INTERNATIONAL RESEARCH JOURNAL ON ADVANCED SCIENCE HUB
e-ISSN : 2582 - 4376
Open Access
RSP SCIENCE HUB
(The Hub of Research Ideas)
Available online at www.rspsciencehub.com

Special Issue of Second International Conference on Advancements in Research and Development (ICARD 2021)

A 128- Bit of Md5 Algorithm with 16 Stages of P&Eline Using Unfolding Design

Ambika. N[1], Dr. T. Menaka Devi[2]
[1]Student of Adhiyamman College of Engineering (Autonomous), Hosur, India.
[2]Professorof Adhiyamman CoI1ege of Engineering (Autonomous), Hosur. India.
ambikanagappa6@gmail.com[1], drmenakadevit@gmail.com[2]

Abstract

We defined the hash algorithm as a security object during data transmission. The hash features are 4 more types 1. Message digest (MDS) 2. RIPMED (Race Integrated primitivity of Message Digest) 3. SHA (Secured Hash Function i) 4. WHIRLFOOL. From this group of hashes, Message Digest 5 is mostly using in embeddedsecuritysystems. To obtain high frequency and throughput in regions of Area, frequency and throughput by the unfolding transformation technique factor of 2. To get better outcomes to overcome the dis-advantages of existing method, the proposed design is reduced the data fetching stages from 32 to 16 stages. However, the overall process is speedy than the existing algorithm.

Keywords: MD5, Hash function, Throughput. Message Digest.

1. Introduction

In hash function we kept input as message and the output gain is described as hash value and H is located between the arbitrary and the output. The merit of hash length is credit a value by itself. The above mentioned hash function features are examine here. The first one is Message Digest (MD5), it is 128- bit function and its application in software world. The next type is RIPMED, it's a 128-bit, 160- bit function and also in 256-bit and 320-bit. Third one is SHA, it's a 160 bit function. Last one is Whirlpool, it's a 512 bit function and has 3 versions namely WHIRLPOOL-O, T, WHIRLPOOL.

1.1 Design Procedure of Hash function

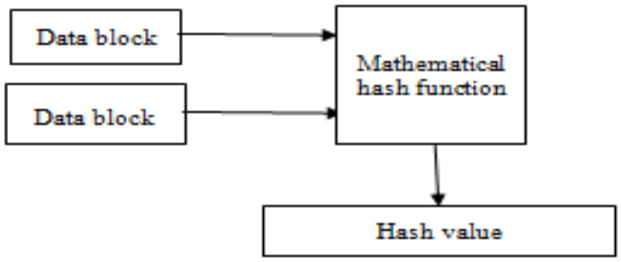


Fig 1: Designing process of hash algorithm

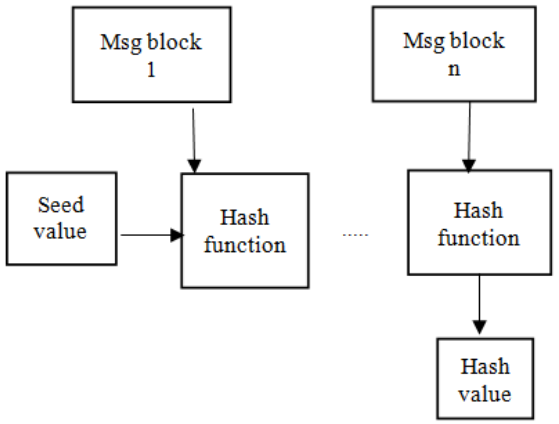


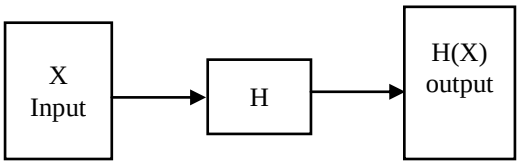
Fig 2: Schematic diagram of hash function

The entire algorithm depends on hash coding, we need a 2 fixed size data to build code. The duration of each block of data varies in accordance with the type of algorithms. And the size of block changes from the 128 bit to 512 bit. From the architecture of turbo decoder we can operate the central Add analyse Select (ACS) activity. Because of the parallel processing the ACS blocks have lower number of preparing steps, so we gain less amount

of transmission energy and less multifaceted nature about 71 percentage. The proposed work throughput is 1.03 Mb/s, and the memory necessity of proposed design is 128.8 Kbps, the unpredictability is decreased by four percentage and the force utilization is diminished by 32%. By OFDM, the VLSI architecture was implemented.

1.2 Features of hash function:

The common features of this algorithm outputs are fixed length and next is efficiency of operation. Here, each feature performs various operations in an algorithm design. One is hashing the data which means covert the given data from message length to fixed length. Further process is compressing the function is explained as the functions are moved to be compressed. MD5 is refered as a small representation hash function. And final one is generating N bits of hash. Apart from the first feature, this is more systematic in performing the operation and the provide result is quit rapid than the symmetric encryption .[1-4]



From the shown diagram representation we can learn the hash properties namely pre-image, second pre-image and collision resistance. The three properties depends on that same input and try to find our different values with same hash algorithm. The earliest property of this function says that it's impossible to discover any other arbitrary value that matches the existing value. The second property says that if this property gives an input with hash value then it should be heavy process to search another input that matches the previous hash value. And final property simply says that it take huge time to seeking of two arbitrary values with equal length of data.

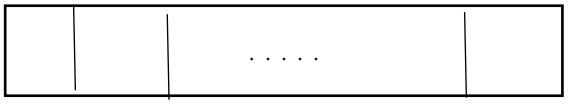
2. Algorithm design:

The algorithm is broadly usage hash function which produces one hundred and twenty-eight bit hash. In pre-processing, the design start by padding messages in a specializing format of little endian. By attaching the input message with include by adding 1 bit at the fixed range of message duration. The processes in continued until the 0 bit is filled with message. In 128-bits, the sixty-four bits are resolved for period of arbitrary message.

Step 1: Padding of bit's

And the remaining sixty-four bits message length also in little endian format. So, the arbitrary message total length is 512 bits. Here, after completing the message padding, the second method hash computation take place.

The total 512 bits are equally breakdown into 16 block, each block carries 32 bit.



M [0], M [1], ……., M [15]

Step 2: Buffer initialization of md5

The entire algorithm is divided from 128 bits into 32 bit words, and they are A, B, C, D. Here, I'm using four possible functions with different Non-linear function use for a each round. The four buffer registers are defined as

A= 08 ab 32 ef

B= 98 b dc f4

C= fe dc ba 98

D= 76 54 32 10

Step 3: Processing the blocks

The MD family has four Non-linear functions namely F, G, H, I .Each Non-linear functions has 16 rounds and the entire algorithm has sixty-four to reap 128 bits of hash code. The bunch of rules in MD5 has sixty-four rounds of operations. While F denotes Non-linear function, and only one characteristic is utilizing in every round. M_i indicates a thirty-two bit of input message and K_i designate a thirty-two bit of constant, and it's miles exceptional for each round of operation.$<<_s$ denotes left shift rotation of value s, it changes for every round of operation by itself.$\oplus$ indicates a addition modulo of $2^{32.}$

The four possible functions are accept input as a 32 bit word and provide output as also as a 32 bit and establish the logical functions AND, OR, NOT and XOR the input values are applied.

F(X, Y, Z) = XY v not(X) Z

G(X, Y, Z) = XY v Y not Z

H(X, Y, Z) = X xor Y xor Z

I(X, Y, Z) = Y xor (X v not(X) Z)

The auxiliary functions of X, Y, Z are self-sufficient and at a same time balanced. In case the F(X, Y, Z) are self-sufficient and balanced means the rest of three functions also same independent and balanced performed parallel process vice-versa process.

In accordance with the independent and balanced function F (X, Y, Z), G(X, Y, Z), H(X, Y, Z) and I(X, Y, Z) are balanced and independent without any external control. [5-8].

The six module names are ABCD_init, input_ABCD, MD5_initial, func_process, getdata and MD5_hash.The MD5 architecture is displayed below.

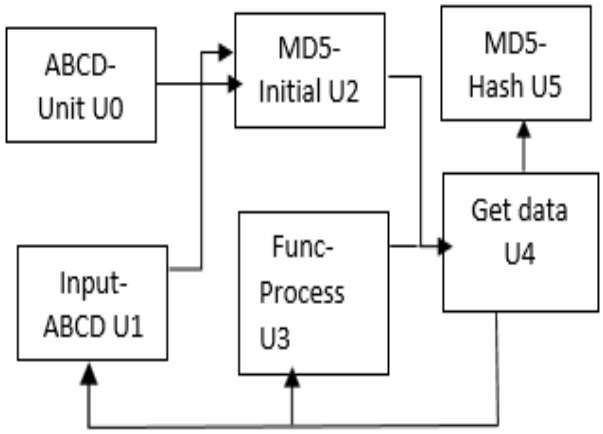


Fig 3: Top level unfolding of MD5

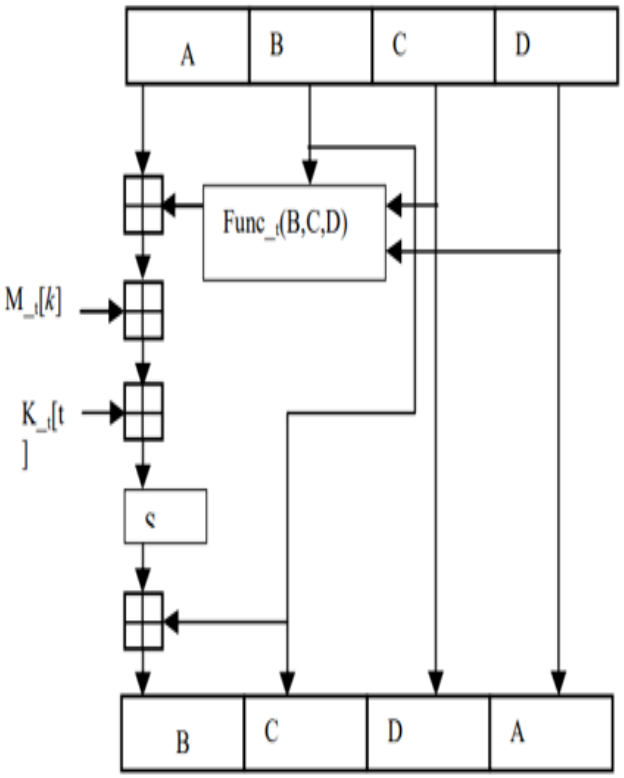


Fig 4: Compression function of MD5

The MD5 is a bunch of rules which is nearly new for message processing. The message of 512-bits of blocks which are performed separately. This algorithm made up of 4 iterations, one and all rounds are comprises with 16 steps. The performance of algorithm as follows: first, values of A, B, C, D are stored into temporary variables. Then, every stage of operations are carry out of i = 0 -63 as follows:

$$B_{i+1} \leftarrow B_i + ((R(B_i, C_i, D_i) + A_i + T_{i+1} + M_j[i+1]) <<< S_{i+1})$$
$$A_{i+1} \leftarrow D_i, C_{i+1} \leftarrow B_i, D_{i+1} \leftarrow C_i$$

However, the temporary variables are aggregate to the obtained values from the algorithm, and the results are reserved in the registers A, B, C, D. After processing allmessage blocks, the messages are assigned in A, B, C, and D.

3. Analysis and optimization for MD5

The MD5 rules which is utilize for calculating the A, B, C, and D in every step of operations. From the algorithm, we can examine the utilities of A, C, D can be got directly, while the calculation of B is quite complicated, which has four mod 232 additions, a logical function and a circular shift left operation, forming the modality of the MD5 algorithm. And the delay of the modality is: T=4×Delay(+) +Delay(R)+Delay(<<S)which is much sizeable than the iteration bound T∞=2×Delay (+)+Delay(R)+Delay(<<S) proposed it. Therefore, to achieve a throughput optimal design, we must compress the architecture of shortest iterative path.

4. Unfolding technique based on pipelining stages:

The 4-stage pipelined architecture is presented in Fig, which introduces the unrolling technique based on MD5. It unrolls all the 16 steps in each round, and performed in alone, so in this architecture, each round calculation will be processed in one cycle at least.

In the 4-stage pipelining are excluding the first 512-bit of message block will be performed in 4 clock cycles, the later ones will be performed in only a one clock cycle. As can be shown in equationBi+1=Bi+((R(Bi,Ci,Di)+Ai+Ti+1+Mj[i+1]))<< S_{i+1} , T_{i+1} and S_{i+1} are constants in every clock cycle, Mj[i+1] is a 32-bit message in a 512-bit of block, which also demonstrated as a fixed constant after the 512-bit is inputted, and the usefulness of Ai equals to Di-1, so if we introduce a temporary variable Tempi in the i^{th} clock cycle, and let Tempi be Di-1+Mj[i+1]+Ti+1, then in the i+1th clock cycle, Bi+1 can be simplified as:Bi+1=Bi+((R(Bi,Ci,Di)+Tempi))< S_{i+1} some pre-computation is performed.

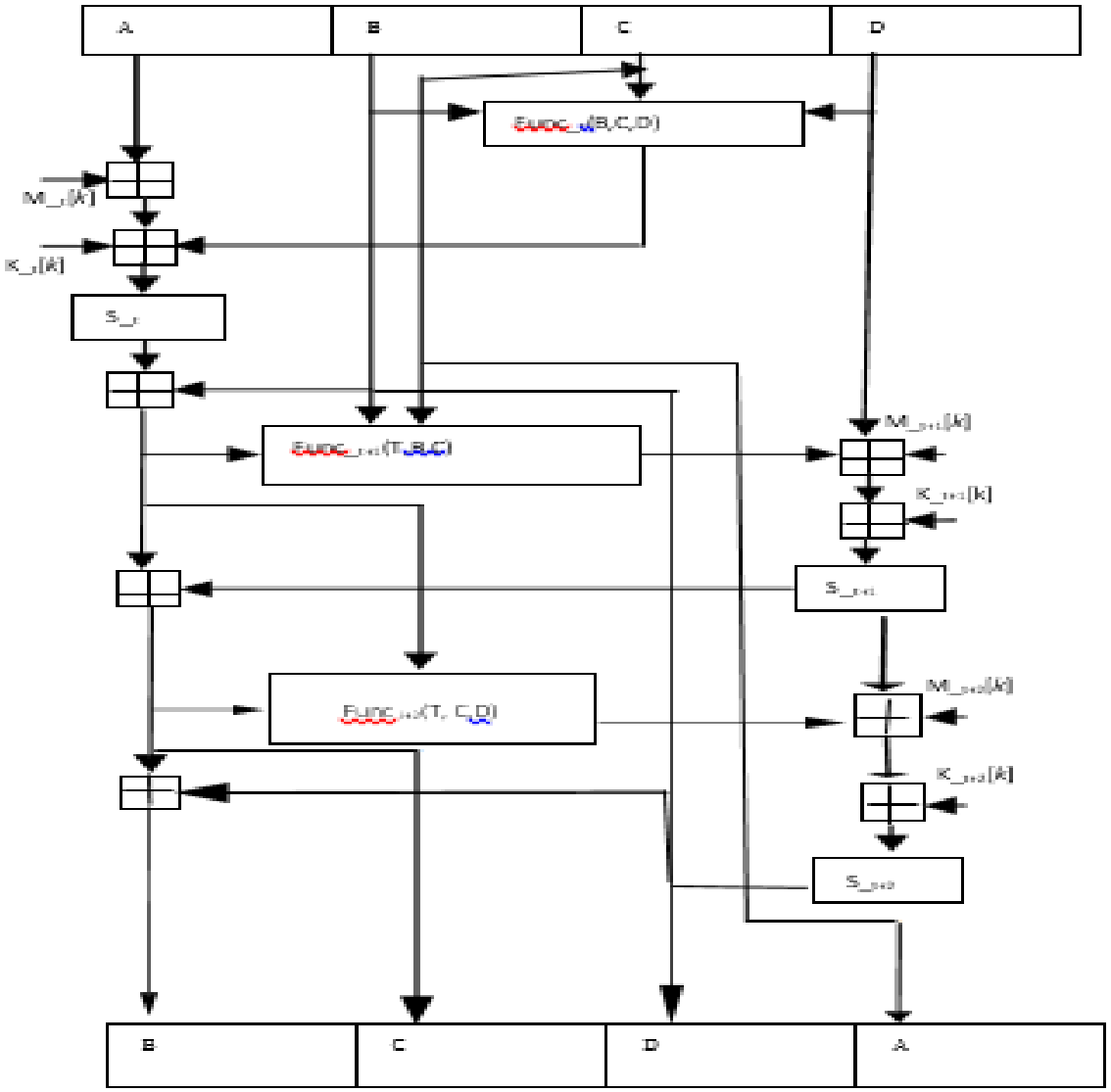


Fig 5: Proposed compression of MD5

5. The pipelined architecture based on iterative technique

After the optimization, every step operations are:

$$B_{i+1} \leftarrow B_i + ((R(B_i, C_i, D_i) + Temp_i) <<< S_{i+1})$$
$$A_{i+1} \leftarrow D_i, C_{i+1} \leftarrow B_i, D_{i+1} \leftarrow C_i$$
$$Temp_{i+1} \leftarrow D_i + T_{i+2} + M_j[i+2]$$

Since register A is not usefulness in the following calculations, the merit of Tempi can be stored in register Ai, i.e. Ai+1=Di+Ti+2+ Mj[i+2]. Moreover, a Carry SaveAdder can also be applied in Di+Ti+2+Mj[i+2], which can save some area and reduce some delay. After the optimization, the algorithm is shortened, there are only two additions instead of four, which improves the speed significantly. And the delay of the path is T=2×Delay (+)+Delay(R)+Delay(<<S) which is equal to the iteration round of MD5 proposed in [4], according to [4], the optimized architecture achieves the maximum outcome of throughput by iterative architecture. Moreover, the logicarea of the considered structure is not increased almost distinguish to the earliest structure.

This structure is found on 4-stage pipelining, it can calculate four different 512-bit of message blocks simultaneously, exclude the first 512-bits of message blocks will be performed in 64 clock cycles, thelater ones will be performed in only 16 clock cycles. Start signal is utilized when a derivation of a new message digest 5 is started, for example, when M0 is processed, and Continue signal is nearly new for the later ones, such as Mj,

where j≥1 . The architecture counts from 0-64 and it is restart to zero when Start or Continue signal is high. Other blocks, comparatively the input block, and memory block and the encrypting block are all controlled by the counter. Then the 4 BUF blocks are used to keep four different groups of 512-bit of message blocks, which are make use of 4 encrypting blocks respectively. The ABCD_REG blocks are used to keep four different groups of middle outcomes of the A, B, C, D. T_REG is recycled to keep the constant T. The four encrypting blocks are the main functional units of the algorithm, which are utilized for the four rounds of calculations respectively. Each encrypting block is fetched to the equivalent BUF block, and which can just only process the given statistics in the connected BUF block. After processing the corresponding data, the result will be transferred to another encrypting block, then the featuresof the connected BUF block are also transferred to the upcoming BUF block. This pattern has the advantages as follows: first, it avoids the bus competition, and each encrypting block can own the whole bandwidth of the connected BUF block, which reduce the delay of the design. Second, this pattern simplifies the logic control, and it also reduces the logic requirements demanding. In addition, this structure has well expansibility, when it needs to increase or decrease the pipelined stages, we can add or delete the corresponding encrypting structure and the connected BUF blocks simply. Take the one stage pipelining in this proposed work for example, which is only needed to delete three encrypting development and the three BUF blocks connected from the constructed Fig. And for the 32-stage pipelining, add encrypting blocks and the corresponding BUF blocks to 32

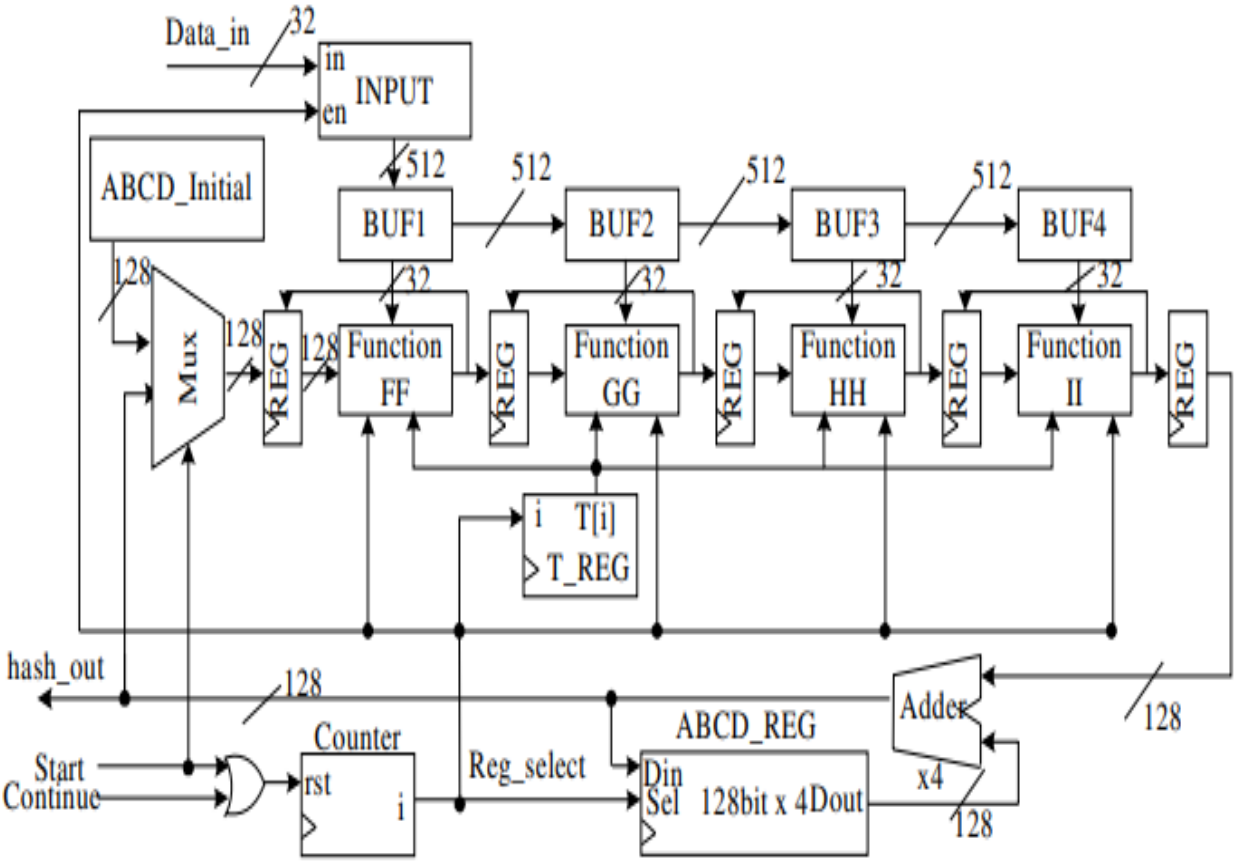


Fig 6: Pipelined architecture based on iterative design

6. The outcomes of the implementations

The unfolding transformation of MD5 is generated from software of Model-simulation and developed by coding language of Verilog. Here we used FPGA registers to Put into effect the MD5 in an effort to gain high through-put and excessive frequency. MD5 provides high performance while

we compare with other hash-functions which includes SHA and RIPEMD. The hash-function utility of MD5 provide more security in data transmission and it is using in so many real time applications example embedded security systems .And one hash utility is cryptographic, it provide tons of greater protection and along with one way function changes plain text to a unique digest of message that is irreversible. In another words, cryptographic hash-function developed to offer protection. There are 2 actual time packages of hash feature depend upon the cryptographic properties. One is Password storage and one more application is Data integrity check.

Advantages:

- perform very well both in required area and performance, although an exact comparison to certain implementations is difficult
- The necessity of the good execution is designs in this designated work is most probably utilizing factor in the reasonable architectures, especially, the optimized critical path, loop unrolling method and the reasonable pipelined stages.

Software Used:

1. Xilinx 14.2 and higher.
2. Hardware Kit: Xilinx Spartan 6

From the chart shown below, the X axis hold the values of total number of registers and Y axis holds the number of pipeline stages.

The compression function of 64 pipeline into 32 need a huge number of registers to gain high throughput. The proposed work attain high throughput by increasing the iterations shown in fig 8. By the stages reduction the amount of latency is reduces from 66 to 65 and the entireprocess is not sluggish when compared with early processed algorithm shown in fig 9.

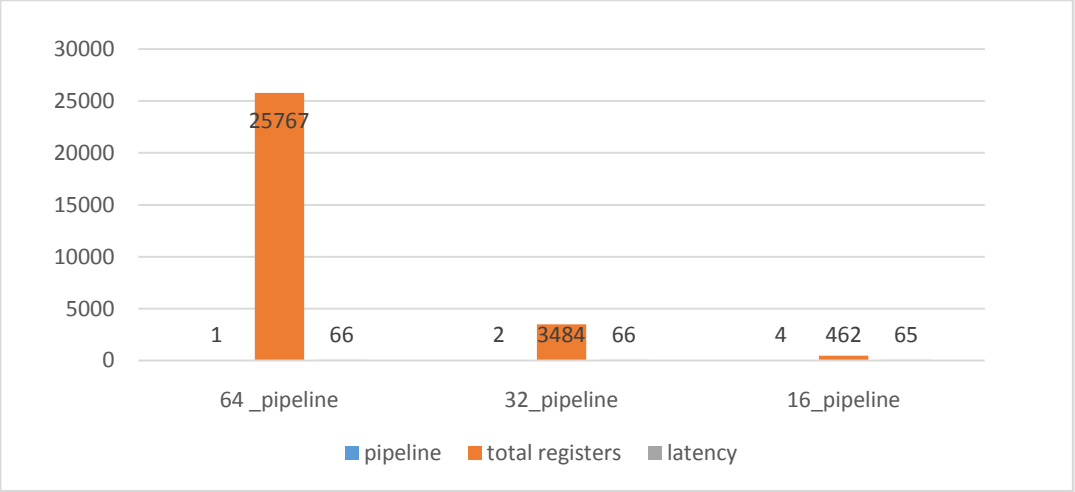
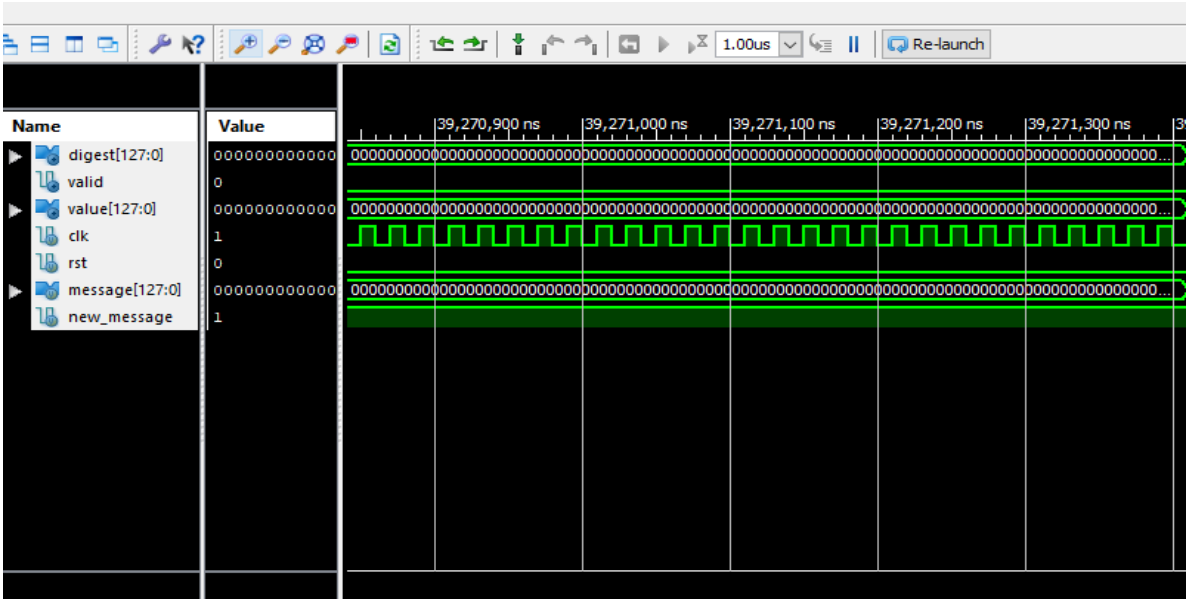


Chart 1: Result of Hardware implementation

Table 1: The frequency and throughput values of implemented work

Design	Clock (Mhz)	LUT's	Throughput Mbps
MD5_16_iterative	102.7	1352	12428
MD5_32_pipeline	97.95	5929	3287
MD5_64_pipeline	86.32	36296	810



.Fig 7: output of 4 cycles of iterations

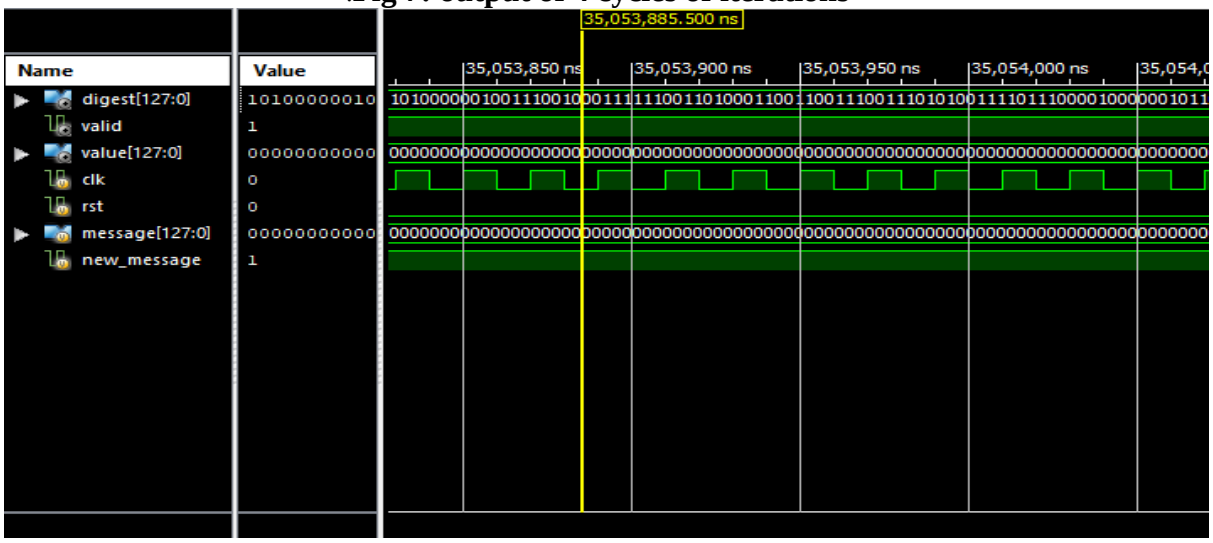


Fig 8: Output of 16 stages of MD5

Conclusion

Four sorts of MD5 configuration dependent on Xilinx 14.2 and higher, Hardware Kit: Xilinx Spartan 6 were effectively incorporated and actualized. The outcomes are shown that the modified MD5 unfolding with 16 phases of pipelining gives a superior MD5 plan as greatly as speed, area and throughput. The gain of unfolding change a span of throughput of MD5 to diminishing its latency. Additionally, the pipelined MD5 configuration has rapid in the increasing the frequency of pipeline. Subsequently, a MD5 configuration consolidating the unfolding change and pipelining can improve the output fundamentally.

References

Journals

[1] J. Deepakumara, H. M. Heys and R. Venkatesan: FPGA implementation of MD5 hash algorithm, Proceedings of the Canadian Conference on Electrical and Computer Engineering, CCECE 2001, Toronto, Canada, Vol.2, pp. 919-924, May 13-16, 2001.Ebert, L. B. (1997). Science of fullerenes and carbon nanotubes. Carbon.35(3),437-438

[2] S. Suhaili and T. Watanabe: High throughput evaluation of SHA- 1 implementation using unfolding transformation, ARPN Journal of Engineering and Applied Sciences, ISSN

1819-6608, Vol.11, No.5, pp. 3350-3355,2016.

[3] Dai, H. (2001). Nanotube Growth and Characterization. Carbon Nanotubes SE - 3, 80, 29–53.

[4] Y. Wang, Q. Zhao, L. Jiang and Y. Shao: Ultra high throughput implementations for MD5 hash algorithm on FPGA, High-Performance Computing and Applications, Lecture Notes in Computer Science, Vol. 5938, pp. 433-441,201l.

[5] T Menakadevi, M Madheswaran, "FPGA implementation of direct digital synthesizer using pipelined cordic algorithm", European Journal of Scientific Research, Vol. 79, Issue. 2, pp. 269-278.

Book

[6] F. R. Henriquez, N. A. Saqib, A. D. Perez and C. K. Koc: Cryptographic algorithms on reconfigurable hardware, Springer Series on Signals and Communication Technology, pp. 189–201, 2006.

Chapter in a Book

[7] P. R. Panda, B. V. N. Silpa, A. Shrivastava and K. Gummidipudi: Chapter 2, Power-Efficient System Design, Springer Science Business Media, LLC, 2010.

Conference Proceedings

[8] K. Jarvinen, M. Tommiska and J. Skytta: Hardware implementation analysis of the MD5 hash algorithm, Proceedings of the 38th Hawai International Conference on System Sciences, 2005