Design & Implementation of VLSI Architecture for XTEA

J.Balakrishna
E&CED, NITH,
Hamirpur, H.P, India.
bkrishna9999@gmail.com

Philemon Daniel
E&CED, NITH,
Hamirpur, H.P, India.
phil_dani@nitham.ac.in

Rajeevan Chandel
E&CED, NITH,
Hamirpur, H.P, India.
rchandel@nitham.ac.in

Abstract— These days Security is an important issue in any type of communication systems. The pervasive communication is the new research area. An ever growing demand for security in such types of communication networks particularly wireless pervasive communication system demands efficient hardware solutions. In this circuit size reduction is the one of the requirement as to lower the cost. This paper can talks over the XTEA-based encryption and decryption in the medium secure systems especially in wireless communication systems. From this implementation we can get an area of 0.121mm^2 based *on the 0.125 um CMOS, making it a highly compact core for the pervasive applications.*

I. Introduction

Information is an important part of our everyday life. Hence people expect to be able to send & receive various types of information anytime and anywhere (omnipresent concept). To be able to support the vision that requires appropriate infrastructure, from the network itself to personal devices that provides the interface for interaction.

The advancement in today technologies makes in such a way that networks for wireless pervasive communication is increasingly possible. In order to gain users' trust as well to prevent future problems which may occur, All the same, these systems need to be providing an appropriate level of security. In the past we have several well known encryption and decryption algorithms such as DES (Data Encryption Standard), 3-DES; AES and AES-3 have been successfully employed on the various applications. The implementations have been for both the software and the hardware-specific based. The implementation of these complex systems requires an exceptional to large hardware. These two specified characteristics are not suitable for the wireless pervasive system where power consumption needs to be reduced as much as possible although the hardware is kept minimal to keep the cost of the system as low. The work in [4] showed that a TEA [5] -based encryption core is an area-time-power efficient, and it is making it suitable for the medium secure systems. However, TEA cannot be used directly as a hash function within an authentication system [6]. This paper proposes that the use of Extended Tiny Encryption Algorithm (XTEA), which is a modification of TEA algorithm developed by David Wheeler and Roger Needham of Computer Laboratory of Cambridge University [7], as the main encryption/decryption engine for medium secure systems. The paper is organized as follows. A brief idea of the TEA and XTEA algorithms is given in the section II. And Section III discusses XTEA-based hash implementation for the authentication process and followed by the implementation results and discussion in section IV. Section V concludes the work.

II. ENCRYPTION USING TEA AND XTEA

The Tiny Encryption Algorithm [5] is said to be one of the fastest and most efficient cryptographic algorithms. TEA is a Feistel cipher that uses only XOR, ADD and SHIFT operations to provide the property of Shannon, diffusion and confusion are necessary for a secure block cipher without any need for the P-boxes and S-boxes. The TEA algorithm source code is shown in Figure 1. TEA is assumed to be as secure as the IDEA algorithm [8], but is much simpler and faster than that. It is also in public domain. For software implementation, the code is lightweight and portable and therefore particularly suits real-time applications. Although TEA has a few weaknesses, most notably from equivalent keys and related-key attacks [9-10], TEA still provides good security for mobile systems.

```
void encrypt(long* v, long* k)
{ unsigned long y=v[0], z=v[1], sum=0,
   delta=0x9e3779b9,n=32;
   while (n-->0)
   { sum += delta;
     y += ((z<<4)+k[0]) (z+sum)^((z >>5)+k[1]);
     z += ((y<<4)+k[2]) (y+sum)^((y>>5)+k[3]);}
 v[0]=y; v[l]=z;}
```

Fig.1. TEA encryption source code

For the hardware implementation, the structure is simple, with only hardware for the addition, XOR, and registers are required. This is compared with the other block ciphers such as DES and AES where we need bigger blocks such as the s-boxes are necessary for the hardware. Figure 2 shows a block diagram of TEA encryption (for one cycle).

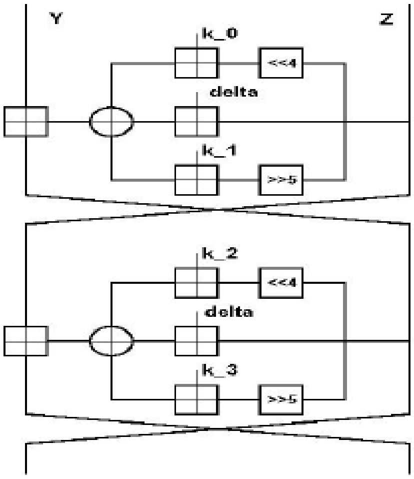


Fig2. TEA algorithm Feistel structure, cycle *i*

```
void decrypt (long* v, long* k)
 { unsigned long y=v[0], z=v[1], delta=0x9e3779b9,
    n=32, sum=32*delta;
    while (n-->0)
   {  z -= ((y<<4)+k[2]) (y+sum)^((y>>5)+k[3]);
      y -= ((z<<4)+k[0]) (z+sum)^((z >>5)+k[1]);
      sum -= delta;
    }
   v[0]=y; v[1]=z;
}
```

Fig.3. TEA decryption source code

A. **XTEA**

XTEA (eXtended TEA) is a 64-bit block Feistel network which is a 128-bit key that is designed to correct the weaknesses in TEA. It was also created by David Wheeler and Roger Needham. The algorithm was first reported in [7]. Similar to TEA, XTEA is also not patented. XTEA is similar to TEA in such a way that it requires addition, XOR and shift operations only. The only key difference is that with the more complex key scheduling and the rearrangement of those three operations (addition, XOR and shift). Usually the number of round recommended in this algorithm is 64. The source code of XTEA is shown in Figure 4, with the block diagram in Figure 5.

```
void encrypt(unsigned long* v, unsigned long* k)
{
unsigned long y=v[0], z=v[1],i;
unsigned long sum=0, delta=0x9E3779B9;
for(i=0; i<32; i++)
{
  y += ((z << 4^ z >> 5) + z) ^ (sum +k[sum & 3]);
  sum += delta;
  z += ((y << 4^ y >> 5) + y) ^ (sum +k[sum>>11 & 3]);
}
v[0]=y; v[l]=z;
}
```

Fig.4. XTEA encryption source code

```
void decrypt(unsigned long* v, unsigned long* k)
{
unsigned long y=v[0], z=v[1], delta=0x9e3779b9,
  sum=32*delta;
unsigned int i;
for(i=0; i<32; i++)
{
 z += ((y << 4^ y >> 5) + y) ^ (sum +k[sum>>11 & 3]);
 sum -= delta;
y +-= ((z << 4^ z >> 5) + z) ^ (sum +k[sum & 3]);
 }
v[0]=y; v[l]=z;
}
```

Fig.5. XTEA encryption source code

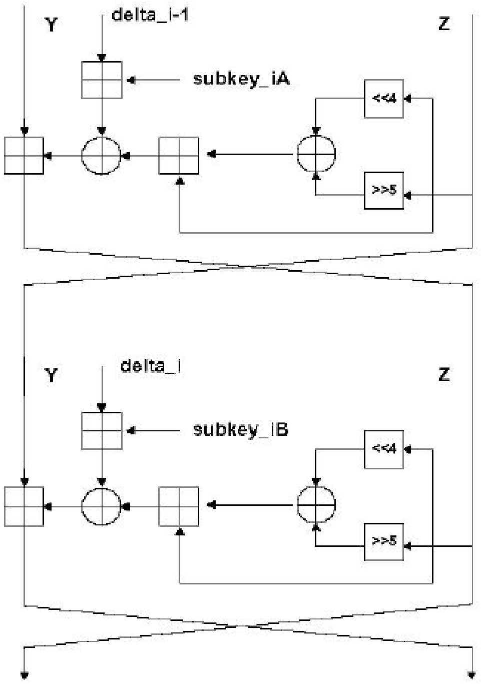


Fig.6. XTEA algorithm Feistel structure, cycle *i*

III. XTEA BASED HASH FUNCTION FOR AUTHENTICATION

A hash function translates the message of any length into a fixed-length string of output that called as a digest. The message digest is the unique for each message, hence digests produced by sender and receiver can be compared to perform the authentication or for checking the message integrity. Hash is a one-way function, that the message cannot be recovered by examining the digest. The two most commonly used Hash functions are MD5 and SHA-1. But they are computationally complex and have proven to be reasonably secured, although in 2005 security flaws for both algorithms have been reported [11-12].

Instead of using the specific hash functions, other way of realizing the hash is to use block cipher as the compression function. Several schemes such as Miyaguchi-Preneel, Davies-Meyer and Matyas-Meyer-Oseas [13] have been proposed. They used inside the Merkle–Damgård structure to build actual hash function. The block cipher based hash function is usually slower than a specially designed hash function. The benefit of using such approach is that the same hardware can be shared for both encryption and decryption. These can be particularly useful in wireless pervasive communication where we need hardware resource is at premium.

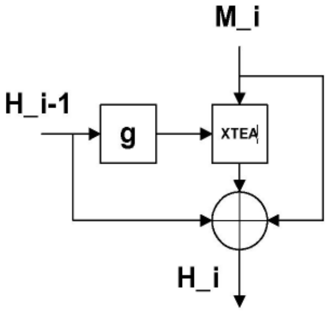


Fig.7. The Miyaguchi-Preneel Hash construction

To illustrate the scheme, this paper uses the Miyaguchi-Preneel Hash compression function as one of an example. The Miyaguchi-Preneel hash compression function is an extended variant of the Matyas-Meyer-Oseas. The function implements the following equation.

$$H_{i} = E_{g(H_i - 1)}(m_i) \oplus H_{i-1} \oplus m_i$$

In this process, the plaintext in the form of a block message mi is encrypted using XTEA, then XOR operation with m_i itself and the previous hash value H_{i-1} (or a pre specified initial value H_0 for first round). The result is the next hash value H_i. The scheme is illustrated in Figure 7. The function g is used to pad or convert H in case of there is a difference in block and key sizes. Note also that the roles of M_i and H_i, may also be switched, so that H_i, is encrypted under the key m_i instead.

A Hash lock system for secure RFID proposed in [14] is described here as an example of a potential usage of XTEA enabled, hash authentication scheme in secure wireless system. During the procedure, an RFID tag responses to a query from a reader by sending its metaID that is a hash value of its authentication key k and a random number. Reader will look in the database for the key k given the metaID and the random number received and sends the value back to tag. The tag computes a hash value of the received authentication key and compares with the stored value to authenticate the reader.

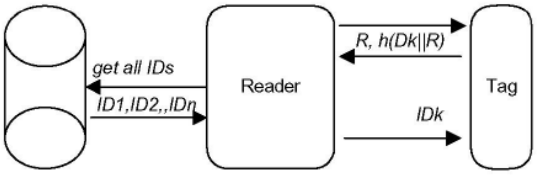
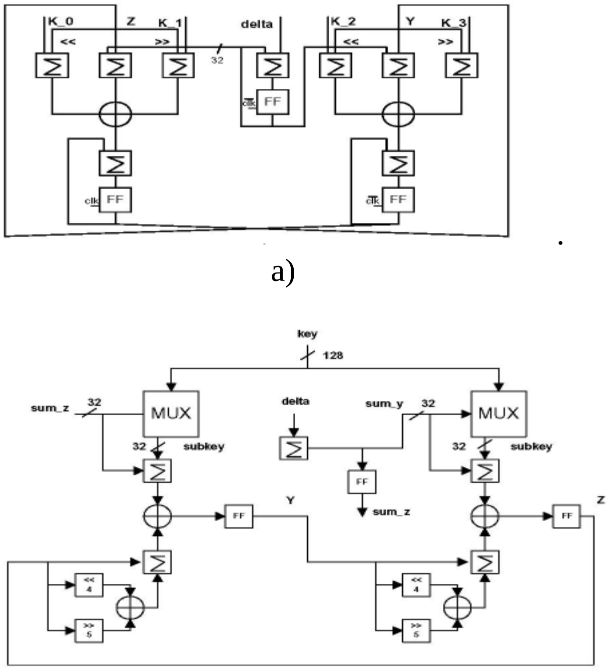


Fig.8. Hash locking in randomized read access control

IV. DISCUSSIONS ON IMPLEMENTATION

In [4], the TEA encryption algorithm was already shown to be very efficient considering area-time-power criteria. The core of the algorithm was best implemented using multiple 32-bit adders that were employed simultaneously to perform operations needed for one encryption cycle. Other architectures considered included sequential and digit-serial designs. In summary the parallel design was shown to be best suited to TEA, giving significantly better performance in terms of speed and power, while suffered only a relatively small increase in the area compared to the other two. The speed advantage was obvious given its parallel architecture, while power consumption of the design was also better given that other two suffered from an additional power due to the additional control logics clocked at a much higher speed to achieve the same throughput. The areas were relatively similar because of the fact that when implementing a small circuit like TEA the area reduced by using sequential designs at the world or sub-word (digit) levels is largely compensated by the requirement for additional control logics. Given the information, and the fact that TEA and XTEA are mostly similar in terms of architectures and operations, this paper will concentrate specifically on the parallel architecture

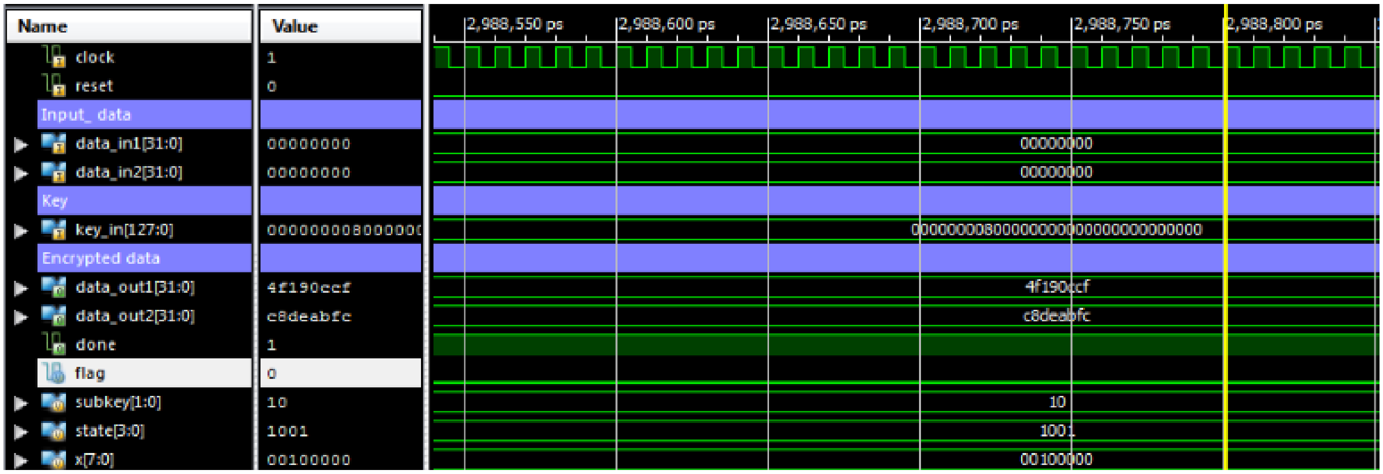


a)

b)

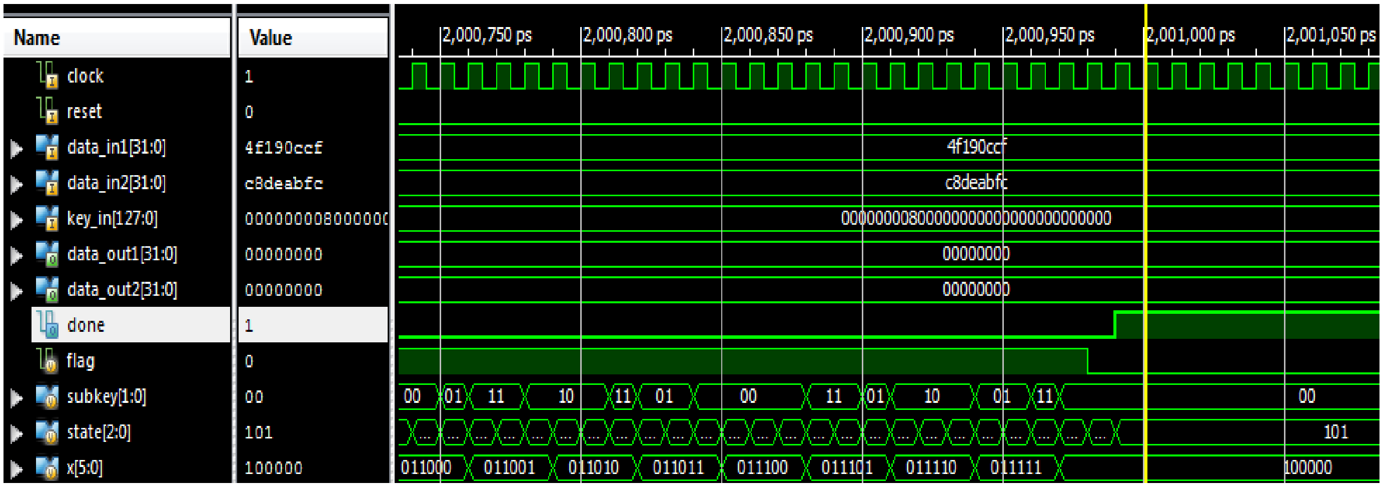
Fig.9. Architectures a) TEA, b) XTEA

The parallel-style cores for TEA and XTEA algorithms are implemented using Verilog HDL and are functionally verified using ModelSim for the Xilinx FPGA implementation. The two architectures are as shown in Figure 9. The designs are also synthesized for ASIC implementation using Mentor Graphic's Leonardo Spectrum targeting 0.125 um CMOS technology [15]. The estimated results for area and time parameters are as shown in Table I. It can be seen that the performance of the XTEA core is similar to the TEA, with changes in terms of area and speed. These numbers illustrate its suitability as a compact and efficient core for wireless communication. The performance of the core will also be significantly better when implementing on a smaller CMOS technology. To build a complete encryption and authentication system, the core will need some further additional logics. Such designs are currently being investigated.



Name	Value
clock	1
reset	0
Input_ data	
data_in1[31:0]	00000000
data_in2[31:0]	00000000
Key	
key_in[127:0]	000000008000000
Encrypted data	
data_out1[31:0]	4f190ccf
data_out2[31:0]	c8deabfc
done	1
flag	0
subkey[1:0]	10
state[3:0]	1001
x[7:0]	00100000

Fig.10. simulation results for Xtea encryption



Name	Value
clock	1
reset	0
data_in1[31:0]	4f190ccf
data_in2[31:0]	c8deabfc
key_in[127:0]	000000008000000
data_out1[31:0]	00000000
data_out2[31:0]	00000000
done	1
flag	0
subkey[1:0]	00
state[2:0]	101
x[5:0]	100000

Fig.11. simulation results for Xtea decryption

TABLE I. IMPLEMENTATION RESULTS

	XTEA	TEA
Area(mm^2)	0.121	0.117
Max clock(MHz)	48.1	51.2
Clock per cycle	1	1
Max. cycle Throughput(MHz)	48.1	51.2

V. CONCLUSION

In this paper, a potential solution for security core for wireless pervasive communication based on XTEA algorithm is proposed. The XTEA algorithm is shown to be suitable for hardware implementation of a hash function for authentication. Implementation of XTEA core using parallel architecture based on 0.125 um CMOS technology has the area of 0.121 mm^2 and a speed of 54.5 MHz.

REFERENCE

[1] *Ehrsam et al., Product Block Cipher System for Data Security, U.S. Patent 3,962,539*

[2] *Joan Daemen and Vincent Rijmen, The Design ofRUndael: AES- The Advanced Encryption Standard, Springer-Verlag, 2002*

[3] *NIST (National Institute of Standards and Technology) FIPS PUB 180-2 http:Hcsrc.nist.gov/CryptoToolkit/tkhash.html*

[4] *P. Israsena, "Design and Implementation of Low Power Hardware Encryption for Low Cost Secure RFID using TEA," Proc. International Conference on Information and Communication Systems (ICICS 2005), pp. 1402-1406, Dec 2005*

[5] *David J. Wheeler and Roger M. Needham, "TEA, a tiny encryption algorithm," Proc. Fast Software Encryption: Second International Workshop, Lecture Notes in Computer Science (LNCS), vol. 1008, pp. 363-366, December 1994*

[6] *John Kelsey, Bruce Schneier, and David Wagner, "Keyschedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES," Lecture Notes in Computer Science (LNCS), vol. 1109, pp. 237-251, Springer-Verlag 1996*

[7] *Roger M. Needham and David J. Wheeler. "Tea extensions," Technical report, Computer Laboratory, University of Cambridge, October 1997*

[8] *Xuejia Lai and James L. Massey, "A Proposal for a New Block Encryption Standard," EUROCRYPT 1990, pp. 389-404, 1990*

[9] *John Kelsey, Bruce Schneier, and David Wagner, " Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X NewDES, RC2, and TEA," Lecture Notes in Computer Science (LNCS), Vol. 1334, pp. 233-246, Springer-Verlag 1997*

[10] *Seokhie Hong, Deukjo Hong, Youngdai Ko, Donghoon hang, Wonil Lee, and Sangjin Lee, "Differential cryptanalysis of TEA and XTEA," Proc. ICISC 2003, 2003b*

[11] *Xiaoyun Wang and Hongbo Yu, "How to Break MD5 and Other Hash Functions," EUROCRYPT 2005*

[12] *Xiaoyun Wang, Yiqun Lisa Yin and Hongbo Yu, "Finding Collisions in the Full SHA-1", CRYPTO 2005*

[13] *Menezes, van Oorschot and Vanstone, Handbook of Applied Cryptography, chapter 9, 2001*

[14] *Stephen A Weis, Sanjay E Sarma, Ronald L Rivest, and Daniel W Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems," Proc. First International Conference on Security in Pervasive Computing, Boppard, Germany, March 12-14, 2003*

[15] *AustriaMicroSystems 0.125um CMOS Digital Standard Cell Databook, 2003*