RESEARCH ARTICLE OPEN ACCESS

The Skien Hash Function

Sivasankari. S. A[1], Allan Mary George [2], Pavithra. S [3]
[1]Assistant Professor, [2]Assistant Professor, [3]Assistant Professor
Department of ECE, Saveetha School of Engineering, Saveetha University, Saveetha Nagar, Thandalam, Chennai-602 105, Tamilnadu, India

ABSTRACT
In this paper, we simulate the performance of skein hash function in VHDL FPGA. Skein is a new cryptographic family. It is designed in such a way that it is simple, easy and flexible. This function which is implemented in software based. This is a hashing function which provides security.

Keywords: Skein hash function, Threefish block cipher, MIX function, Key schedule, unique block iteration (UBI).

I. Introduction

The new family of cryptographic hash function is skein. The advantages of this hash function are speed, security, simplicity and flexibility due to its design as well as easy to analyse. Skein hash function is fast and secure Skein has three different internal state sizes as 256, 512, 1024 bits and output may be any size. It is a good replacement of SHA family. Skein is an efficient tool to be used for large number of functions due to its optional and extendable argument system [1]. Skein is efficient for both hardware and software platforms.The remainder of this paper is organized as follows:

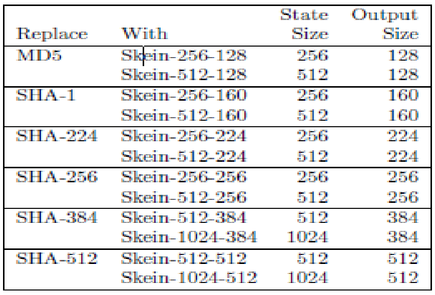
Section II presents the Basic structure of skien, Section III discusses about the skein algorithm, and Section IV draws the conclusion.

II. Basic structure of Skein

2.1. Overview of Skein

Skein is a new family hash function with three internal state sizes: 1024,512,256 bits.

- Skein- 512 which is used for all present hashing applications with secures and remains secure for the future.
- Skien-1024 is ultra conservative variant [1]. Skein-1024 is the twice of skein- 512. Skein 1024 is twice as fast as Skein- 512.
- Skien-256 is low memory variant [1].
  The above discussed internal state size which can support any size of output.
  Replacement of one type of hash function with other type of hash function shown in Table: 1[1]



Replace	With	State Size	Output Size
MD5	Skein-256-128	256	128
	Skein-512-128	512	128
SHA-1	Skein-256-160	256	160
	Skein-512-160	512	160
SHA-224	Skein-256-224	256	224
	Skein-512-224	512	224
SHA-256	Skein-256-256	256	256
	Skein-512-256	512	256
SHA-384	Skein-512-384	512	384
	Skein-1024-384	1024	384
SHA-512	Skein-512-512	512	512
	Skein-1024-512	1024	512

Table: 1: Replacements for MD5, SHA-1 and SHA-2

2.2. Basic building blocks

Three new components are needed to build Skien.They are three fish block cipher, Unique block iteration (UBI), Optional argument system.

➢ **Three fish Block cipher**

It is a tweakable block cipher at the Skein's Core with any internal size as 256, 512, 1024 bits.

➢ **UBI- Unique Block iteration**

This is the type of chaining mode which uses three fish in order to build the compression function which maps any input size to any or fixed output size.

➢ **Optional argument system**

This makes skein which supports any optional Features.

2.2.1. Threefish Block cipher

Threefish is a large tweakable block cipher [66].It is also defined for three different block sizes as 256, 512, 1024 bits. The key which should be same size as the block and the tweak value is only 128 bits which is applicable for all the three different block sizes.

The main principle of Threefish is that more number of simple rounds which are secured than few numbers of complex rounds. There are three different mathematical operations used by Threefish block cipher are Exclusive-OR (XOR), Addition and constant rotation.

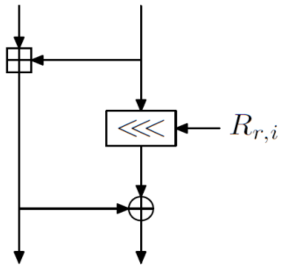


Fig: 1. MIX Function

The function in three fish which is actually a MIX function. **Fig: 1[1].** Each function which consists of a single addition, a rotation by one constant, an Exclusive OR function.

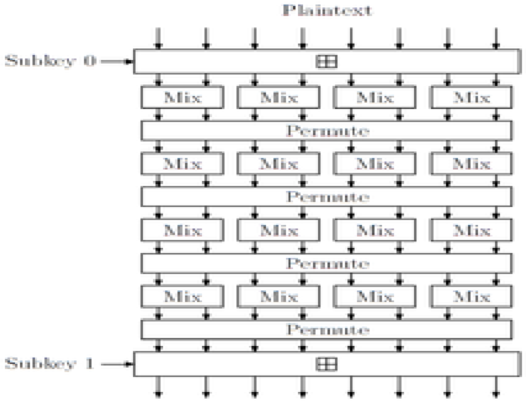


Fig: 2 .Three fish-512.

Figure.2 which shows the operation of MUX function to build threefish-512.Each skien-512 has 72 rounds which has four mix functions which gets followed by a permutation of eight 64 bit words. Each function made on 64 bit words. At every four rounds one sub key will be generated. The word permutation is same for every round. The rotations here are constant which are chosen in such a way to maximize diffusion [1].

These rotations are constant for every eight rounds. The sub keys are generated from the key schedule. The key schedule generates the sub keys by use of key and tweak. Each sub key contributes key words, tweak words and counter value. To create a key schedule, key and tweak words which are extended with one parity word extra which is XOR of all the words .For threefish-256 there are 72 rounds and 2 MIX functions followed by permutation.

For threefish-1024 the rounds are 80 and have 8 MIX functions. The rotation constants and permutations are different for different internal Block sizes. But it should maximize the diffusion.

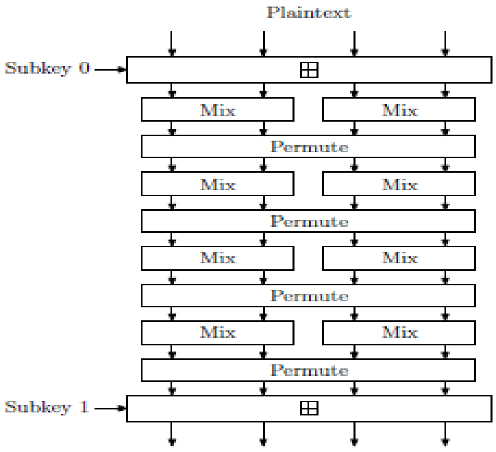


Fig: 3.Threefish-256

2.2.2. Unique Block Iteration (UBI)

It is a chaining mode which produces fixed output by using input chaining value and arbitrary length input string. The tweak is the heart of UBI .Every block is processed with unique variant of the compression function. UBI chaining mode which ensures the above work. UBI which follows Matyas-Meyer-Oseas [2] hash mode. This hash mode which follows the message input to hash function should have to be same as the plain text input to the block cipher.

2.2.3. Optional Arguments

To increase the flexibility of Skein, Some Optional Inputs can be enabled. The key which should process first to achieve the security. But is optional. KEY:

CONFIGURATION:

It is required in order to encode the desired output length.

PERSONALIZATION:

It is used to create different functions for different users.

PUBLIC KEY:

When hashing a message for signing it is must to public key.

KEY DERIVATION IDENTIFIER:

It is used for key derivation. To derive a key, master key should be given as input key and the requested identifier will get the key.

NONCE:

It is particularly used in Stream cipher mode and randomized hashing [1].

MESSAGE:

It is the normal message given as the Input to the hash function.

OUTPUT:

It is the one which is required and retrieved. But the main advantage of this hashing function is that whenever in need it is possible to add any function which is standardized here.

III. Skein Hash architecture

It has different types of architecture. It will face either in Hardware or software mode. The skein hash algorithm's structured and its different types of architecture will be discussed here.

3.1. Algorithm

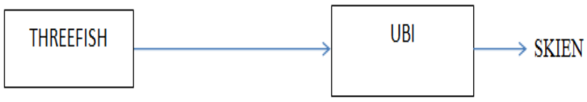


Fig: 4.Stick diagram

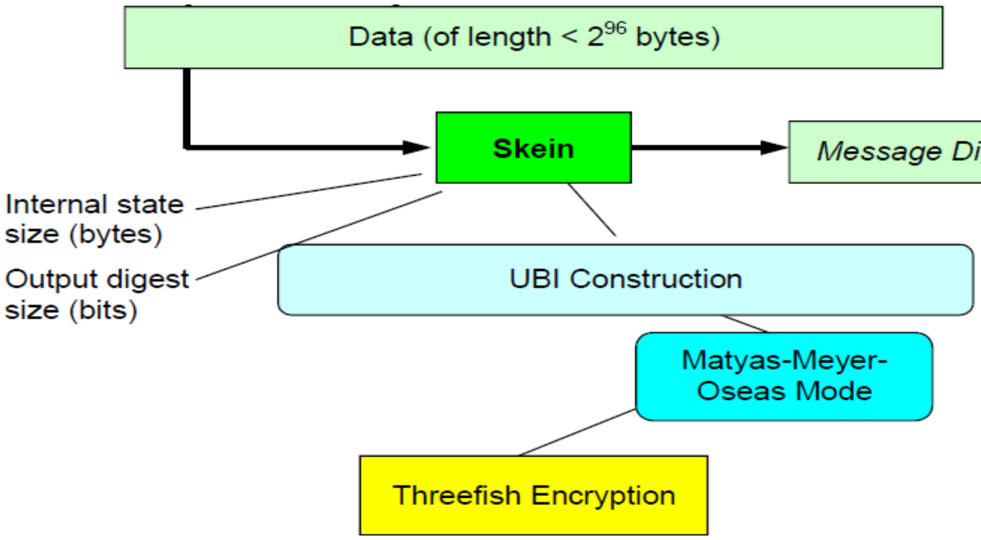

Data (of length < 2^96 bytes)
Skein
Message Digest
Internal state size (bytes)
Output digest size (bits)
UBI Construction
Matyas-Meyer-Oseas Mode
Threefish Encryption

Fig: 5.Basic Block Algorithm

3.2. Architecture

This is the architecture which is meant and defined for rounding. There are three different types as Iterative round, Loop unrolling and Pipeline architecture.

3.2.1. Iterative Round architecture

One round of skein algorithm is implemented in the round circuit.Skien-512 requires the 72 number of rounds. Hence 72 times of iteration is needed to complete one hardware round for skein. The round operation not only carried by input. But also based on key, Tweak and round Number to produce a round key. The result is stored in register. If it is final output it gets XORed with input which forwards to UBI. If it is not final result then it will pass to loop till 72 rounds. Each rotation will carry over by one clock cycle exactly. If there are n number of rounds then n clock cycles are require.

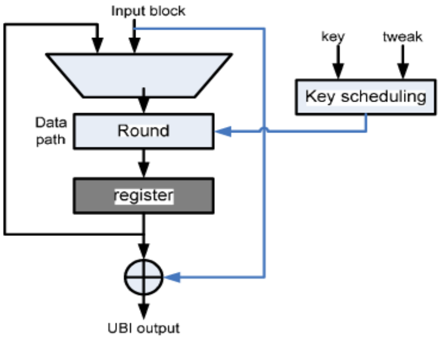


Fig.6.Iterative round architecture

3.2.2. Loop unrolling architecture

The main different of this architecture when compared with iterative is here the exact number of rounds have to place before registering the output. The area requirement is more due to more number of rounds placed one after another. The throughput is also low and hence lowers efficiency.

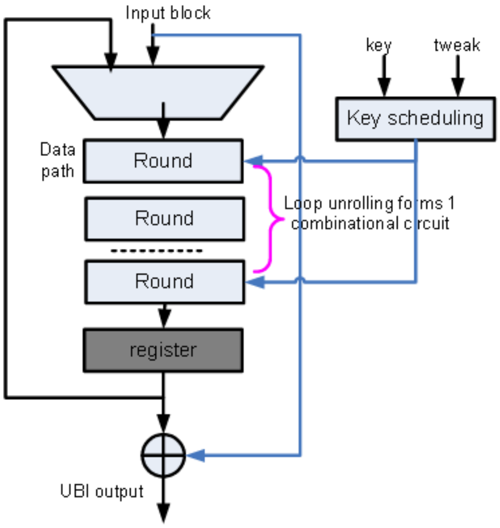


Fig:7. Loop unrolling architecture

3.2.3. Pipeline architecture

The main advantage of pipeline architecture is increasing throughput.The advantageous feature of pipeline architecture is that each and every consecutive rounds have one register.

Skien is the type of hashing algorithm which enforces data serialization.Hashing block of n meaasge which cannot continue if n-1 value is not there.The drawback of this is that it is ideal for some rounds and hence inefficient utilization of hardware resources

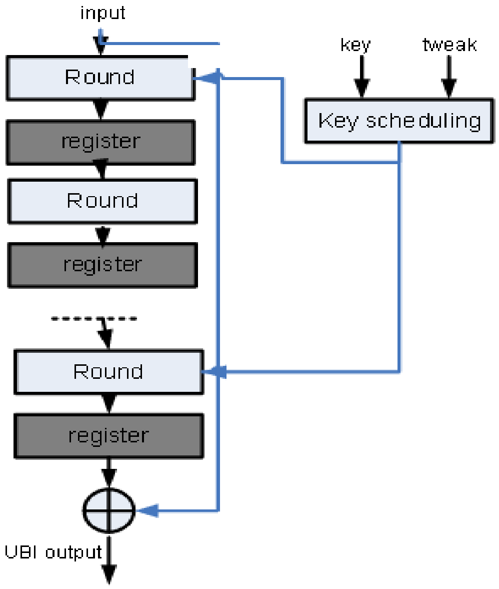
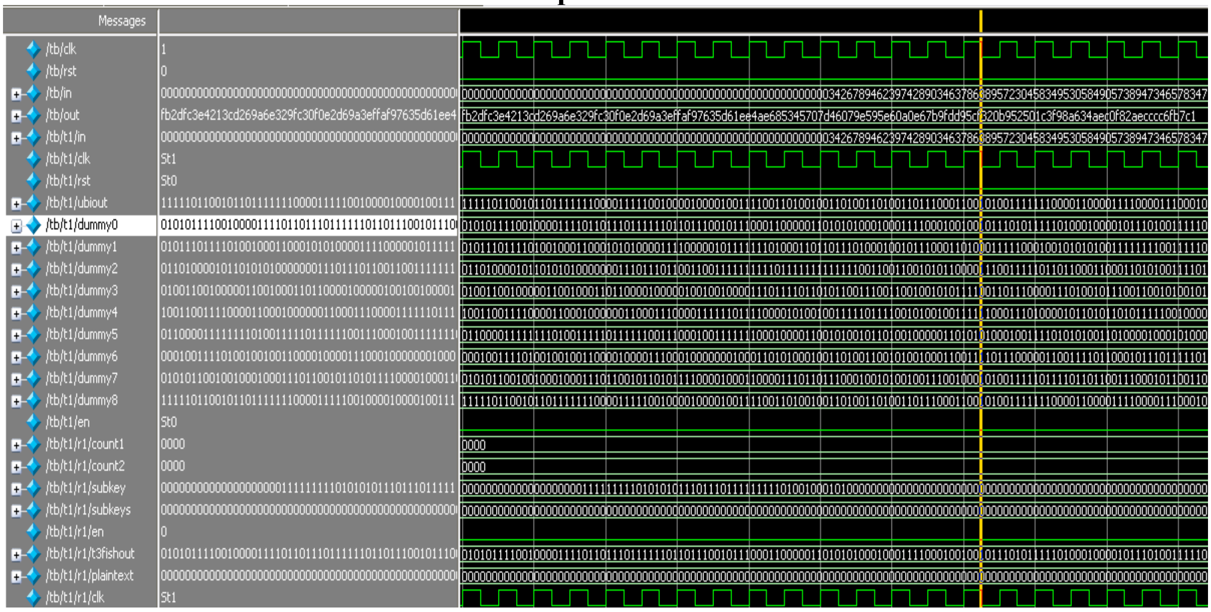


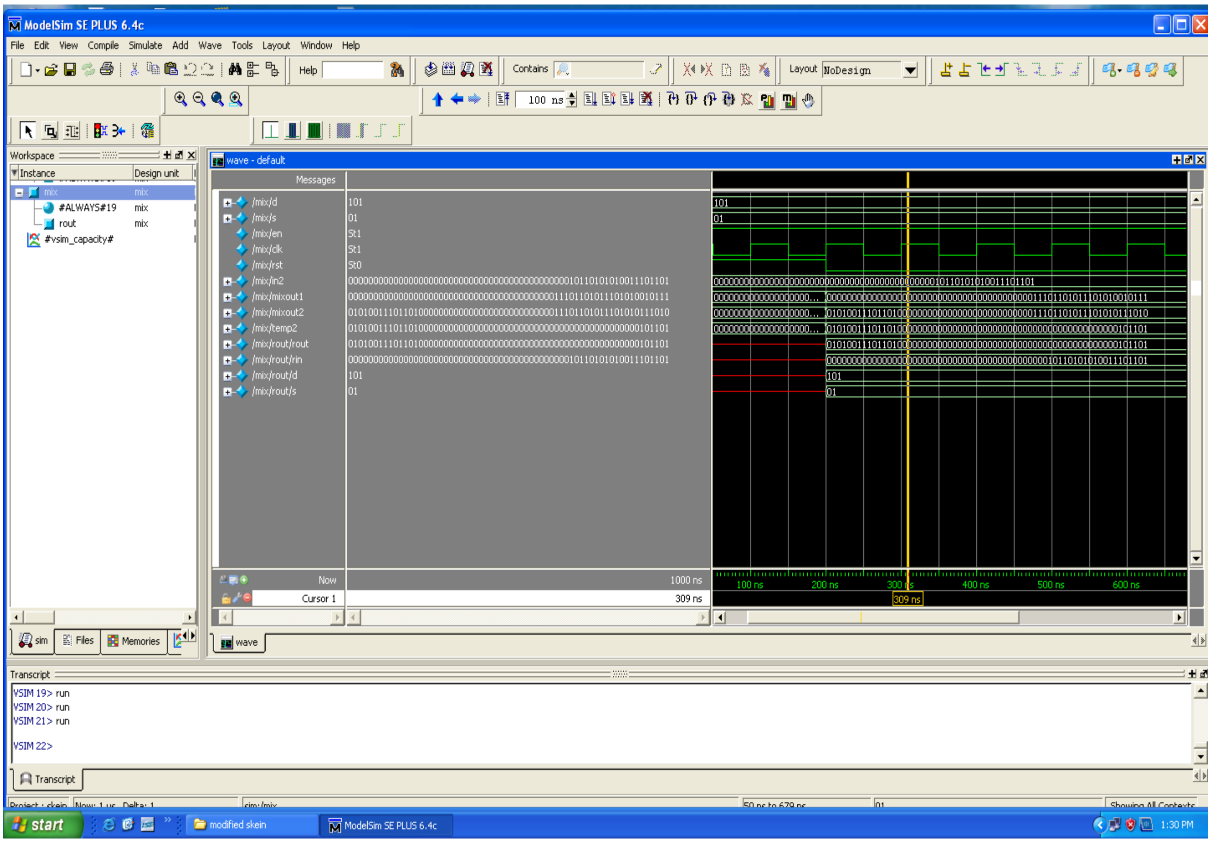
Fig:8. Pipeline architecture

IV. Experimental results



```
/tb/clk              1
/tb/rst              0
/tb/in               0000000000000000000000000000000000000000000000000
/tb/out              fb2dfc3e4213cd269a6e329fc30f0e2d69a3effaf97635d61ee4
/tb/t1/in            0000000000000000000000000000000000000000000000000
/tb/t1/clk           St1
/tb/t1/rst           St0
/tb/t1/ubiout        1111101100101101111111000011111001000010000100111
/tb/t1/dummy0        0101011110010000111101101110111111011011100101110
/tb/t1/dummy1        0101110111101001000110001010100001111000001011111
/tb/t1/dummy2        0110100001011010101000000011101110110011001111111
/tb/t1/dummy3        0100110010000011001000110110000100000100100100001
/tb/t1/dummy4        1001100111100001100010000001100011100001111110111
/tb/t1/dummy5        0110000111111110100111110111111001110001001111111
/tb/t1/dummy6        0001001111010010010011000010000111000100000001000
/tb/t1/dummy7        0101011001001000100011101100101101011110000100011
/tb/t1/dummy8        1111101100101101111111000011111001000010000100111
/tb/t1/en            St0
/tb/t1/r1/count1     0000
/tb/t1/r1/count2     0000
/tb/t1/r1/subkey     0000000000000000000111111110101010111011101111 1
/tb/t1/r1/subkeys    0000000000000000000000000000000000000000000000000
/tb/t1/r1/en         0
/tb/t1/r1/t3fishout  0101011110010000111101101110111111011011100101110
/tb/t1/r1/plaintext  0000000000000000000000000000000000000000000000000
/tb/t1/r1/clk        St1
```

MIX FUNCTION

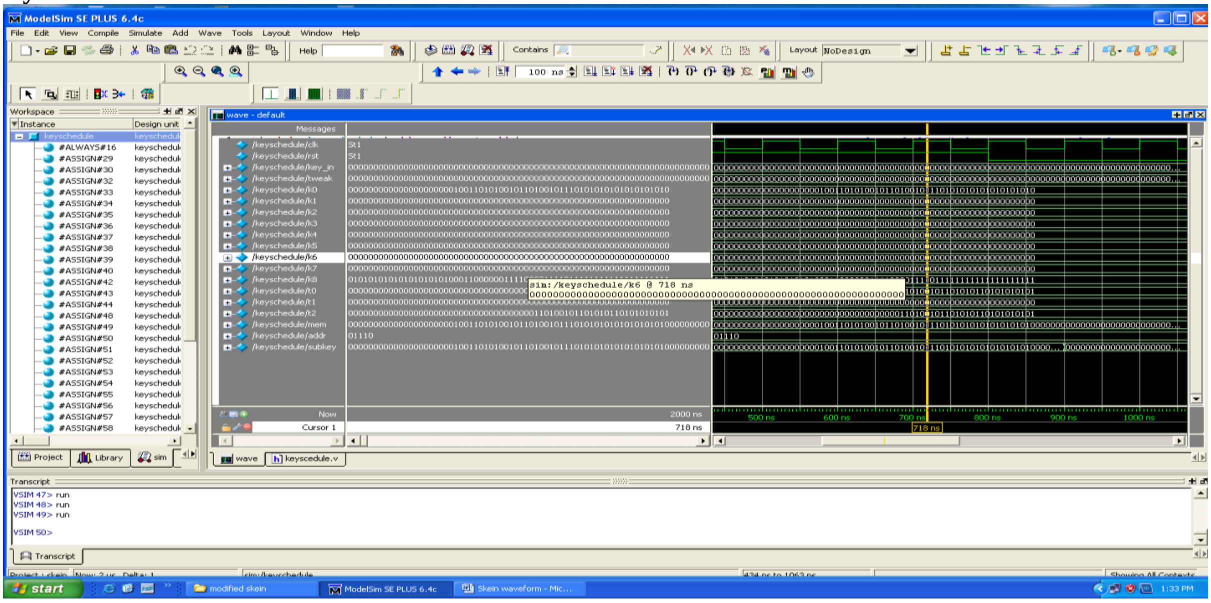


```
wave - default
Messages
/mix/d           101
/mix/s           01
/mix/en          St1
/mix/clk         St1
/mix/rst         St0
/mix/in2         000000000000000000000000000000000000000000000101101010100111011 01
/mix/mixout1     0000000000000000000000000000000000000000001110110101110101001011 1
/mix/mixout2     010100111011010000000000000000000000000000111011010111010101110 10
/mix/temp2       010100111011010000000000000000000000000000000000000000000101101
/mix/rout/rout   010100111011010000000000000000000000000000000000000000000101101
/mix/rout/rin    000000000000000000000000000000000000000000000101101010100111011 01
/mix/rout/d      101
/mix/rout/s      01
Now              1000 ns
Cursor 1         309 ns

Transcript
VSIM 19> run
VSIM 20> run
VSIM 21> run

VSIM 22>
```

Key schedule



```
wave - default
Messages
/keyschedule/clk      St1
/keyschedule/rst      St1
/keyschedule/key_in   0000000000000000000000000000000000000000000000000000000000000000
/keyschedule/tweak    0000000000000000000000000000000000000000000000000000000000000000
/keyschedule/k0       000000000000000000000100110101001011010010111010101010101010 10
/keyschedule/k1       0000000000000000000000000000000000000000000000000000000000000
/keyschedule/k2       0000000000000000000000000000000000000000000000000000000000000
/keyschedule/k3       0000000000000000000000000000000000000000000000000000000000000
/keyschedule/k4       0000000000000000000000000000000000000000000000000000000000000
/keyschedule/k5       0000000000000000000000000000000000000000000000000000000000000
/keyschedule/k6       0000000000000000000000000000000000000000000000000000000000000
/keyschedule/k7       0000000000000000000000000000000000000000000000000000000000000
/keyschedule/k8       01010101010101010101000110000001111
/keyschedule/t0       000000000000000000000000000000000000
/keyschedule/t1       0000000000000000000000000000000000000000000000000000000000000
/keyschedule/t2       000000000000000000000000000000000000110100101101010110101010101
/keyschedule/mem      00000000000000000000010011010100101101001011101010101010101010000000000
/keyschedule/addr     01110
/keyschedule/subkey   00000000000000000000010011010100101101001011101010101010101010000000000
sim:/keyschedule/k6 @ 718 ns
Now                   2000 ns
Cursor 1              718 ns

Transcript
VSIM 47> run
VSIM 48> run
VSIM 49> run

VSIM 50>
```

V. Conclusion

Thus in this paper which I discussed the basic building blocks of skein hash algorithm and its architecture with detail manner. Skein has three different internal state sizes as 256, 512, 1024 bits and output may be any size. It is a good replacement of SHA family. Skein is an efficient tool to be used for large number of functions due to its optional and extendable argument system [1]. Skein is efficient for both hardware and software platforms.

VI. ACKNOWLEDGEMENTS

Jesse Walker provided explanations of Skein hash as well as suggestions for optimizing the hardware implementation. I thank Doug Whiting and Niels Ferguson for their Presentations on the Threefish round implementations. I thanked Dr.Neduncheliar Head of the department of computer science engineering who reviewed my paper and

Dr.P.C.Kishoreraja Head of the department of electronics and communication engineering who encouraged me to do this work.

References

[1] ”The skein hash function family”, version 1.3, 0ct 2010, Niels Ferguson, Stefan Lucks, Bruce Schneier,Doug Whiting,Mihir Bellare,Tadayoshi Kohno,Jon Callas, Jesse Walke

[2] “Implementing Skein Hash Function on Xilinx Virtex-5 FPGA Platform “Men Long, Intel Corporation, 02-Feb-09,Version 0.7 men.long@intel.com

Authors

I am S.A.Sivasankari working as assistant professor ECE Department in Saveetha School of engineering. I completed my UG from Maharaja Engineering College Bharatiyar University and PG from SRM university in VLSI. Currently I am doing my research work in Transmission line effects in interconnects in Deep submicron VLSI circuits.

I am Allan Mary George; I finished my bachelor degree in Biomedical Engineering and masters in Applied Electronics .currently working as Assistant Professor in Saveetha School Of Engineering.

Myself S.Pavithra working as assistant professor ECE Department in Saveetha School of engineering and received my UG in ECE from KCG Engineering College and PG from Muthukumaran institute of engineering and technology in Applied electronics.