
بسم الله الرحمن الرحيم

Efficient Hardware Implementations and Hardware Performance Evaluation of SHA-3 Finalists

By

Kashif Latif, Athar Mahboob, Arshad Aziz and M Muzaffar Rao

ISRL

Information Security Research Laboratory

National University of Sciences and Technology, Islamabad, Pakistan

OUTLINE

- Objective
- Implementation Methodology
- LUT Primitives in Xilinx HDL Library
- Implementation Details SHA-3 Finalists
- Results
- Performance Comparison
- Conclusion
- Q/A

OBJECTIVE

- Resource efficient and high speed implementations
- Fully autonomous designs
- Utilization of specific internal resources of FPGAs instead of direct coding
- Fair comparison using uniform implementation environment

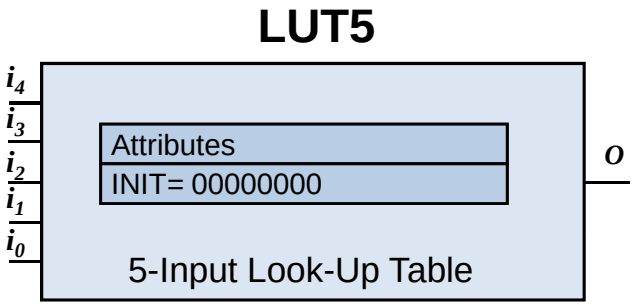
IMPLEMENTATION METHODOLOGY

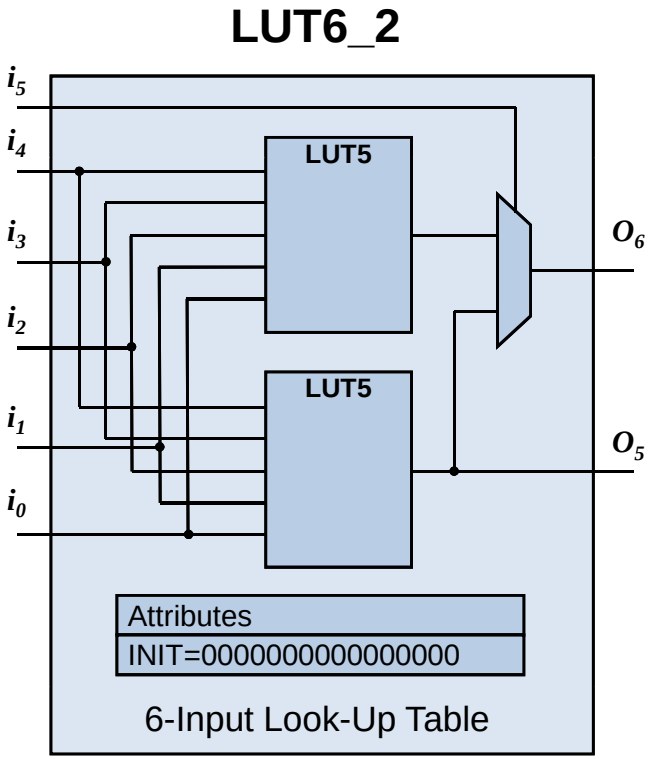
- Common environment:
  - Level of expertise → common implementer
  - Coding language → Verilog®
  - Development platform → Xilinx ISE 13.1
  - Design methodology → on next slide
- Common Input/Output interface
- Overhead suppression
  - Padding → assume already padded message
  - Salt input
  - HMAC
  - Hash Tree functionality

DESIGN METHODOLOGY

- Same set of hardware resources
- Assured it by forcing our designs to map on LUT based logic and not to use dedicated resources like BRAMs, Multipliers and DSP Slices.
- Mapping to LUT is assured by using LUT primitives from Xilinx HDL library
- Memories are implemented using distributed RAMs/ROMs

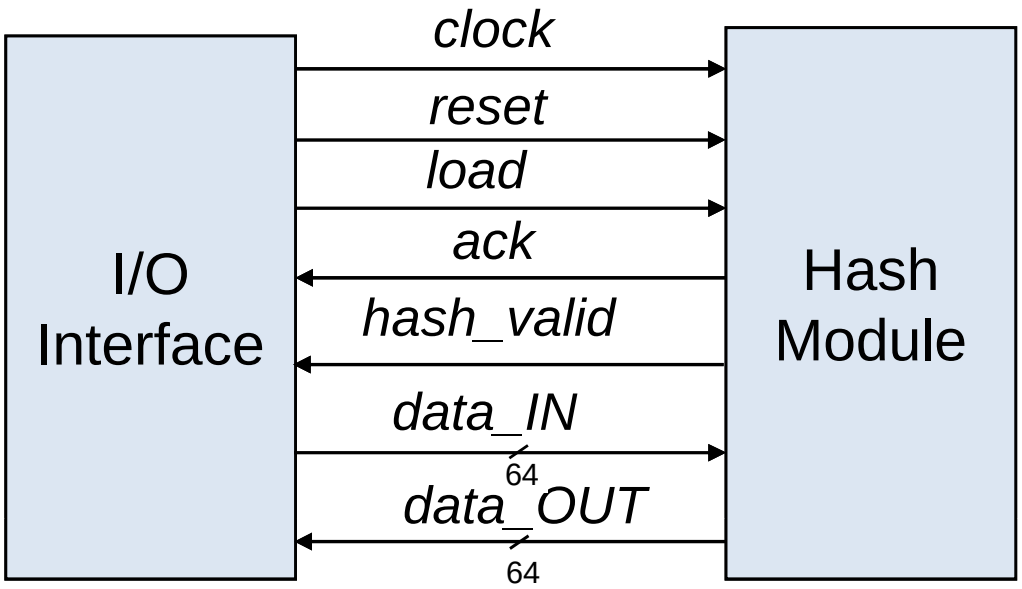
LUT5 and LUT6_2 Primitives from Xilinx HDL Library





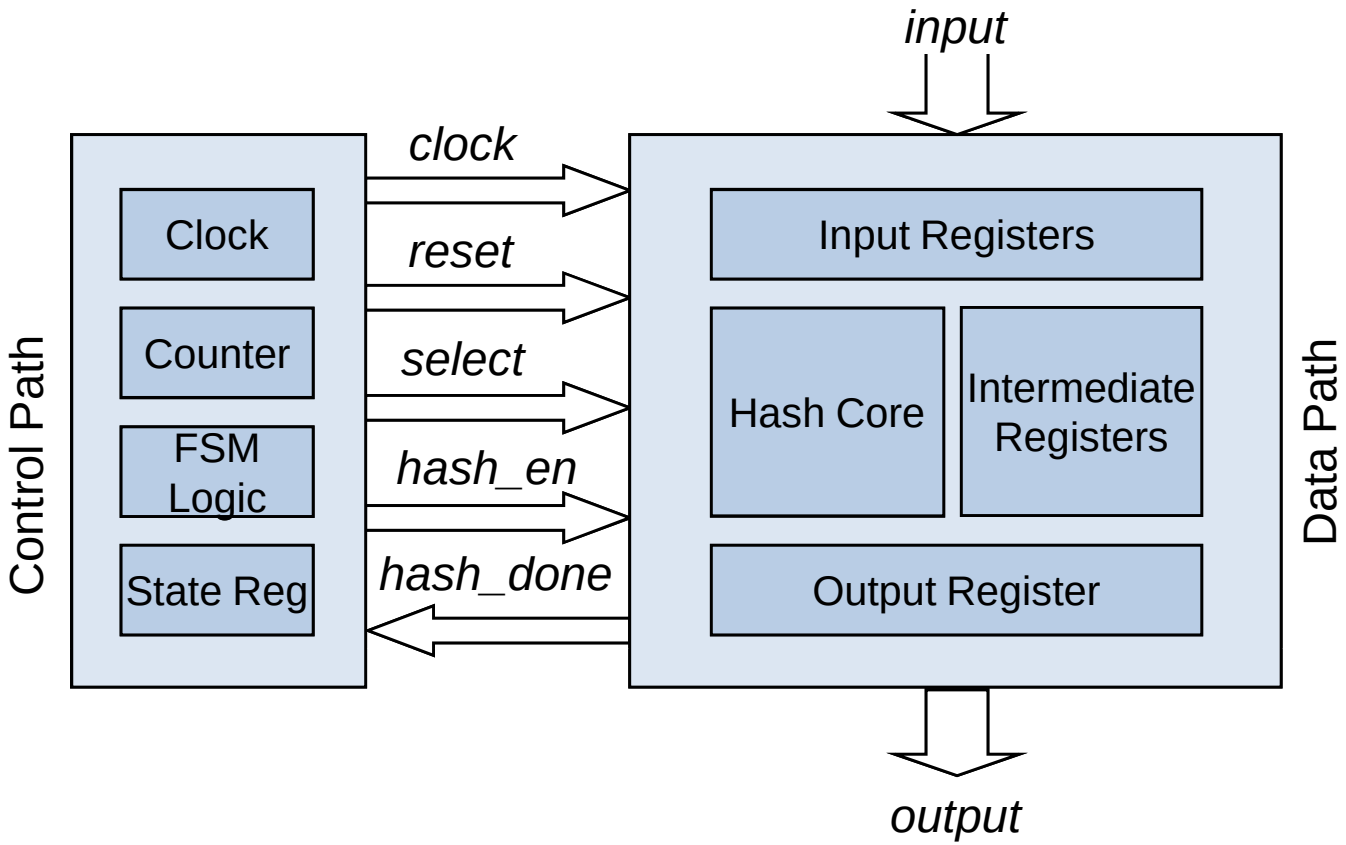
IMPLEMENTATION

- Input/output interface



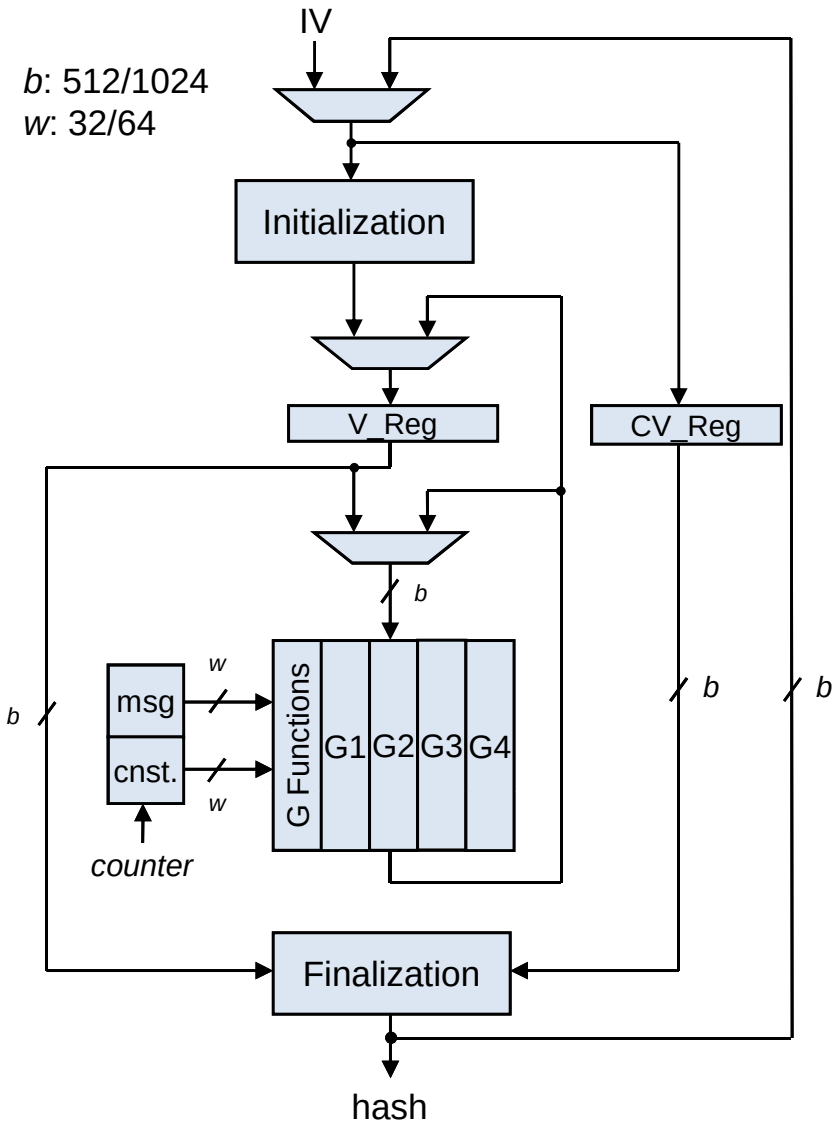
IMPLEMENTATION

- Data path and Control path



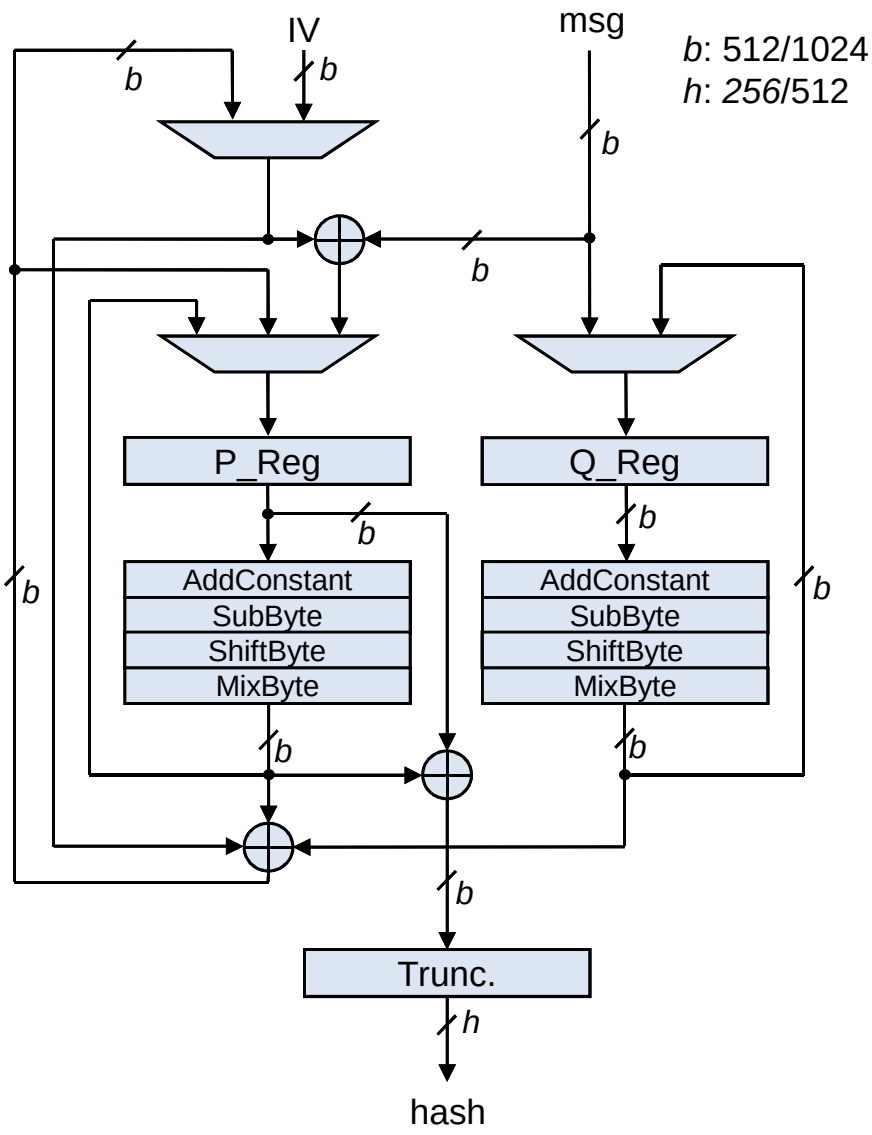
IMPLEMENTATION

- Data path Architecture for BLAKE



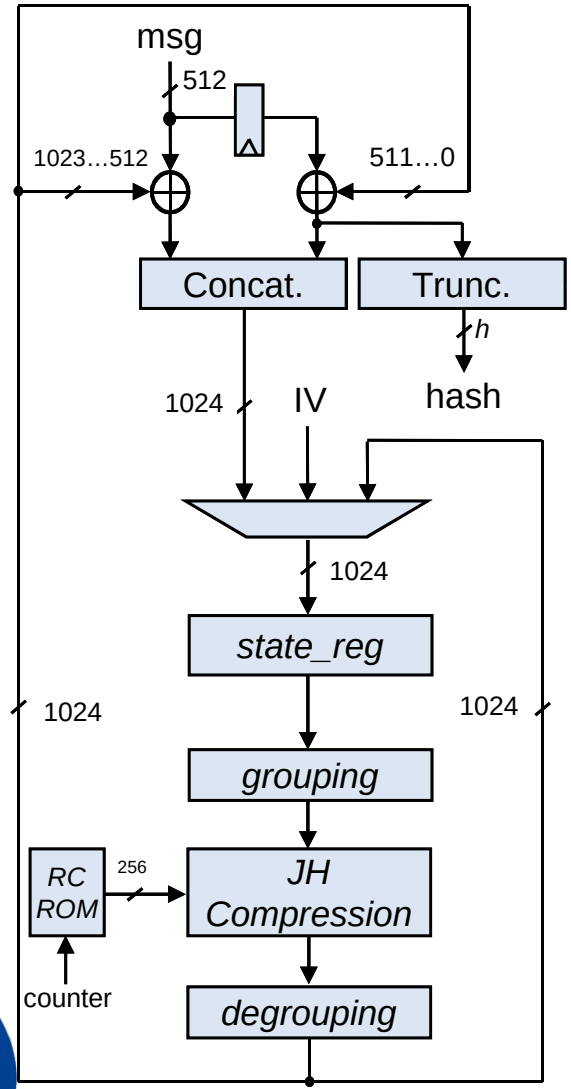
IMPLEMENTATION

- Data path Architecture for Grøstl

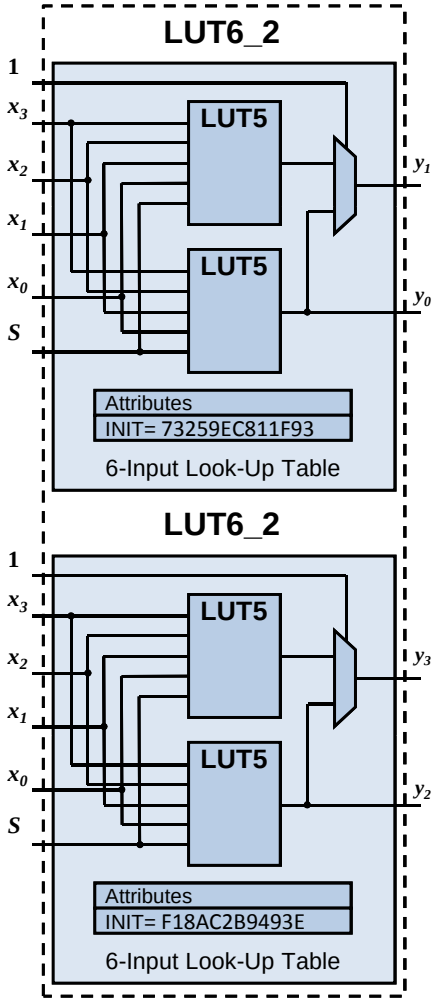


IMPLEMENTATION

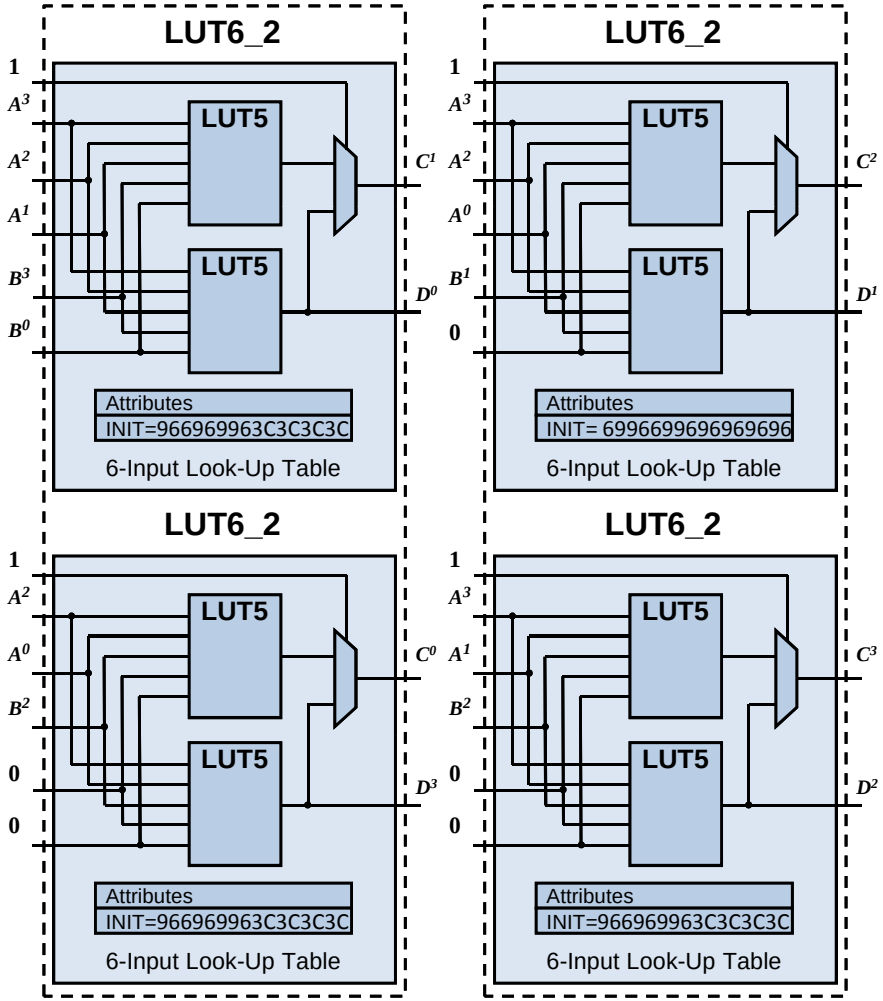
- Data path Architecture for JH



(a) Data path of JH



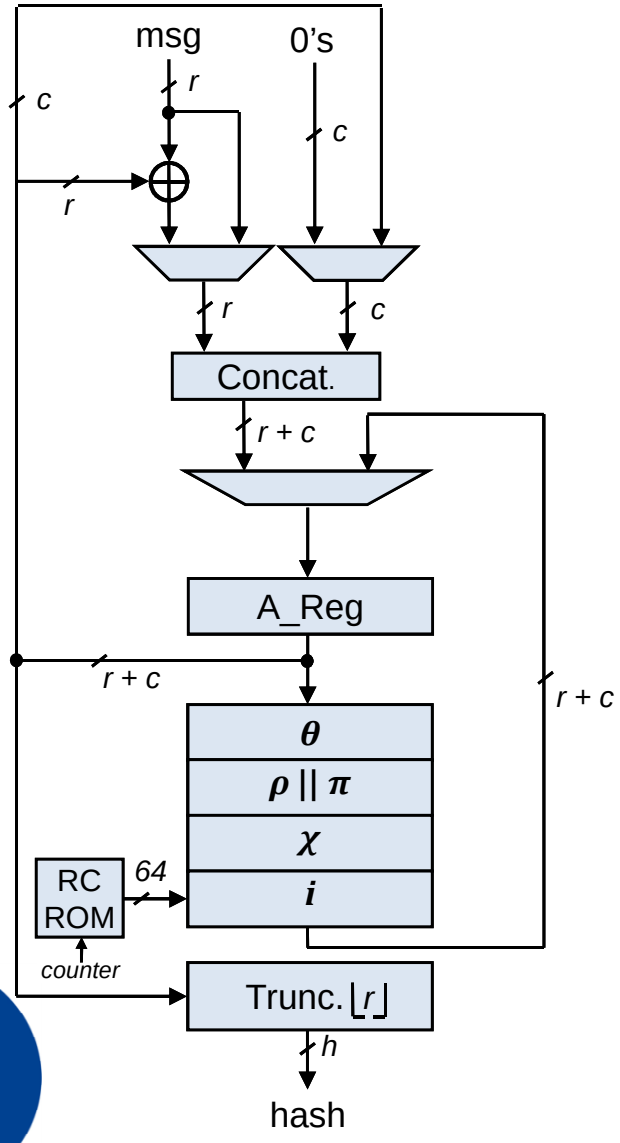
(b) 5-to-4 bit S-box

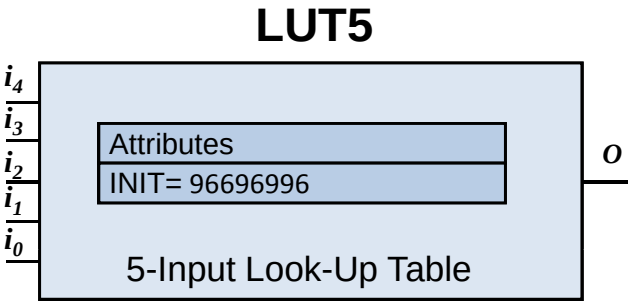


(c) Linear Transformation

IMPLEMENTATION

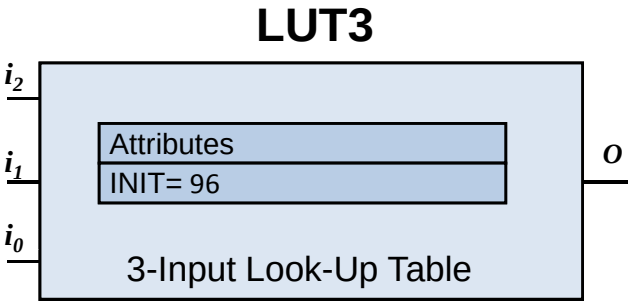
- Data path Architecture for Keccak





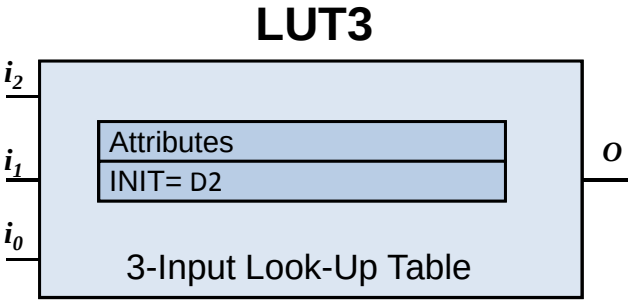
$O = i_0 \oplus i_1 \oplus i_2 \oplus i_3 \oplus i_4$

5-bit XOR used in θ step



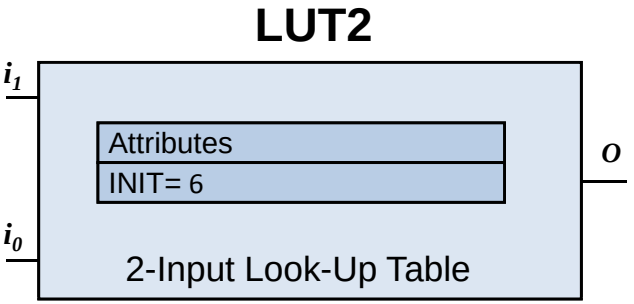
$O = i_0 \oplus i_1 \oplus i_2$

3-bit XOR used in θ step



$O = i_0 \oplus (\sim i_1 \,\&\, i_2)$

3-bit Logic used in χ step

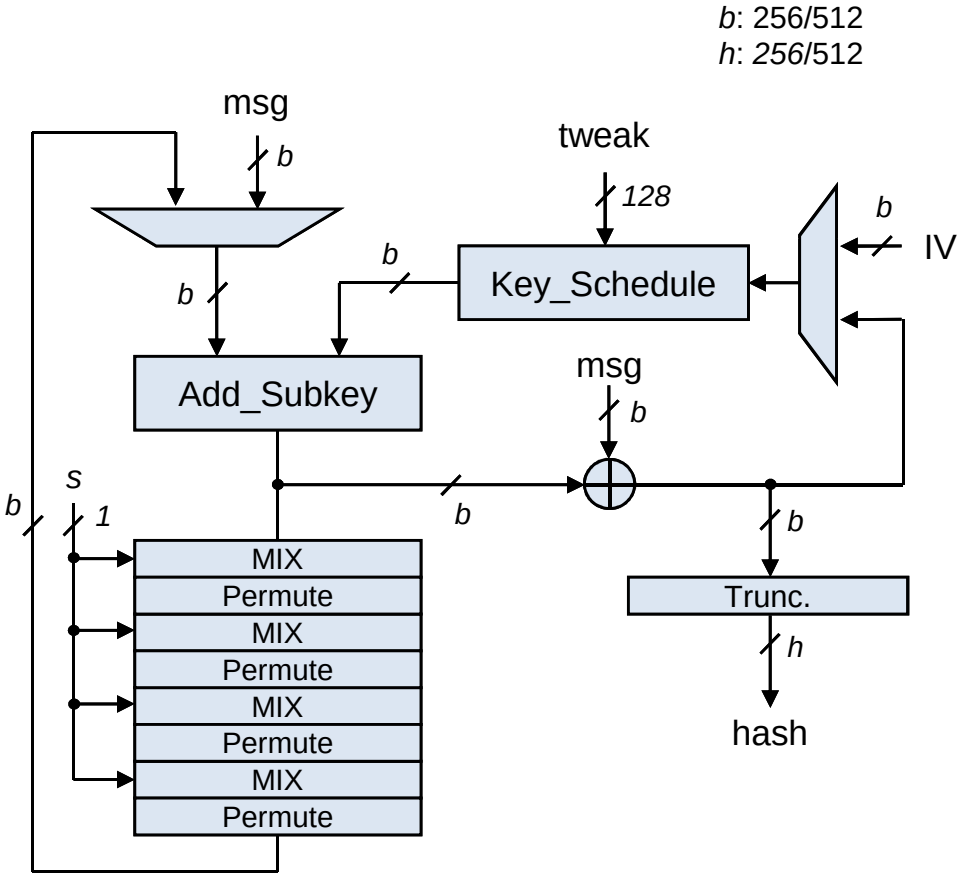


$O = i_0 \oplus i_1$

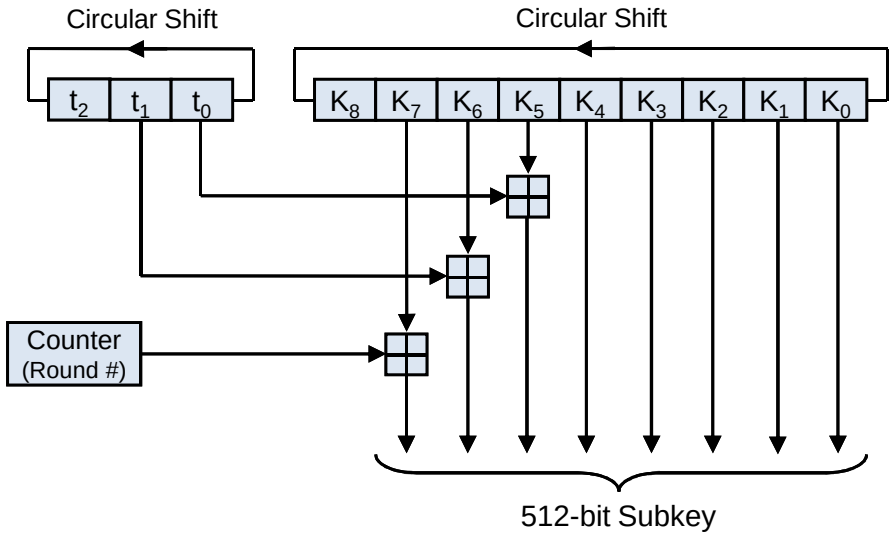
2-bit XOR used in i step

IMPLEMENTATION

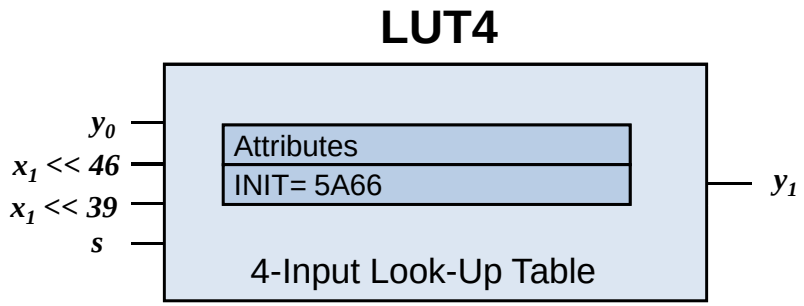
- Data path Architecture for Skein



(a) Data path of Skein



(b) Key_Schedule Module



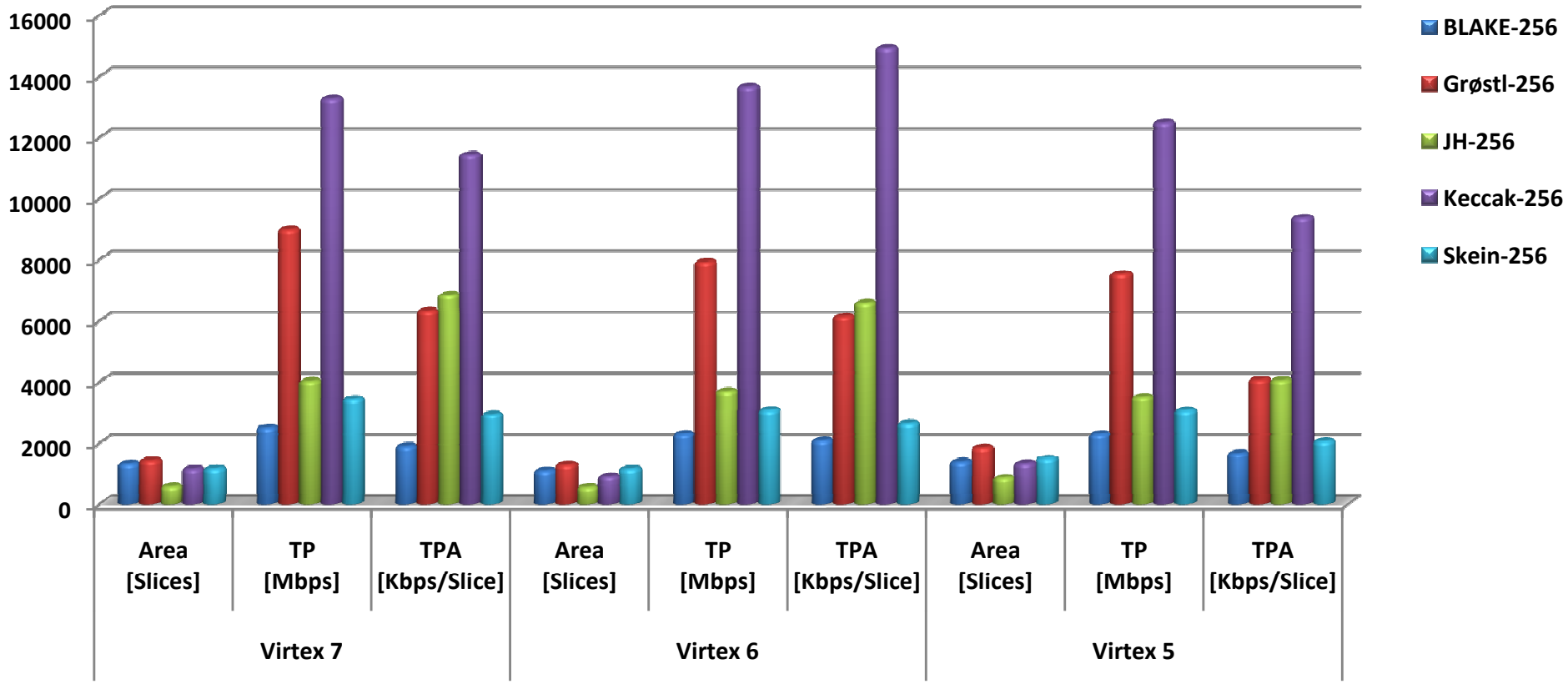
$y_1 = s\ ?\ ((x_1 << 46) \oplus y_0) : ((x_1 << 46) \oplus y_0)$

(c) Selection between two rotation constants in MIX operation

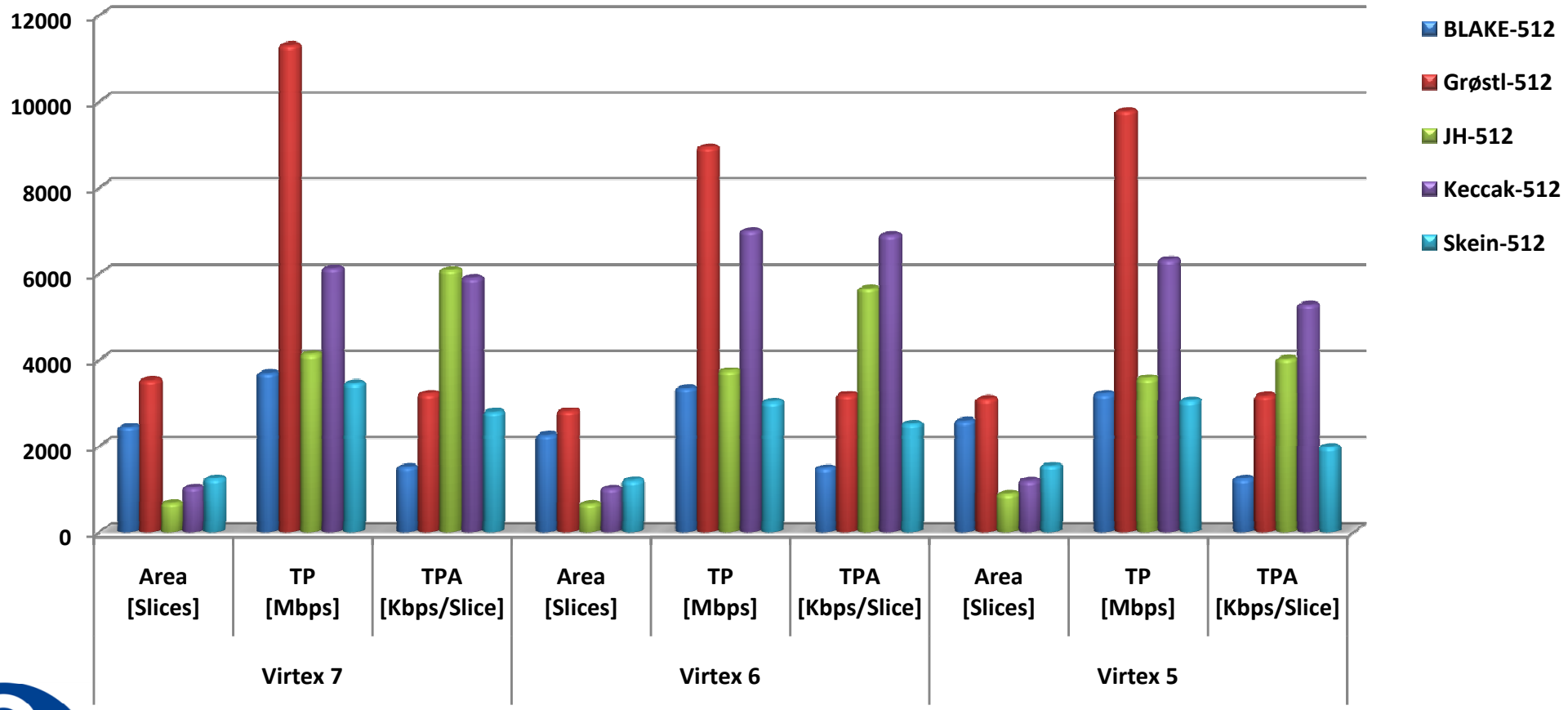
RESULTS

SHA-3 Finalist	Device	256-bit						512-bit					
		Block Size [bits]	N_{clk} [cycles]	F_{max} [MHz]	*Area* [Slices]	*TP* [Gb/s]	*TPA* [Mbps/slice]	*Block Size* [bits]	N_{clk} [cycles]	F_{max} [MHz]	*Area* [Slices]	*TP* [Gb/s]	*TPA* [Mbps/slice]
BLAKE	Virtex 5	512	28	125.55	1382	2.29	1.66	1024	32	100.02	2582	3.21	1.24
	Virtex 6	512	28	125.82	1104	2.30	2.08	1024	32	104.30	2246	3.34	1.46
	Virtex 7	512	28	137.14	1322	2.51	1.90	1024	32	115.01	2441	3.68	1.51
Grøstl	Virtex 5	512	10	121.03	1419	6.20	4.37	1024	14	101.22	2523	7.40	2.94
	Virtex 6	512	10	146.87	1467	9.62	5.12	1024	14	125.44	2359	9.17	3.89
	Virtex 7	512	10	175.65	1421	8.99	6.33	1024	14	155.02	3524	11.3	3.23
JH	Virtex 5	512	42	287.44	865	3.50	4.05	512	42	292.48	888	3.57	4.02
	Virtex 6	512	42	303.65	562	3.70	6.59	512	42	306.37	661	3.74	5.65
	Virtex 7	512	42	329.49	587	4.02	6.84	512	42	338.41	679	4.13	6.08
Keccak	Virtex 5	1088	24	275.56	1333	12.49	9.37	576	24	263.16	1197	6.32	5.28
	Virtex 6	1088	24	301.57	915	13.67	14.94	576	24	291.21	1015	6.99	6.89
	Virtex 7	1088	24	292.74	1161	13.27	11.43	576	24	254.91	1039	6.12	5.88
Skein	Virtex 5	512	19	113.78	1492	3.07	2.05	512	19	113.60	1544	3.06	1.98
	Virtex 6	512	19	114.30	1163	3.08	2.65	512	19	112.36	1203	3.03	2.52
	Virtex 7	512	19	127.73	1170	3.44	2.94	512	19	128.24	1244	3.46	2.78

PERFORMANCE COMPARISON OF 256-bit VARIANTS OF SHA-3 FINALISTS



PERFORMANCE COMPARISON OF 512-bit VARIANTS OF SHA-3 FINALISTS



PERFORMANCE RANKING

Rank	256-bit		512-bit	
	TPA	***TP***	***TPA***	***TP***
1	Keccak	Keccak	Keccak	Grøstl
2	JH	Grøstl	JH	Keccak
3	Grøstl	JH	Grøstl	JH
4	Skein	Skein	Skein	BLAKE
5	BLAKE	BLAKE	BLAKE	Skein

CONCLUSION

- We have presented efficient and high throughput implementations of SHA-3 finalists
- Results shown for Xilinx Virtex 5, Virtex 6 and Virtex 7
- Performance figures reported in terms of Area consumption, throughput and throughput per area
- Performance comparison on latest Xilinx FPGAs is presented
- This work serves as performance investigation of SHA-3 finalists on most up-to-date FPGAs

Question & Answers

Q/A