Implementation of the Compression Function for Selected SHA-3 Candidates on FPGA

Ashkan H. Namin and M. A. Hasan

Department of Electrical and Computer Engineering

UNIVERSITY OF Waterloo

Outline

- Introduction to cryptographic hash functions
  - Definition of hash functions and the most common, SHA-1
  - NIST SHA-3 Competition
- Selected SHA-3 Candidates
  - Blue Midnight Wish hash function
  - Luffa hash function
  - Skein hash function
  - Shabal hash function
  - Blake hash function
  - Comparison of implementations
- Conclusions

Introduction to cryptographic hash functions

- A mathematical function that maps an **arbitrary input** message size into a message digest output **of fixed size.**

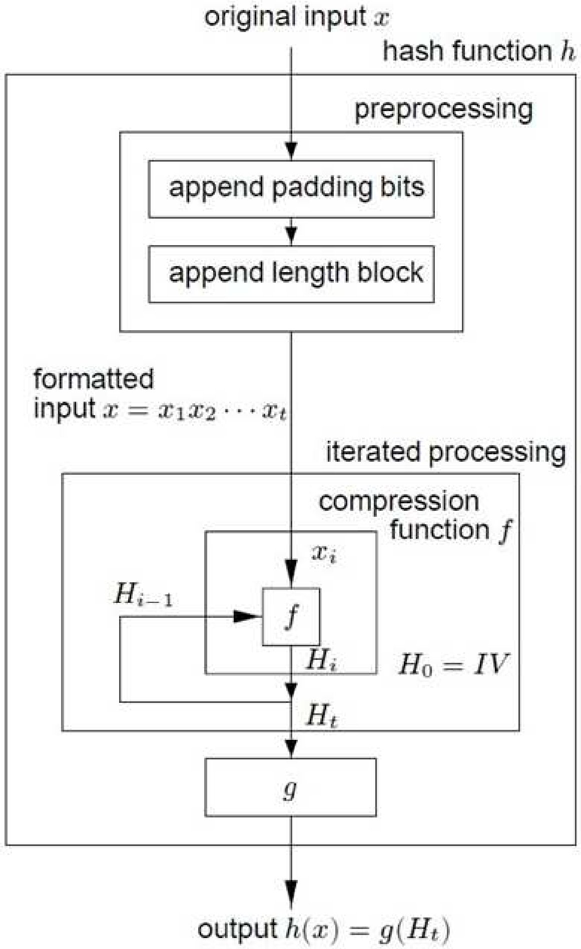
- Secure hash function properties
  - One-way: hard to determine the message of the message digest
  - Collision resistant: hard to find two messages having the same message digest

- Applications: Message Authentication Codes (MAC), digital signature, fingerprinting, checksum, password verification, …

- Protocols: SSL, SSH, TLS, IPSEC

Iterative Hash Functions

- Most hash functions are designed as an iterative processes through a two stage architecture
  - Preprocessing stage
    - Padding the arbitrary input to appropriate size
    - Padding the message length to the message
    - breaking down the message into blocks of smaller fixed sizes (256 or 512 bit size)
  - Hash computation stage
    - Use the compression function iteratively to create the message digest



Secure Hash Algorithm (SHA)

- A set of functions designed by the NSA and published by the NIST as a U.S. federal standard.

- SHA-1 (160 bit message digest)
  - Most widely used hash function
  - Originally published in 1993 (SHA-0) and revised in 1995.
  - A weakness was announced in Feb. 2005 by Dr. Wang.

- SHA-2 (SHA-224, SHA-256, SHA-384, SHA-512)
  - Originally published in 2001 and revised in 2002 and 2004.
  - Architecturally similar to SHA-1, higher security level than SHA-1
  - NIST recommend transition from SHA-1 to SHA-2

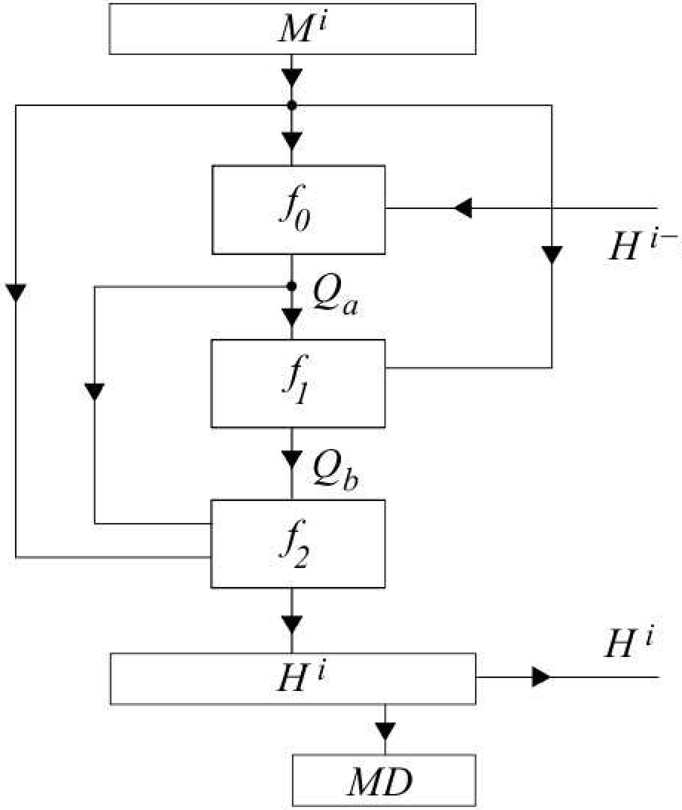
- SHA-3
  - NIST opened a public competition to develop SHA-3 in 2008.

SHA-3 Public Competition

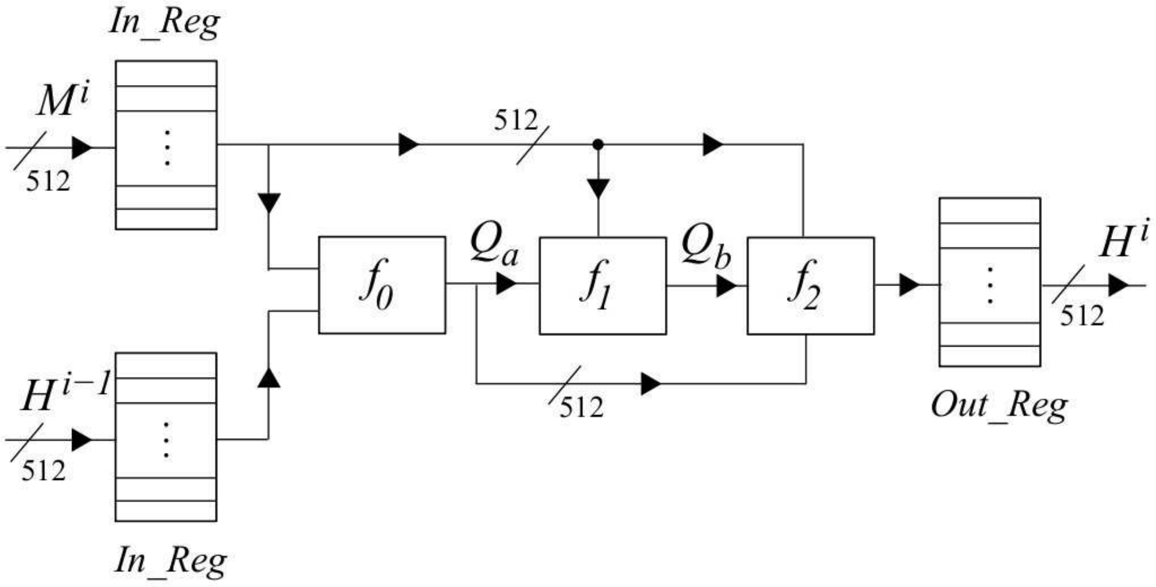
- Candidates
  - 51 candidates in Round-1 (Nov. 2008), 14 candidates in Round-2 (July 2009).
- Software Implementation
  - All candidates submission contains C code of the algorithm
  - A complete list of candidates software performance is on EBASH website
- Hardware Implementation
  - Limited number of candidates submission include hardware implementation
  - Different platform and technologies were used, hard to compare
- Motivation behind this work
  - Implement a group of candidates using the same technology and design approach
  - Selected Candidates: Blue Midnight Wish, Luffa, Skein, Shabal, and Blake
  - Focus on the 256 bit version, using Stratix III FPGA

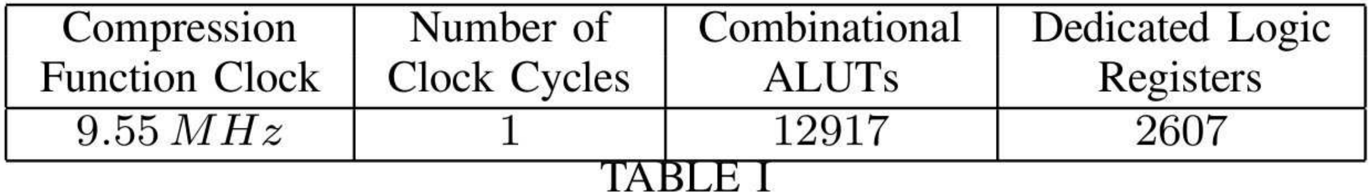
Blue Midnight Wish Compression Function

- Proposed by Svein Johan Knapskog et al. from Norwegian University of Science and Technology (NUST)
- Uses a wide-pipe (double-pipe) hash construction
- Nonlinearity is derived from the overlap of modular addition (2^{32}) and XOR
- There exist a practical near-collision attack against the design



Blue Midnight Wish FPGA Implementation





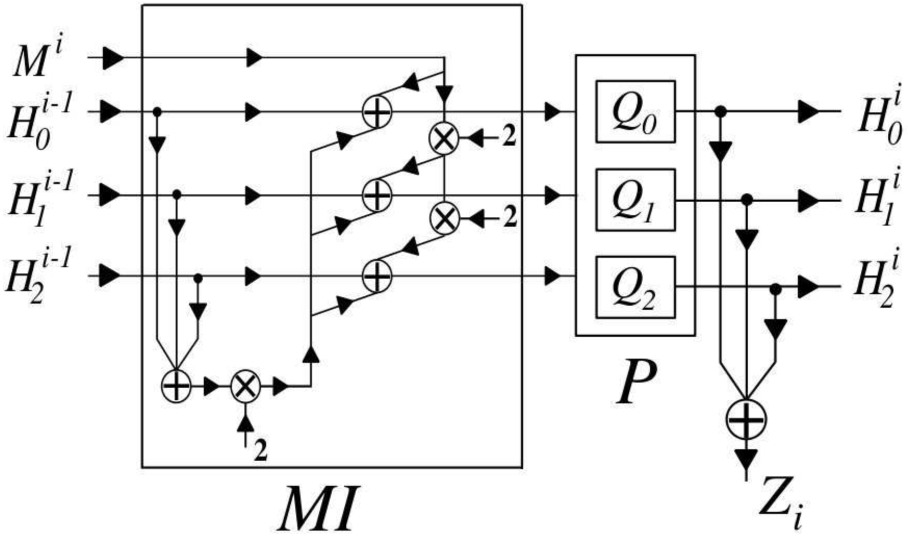
Compression Function Clock	Number of Clock Cycles	Combinational ALUTs	Dedicated Logic Registers
9.55 MHz	1	12917	2607

TABLE I

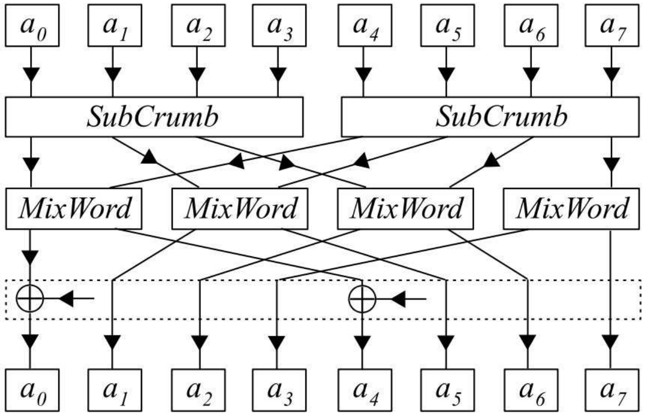
FPGA IMPLEMENTATION SUMMARY OF THE BMW-256 COMPRESSION FUNCTION

Luffa Compression Function

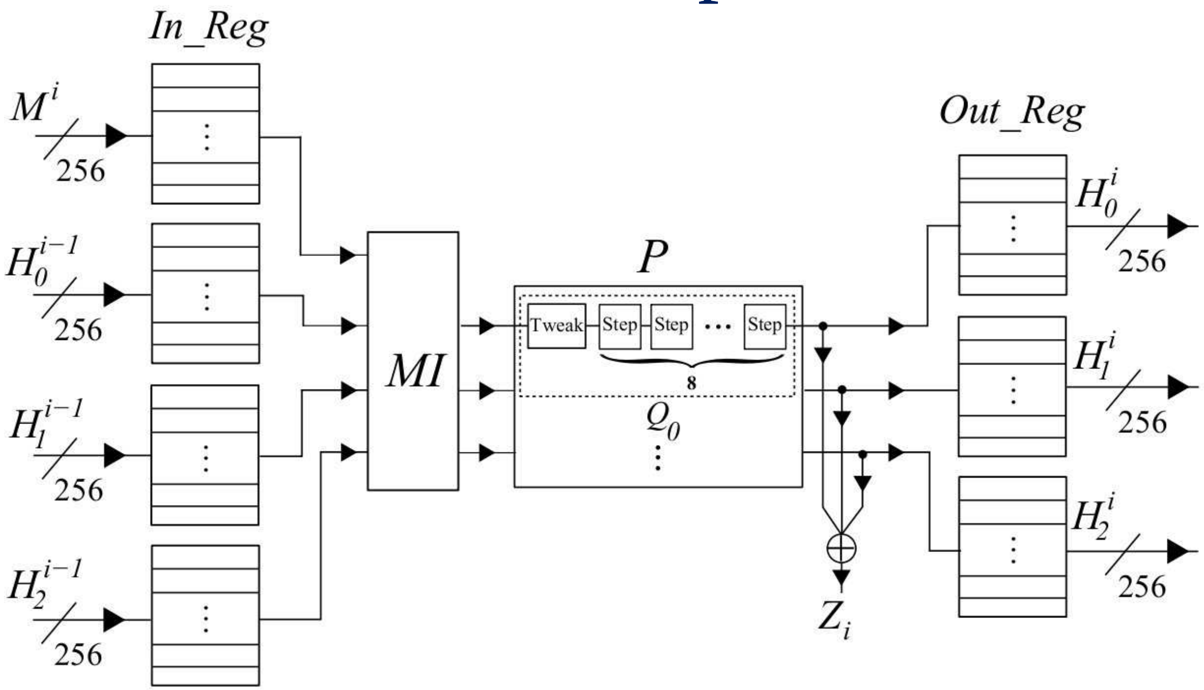
- Proposed by Dai Watanabe et al. from Hitachi company.
- MI and P (3Q) main modules, Each Q is made of 8 Step functions
- makes use of s-boxes/nonlinear permutations in SubCrumb and Shift and XOR in MixWord
- No threat to security has been reported

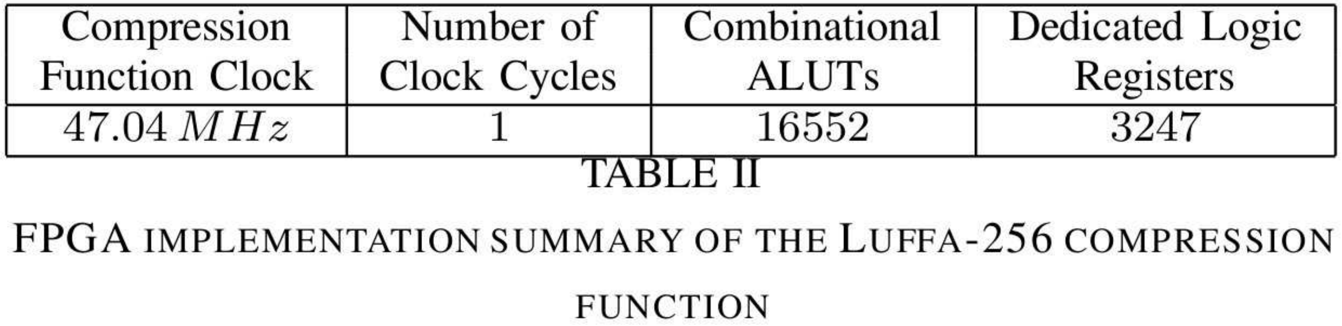


Step Function


a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7
SubCrumb
SubCrumb
MixWord
MixWord
MixWord
MixWord
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7

Luffa FPGA Implementation





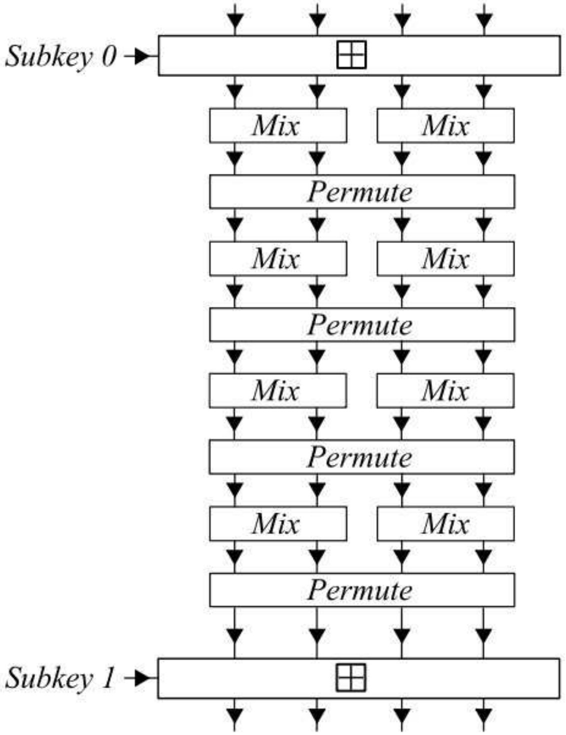
Compression Function Clock	Number of Clock Cycles	Combinational ALUTs	Dedicated Logic Registers
47.04 MHz	1	16552	3247

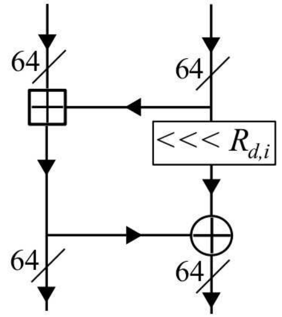
TABLE II

FPGA IMPLEMENTATION SUMMARY OF THE LUFFA-256 COMPRESSION FUNCTION

Skein Compression Function

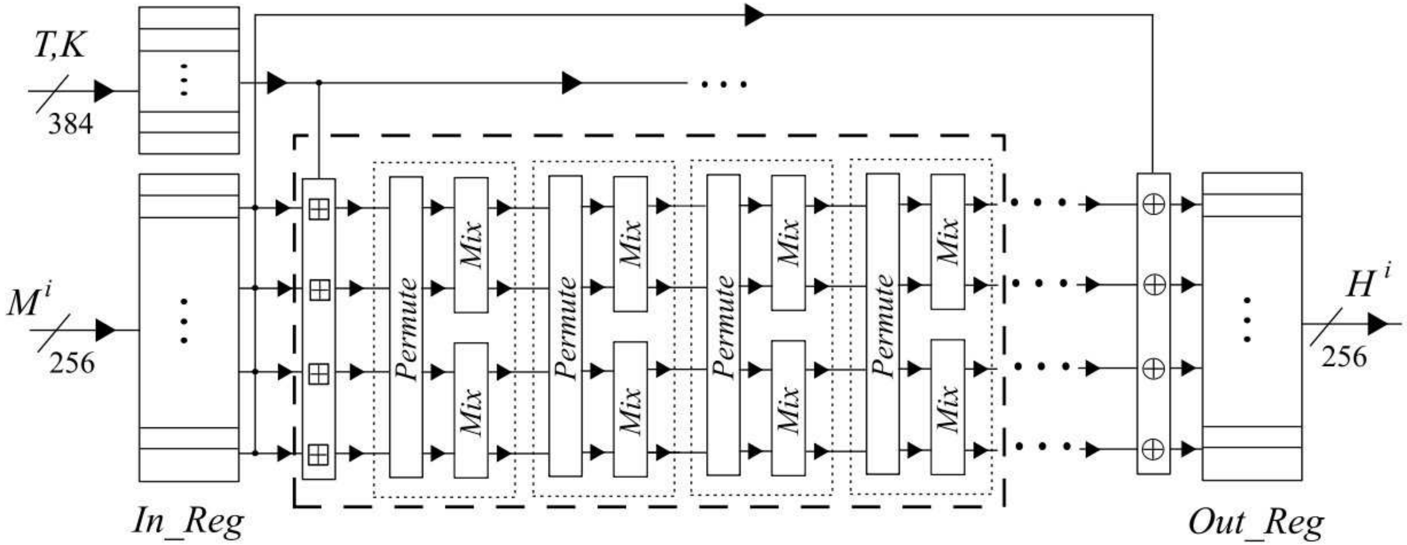
- Proposed by Niels Ferguson et al. from Microsoft and Intel companies.
- Based on Thrrefish block cipher, use large number (72) of simple rounds, highly regular architecture
- Nonlinearity is derived from the overlap of modular addition (2^{64}) and Rotation and XOR
- No threat to security has been reported





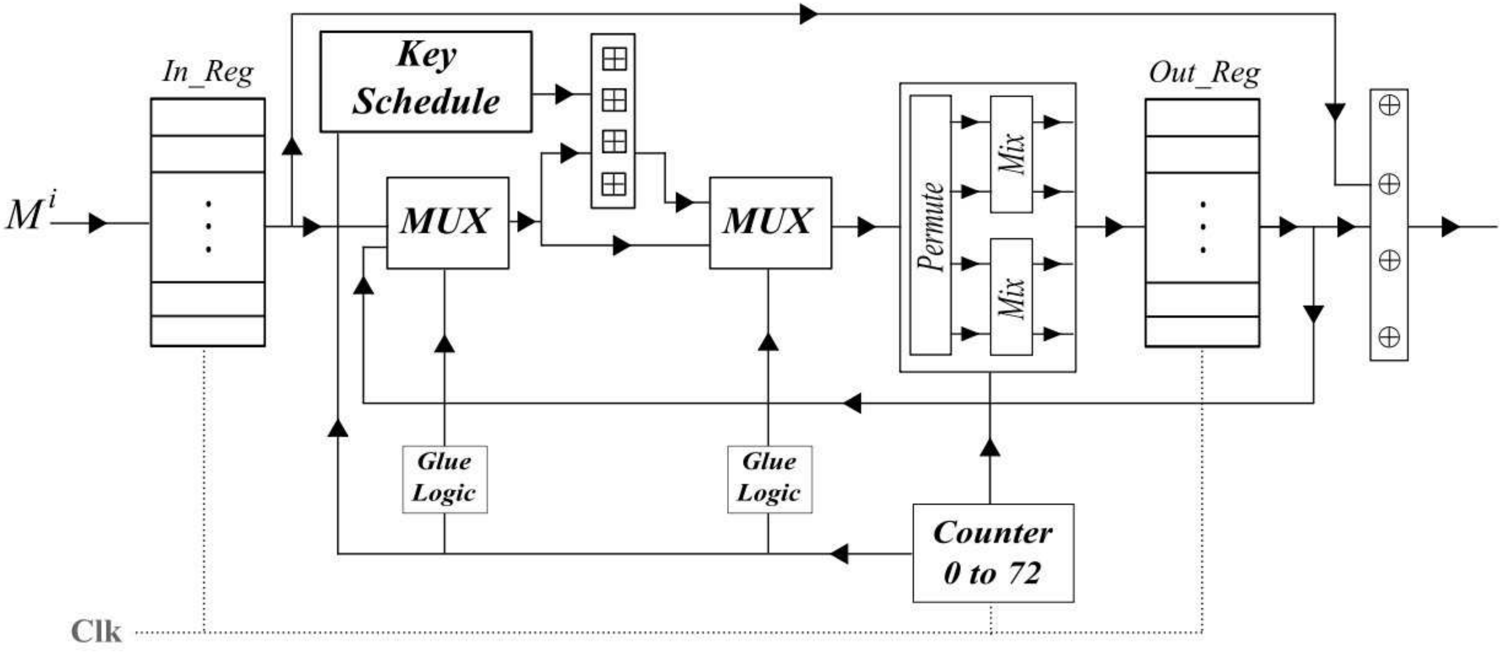
Mix Block

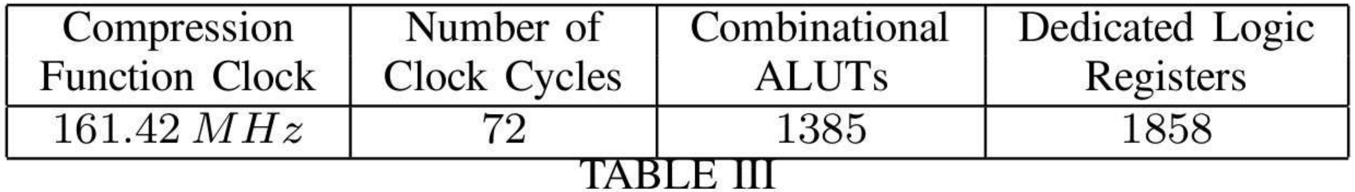
Skein FPGA Implementation



- Full parallel version did not fit on FPGA (Skein makes use of a large number of 64-bit adders (over 200)
- A second version of the Skein algorithm (Skein-1c) using just one round of the Threefish (Two Mix, One permute).

Skein FPGA Implementation





Compression Function Clock	Number of Clock Cycles	Combinational ALUTs	Dedicated Logic Registers
161.42 MHz	72	1385	1858

TABLE III

FPGA IMPLEMENTATION SUMMARY OF THE SKEIN-1C COMPRESSION FUNCTION

Shabal Compression Function

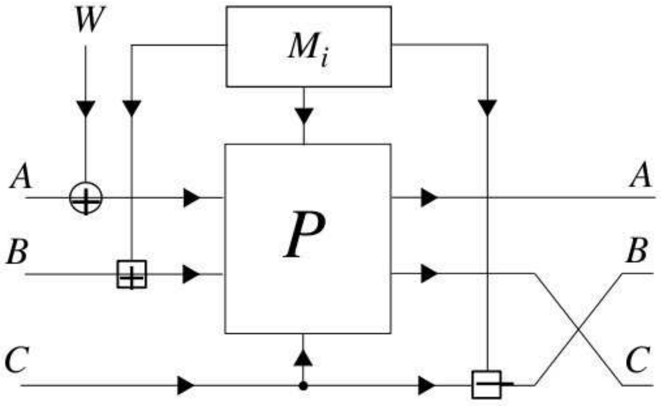
- Proposed by Emmanuel Bresson et al., from French research agency and France Telecom.

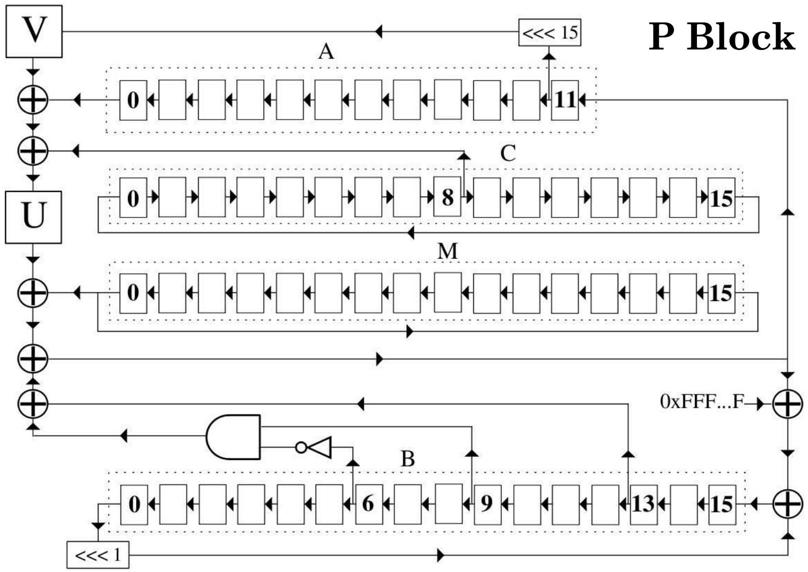
- Based on feedback shift registers

- Nonlinearity is derived from overlap of XOR, AND, and modular addition (2^{32})

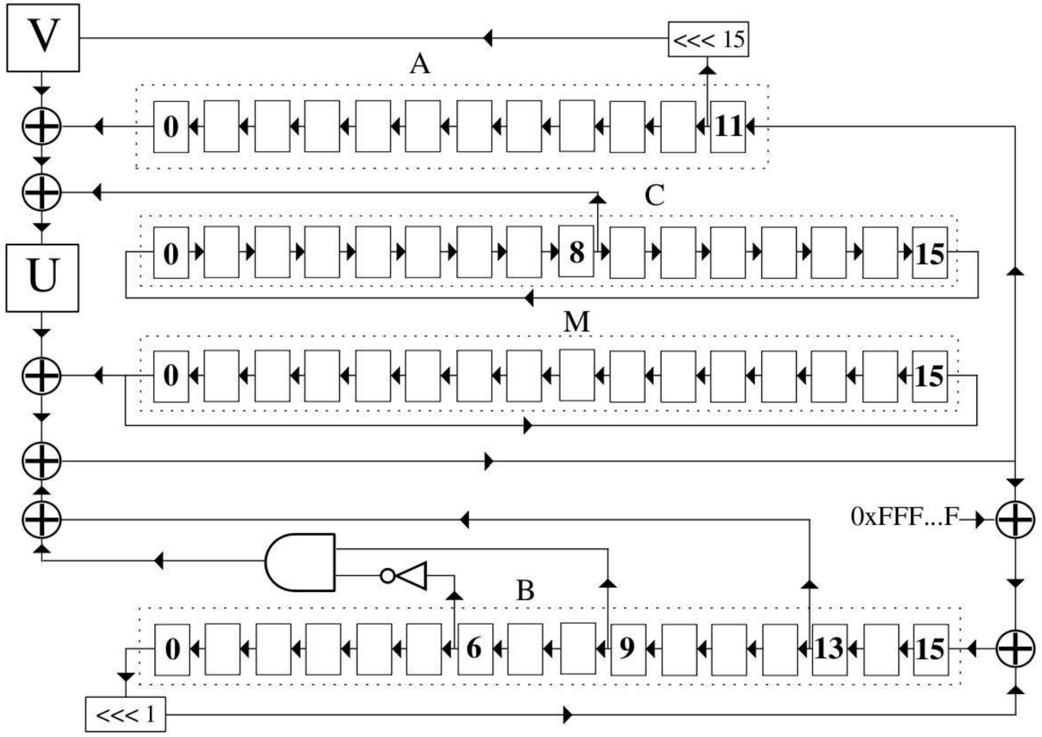
- Requires more memory than other candidates

- Some attacks have been reported against it





Shabal FPGA Implementation



Compression Function Clock	Number of Clock Cycles	Combinational ALUTs	Dedicated Logic Registers
195.35 MHz	48	1440	4000

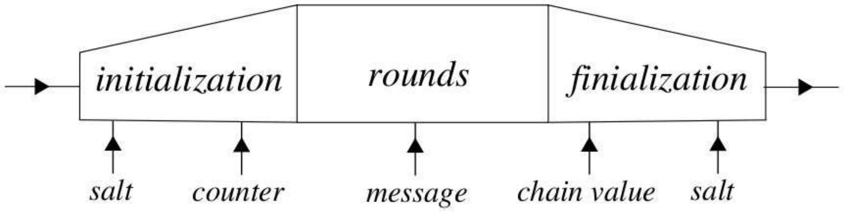
TABLE IV

FPGA IMPLEMENTATION SUMMARY OF THE SHABAL-256 COMPRESSION FUNCTION

Blake Compression Function

- Proposed by Jean-Philippe Aumasson et al., from FHNW, ETHZ Switzerland universities

- Compression function is a modified version of ChaCha stream cipher

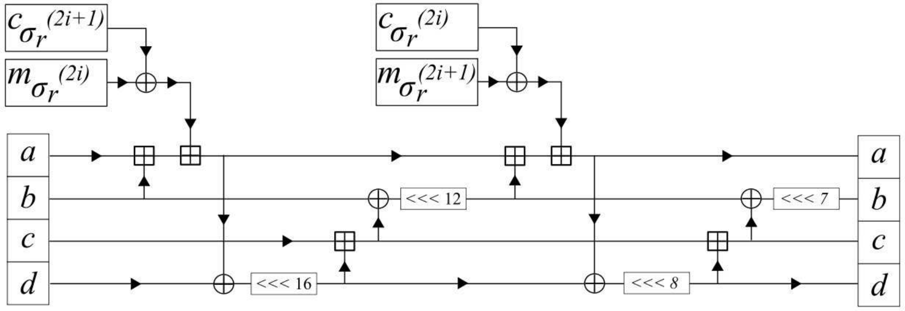


- Uses a wide-pipe structure, 10 rounds each has 8 consecutive G functions

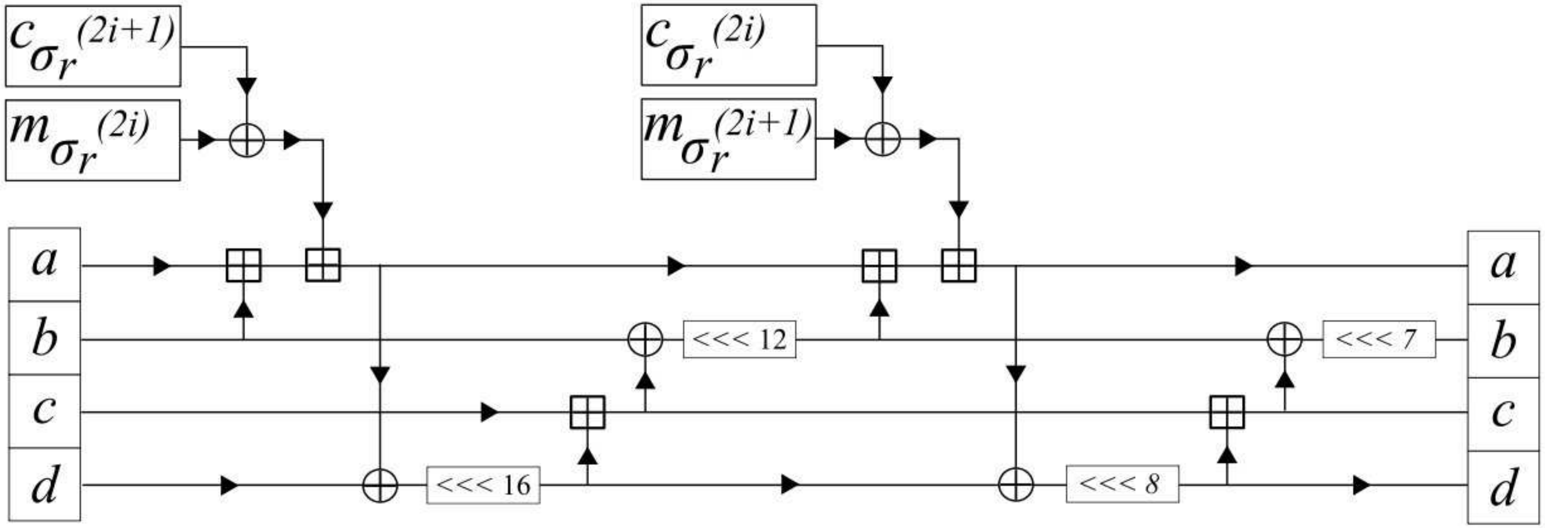
- Nonlinearity is derived by use of modular addition(2^{32}) and XOR

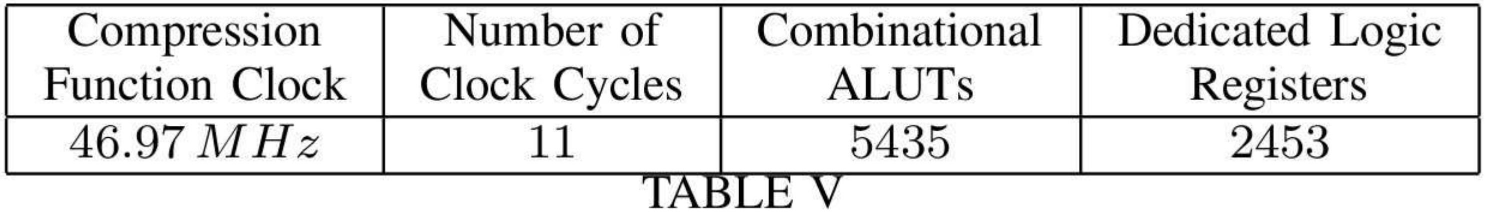
- No security threat has been reported

G Function Block
C: Constants, m: message blocks



Blake FPGA Implementation



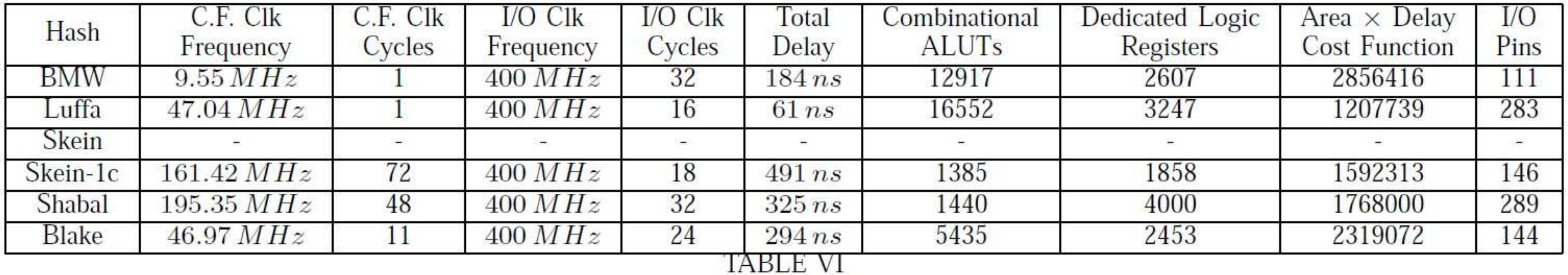


Compression Function Clock	Number of Clock Cycles	Combinational ALUTs	Dedicated Logic Registers
46.97 MHz	11	5435	2453

TABLE V

FPGA IMPLEMENTATION SUMMARY OF THE BLAKE-256 COMPRESSION FUNCTION

Comparison of Implementtaions



Hash	C.F. Clk Frequency	C.F. Clk Cycles	I/O Clk Frequency	I/O Clk Cycles	Total Delay	Combinational ALUTs	Dedicated Logic Registers	Area × Delay Cost Function	I/O Pins
BMW	9.55 MHz	1	400 MHz	32	184 ns	12917	2607	2856416	111
Luffa	47.04 MHz	1	400 MHz	16	61 ns	16552	3247	1207739	283
Skein	-	-	-	-	-	-	-	-	-
Skein-1c	161.42 MHz	72	400 MHz	18	491 ns	1385	1858	1592313	146
Shabal	195.35 MHz	48	400 MHz	32	325 ns	1440	4000	1768000	289
Blake	46.97 MHz	11	400 MHz	24	294 ns	5435	2453	2319072	144

TABLE VI

FPGA IMPLEMENTATION SUMMARY OF THE DIFFERENT COMPRESSION FUNCTIONS

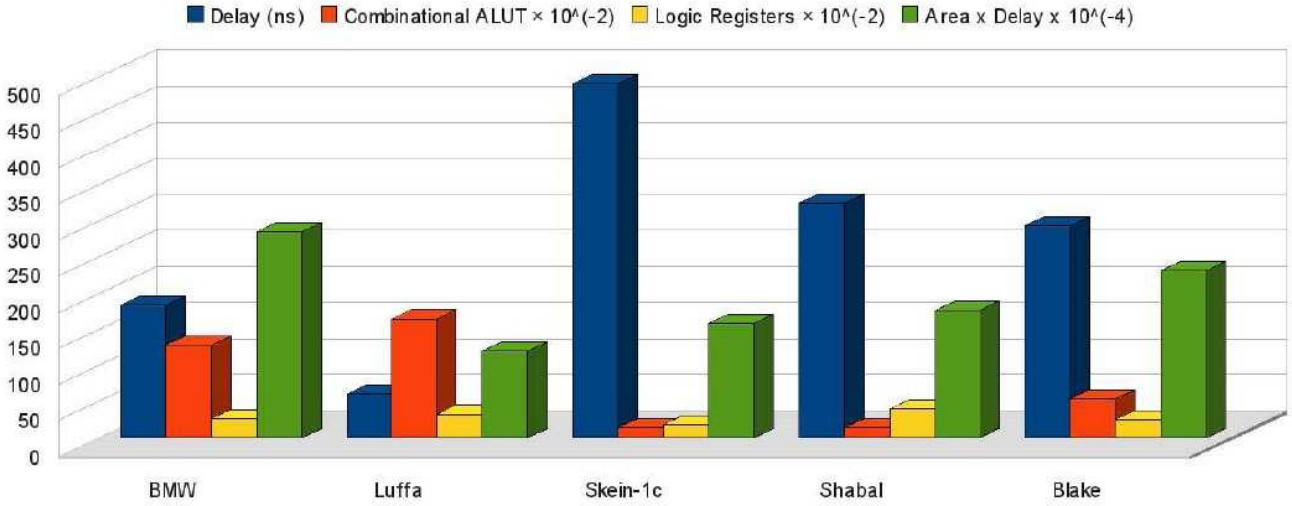


Fig. 19. Area delay complexities for FPGA implementation

Conclusions

- We have presented hardware implementation of five SHA-3 candidates using FPGA
- A fair comparison of hardware performance of the candidates is possible
- Among our candidates Luffa and Skein outperform other candidates in terms of Area x Delay