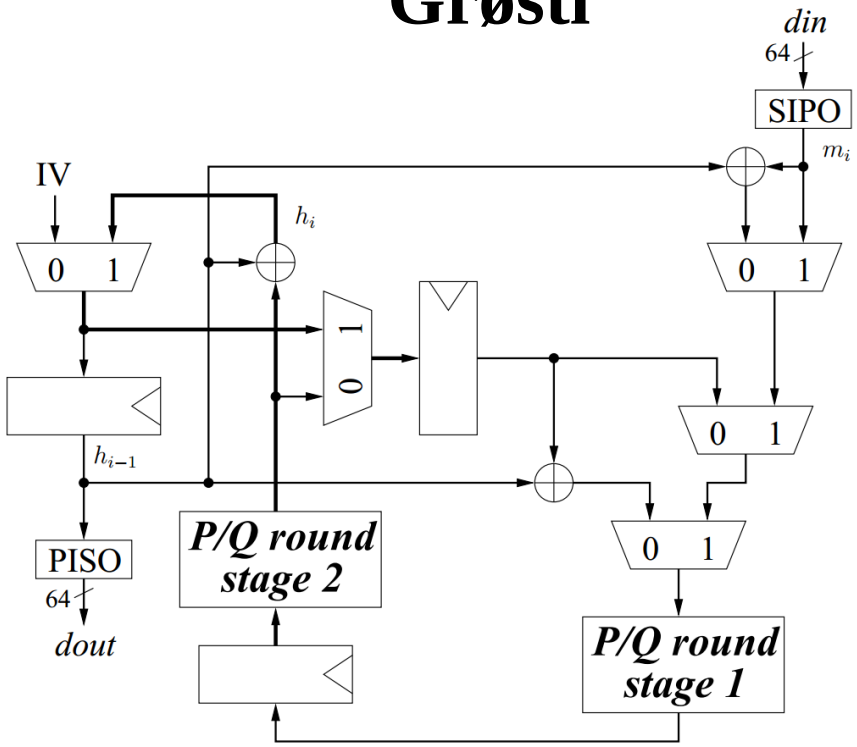
SAMOSA



SAMOSA-256: h=256
SAMOSA-512: h=512

Note: **All buses** are **h-bit wide**, unless specified otherwise

Grøstl



Grøstl-256: b=512
Grøstl-512: b=1024

Note: **All buses** are **b-bit wide**, unless specified otherwise

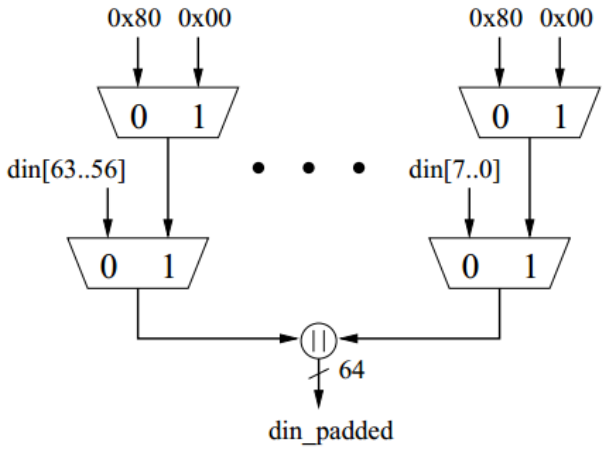
SAMOSA: Hardware Implementation Major Differences

	SAMOSA	**Grøstl**
Clock cycles per round	r	2r+1
Input block size	h	2h
Logic outside of the main round (P or P/Q)		
Number of 1-bit registers	3h	4h
Number of 2-to-1 multiplexers	4h	10h
Number of 2-input XOR gates	h	6h
Number of bits in SIPOs	h	2h

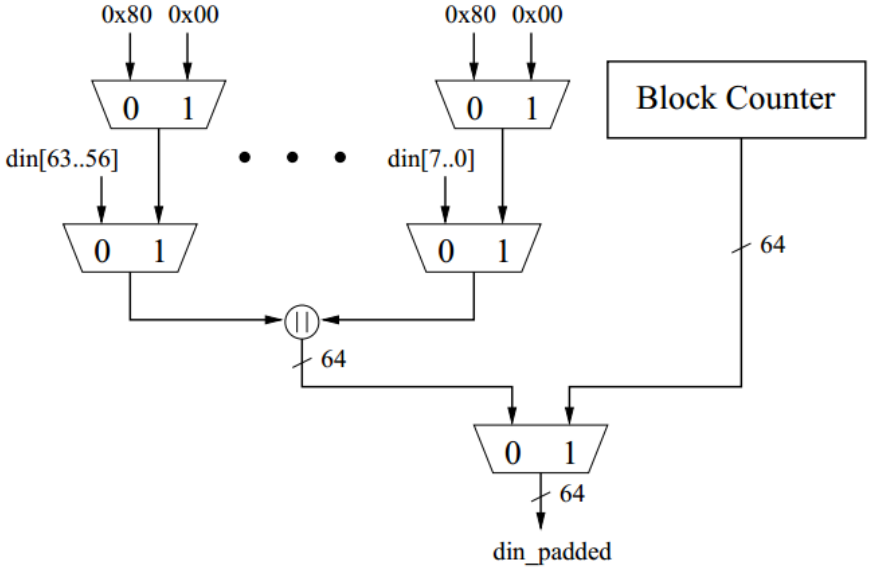
Notation:
r = number of rounds
h = hash size

SAMOSA: Hardware Implementation Padding Unit Comparison

SAMOSA



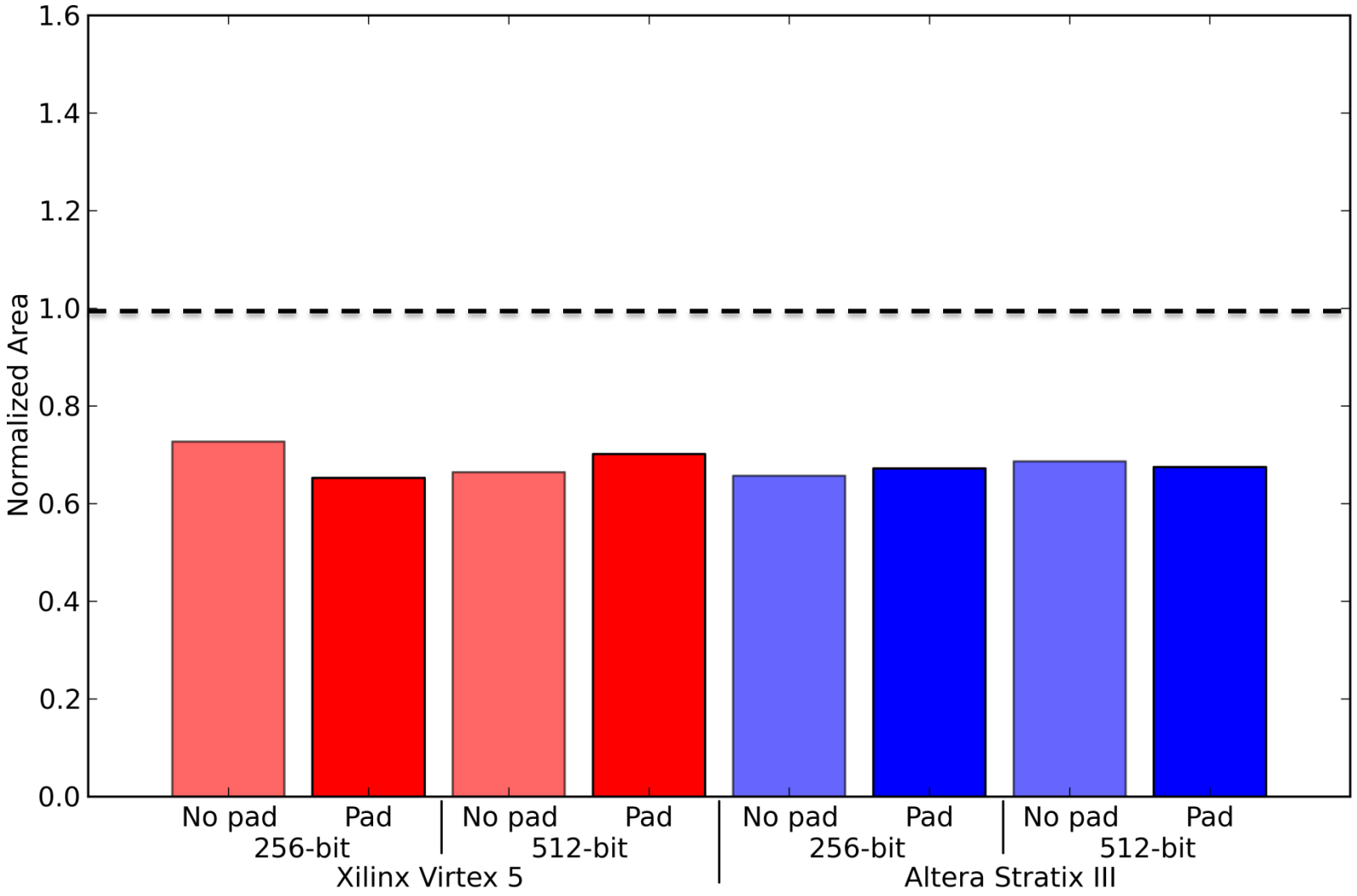
Grøstl



Block counter is not needed any longer in SAMOSA

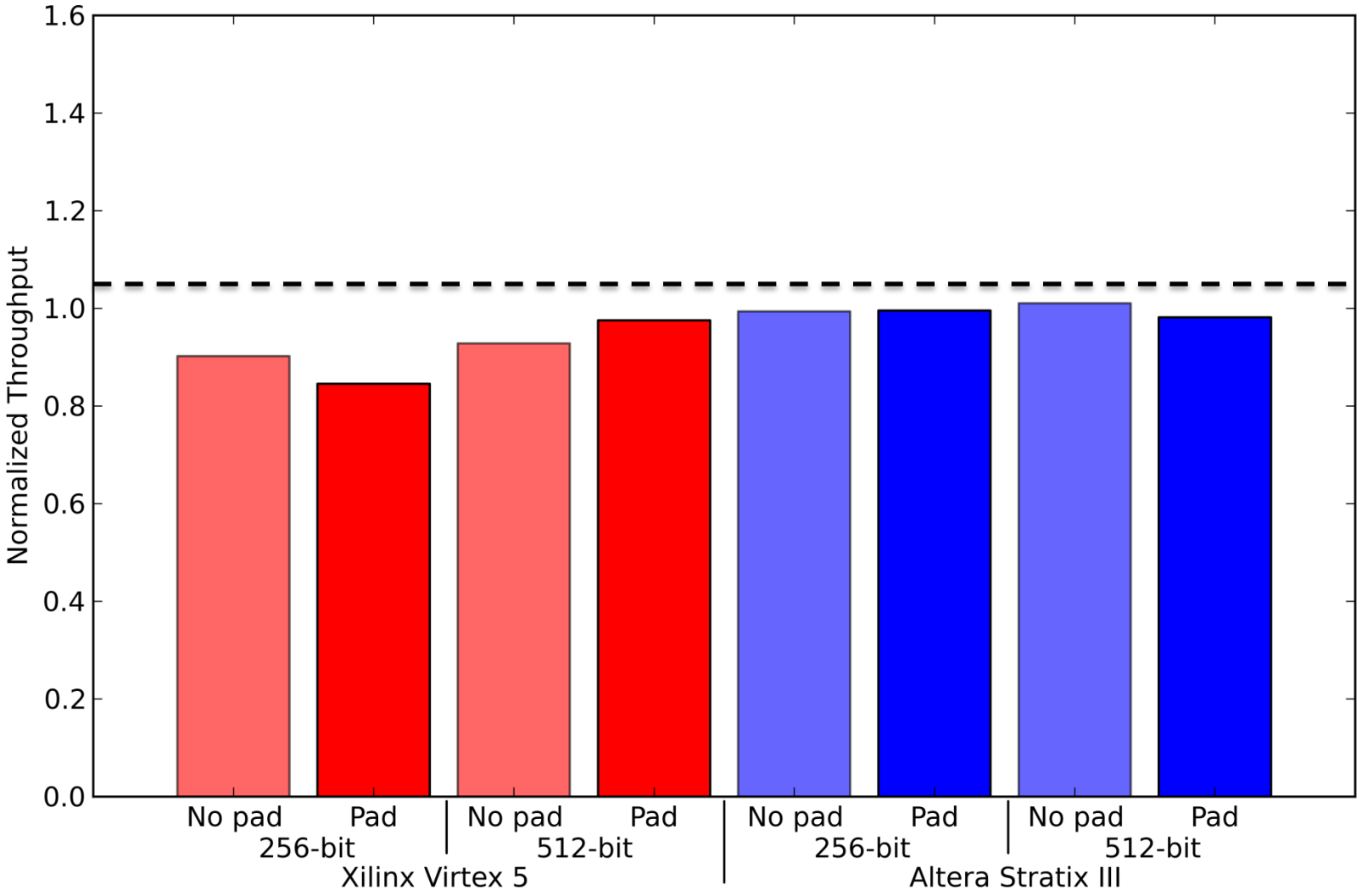
SAMOSA: Hardware Implementation Normalized Area

Normalized to Grøstl with the same padding mode, hash size, and FPGA family



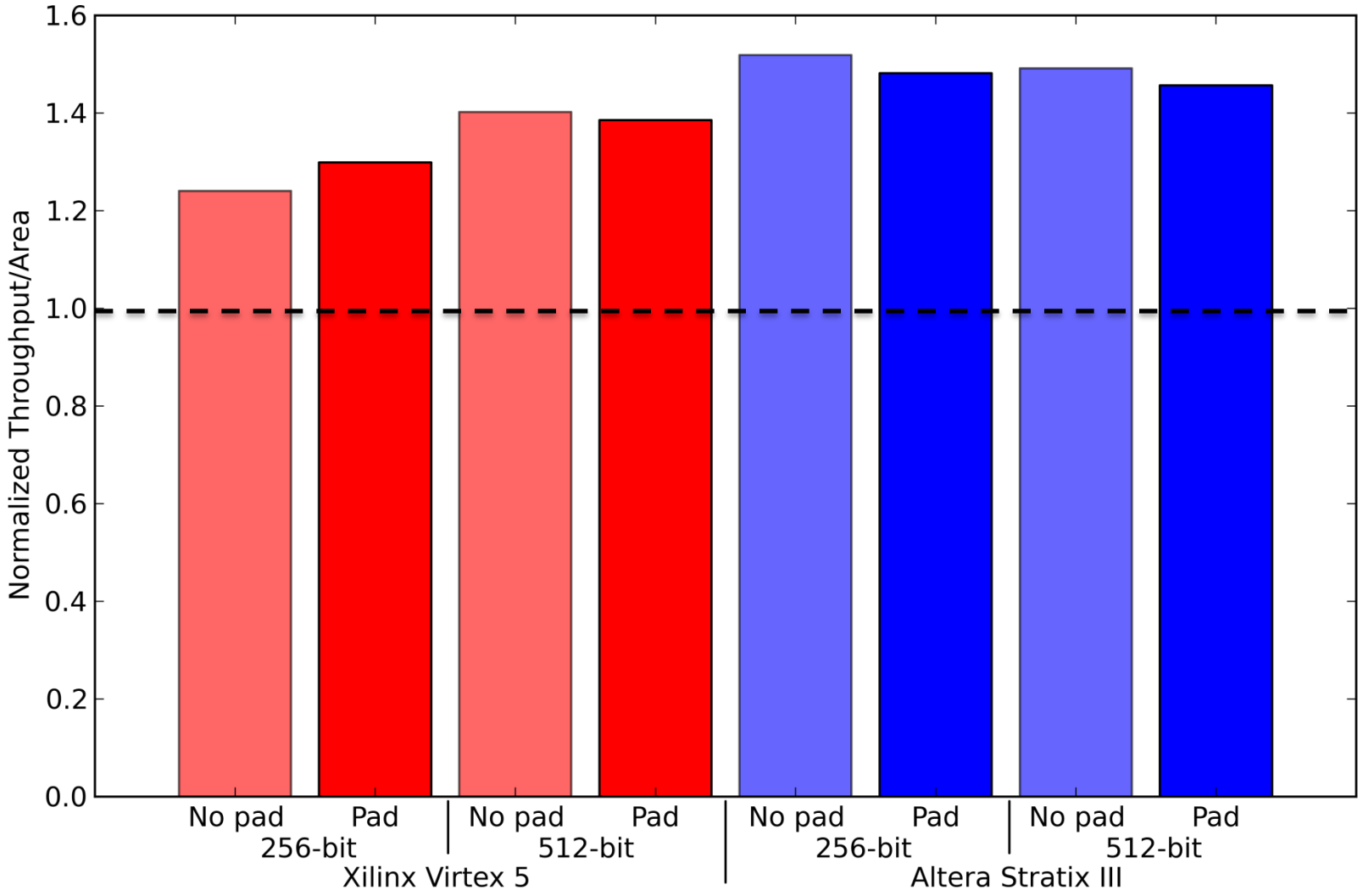
SAMOSA: Hardware Implementation Normalized Throughput

Normalized to Grøstl with the same padding mode, hash size, and FPGA family

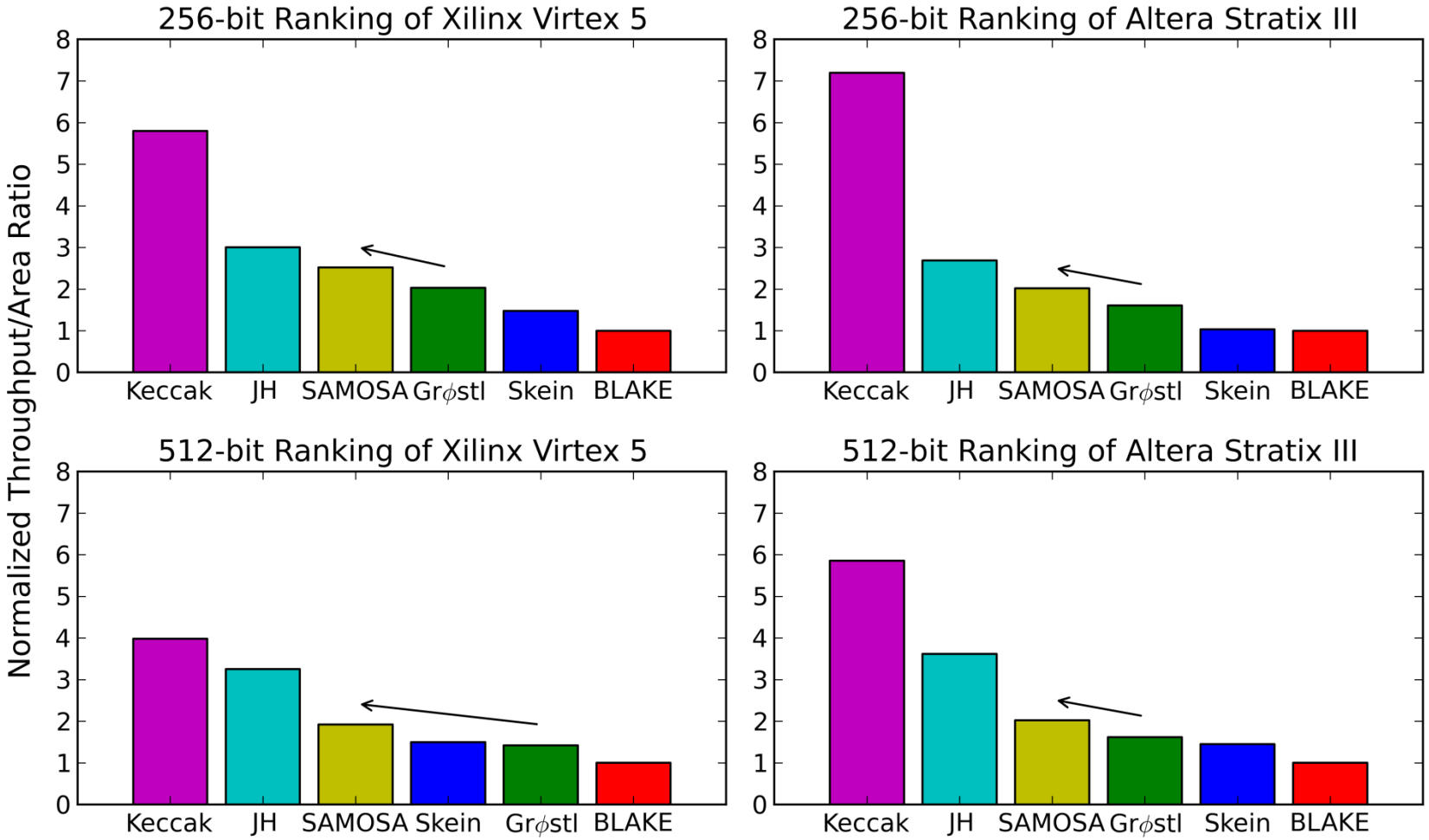


SAMOSA: Hardware Implementation Normalized Throughput/Area ratio

Normalized to Grøstl with the same padding mode, hash size, and FPGA family



SAMOSA: Hardware Implementation Best Single Message Ranking



All values normalized to the lowest throughput/area ratio (i.e., to the results for BLAKE)

Conclusions

- We propose a new permutation-based hash mode of operation FP and the hash function SAMOSA based on that.

- SAMOSA performs consistently better than BLAKE, Grøstl and Skein, and loses only to Keccak and JH

- The performance of SAMOSA may be improved by changing the underlying permutation

Thanks

Questions?