World Applied Sciences Journal 17 (9): 1149-1158, 2012
ISSN 1818-4952

Reliable and High-Speed KASUMI Block Cipher by Residue Number System Code

Hamidreza Mahyar

Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran

Abstract: Third generation cellular network technology (3G) can revolutionize communications and data exchanges between many people in a more overwhelming fashion than 2G and 2.5G networks did. The 3G UMTS, the 3G GSM and the 3G GPRS rely on the KASUMI block cipher. Therefore, increasing speed, decreasing power consumption and error detection/correction are the major concerns of the KASUMI algorithm and its generation. On the other hand, Residue Number System is a non-weighted number system and it is currently considered as an important method for high-speed, low-power, parallel and carry-free arithmetic realizations. Redundant Residue Number System is an extension of RNS that also supports error detection and correction. Multi-Level Residue Number System uses the new Residue Number System for each modulo, so the speed of operations is increased because of decreasing modulo. In this paper we synthesize these numeric systems and use "Multi-Level Redundant Residue Number System" for increasing speed and reliability of the KASUMI. This system also supports high error detection and correction capabilities. Moreover, One-Hot Residue Number System is a high performance technique in which the propagation delay of this implementation is just one transistor. In this document we also utilize "One-Hot Multi-Level Residue Number System" in the KASUMI block cipher to achieve more optimizations in terms of delay, power consumption, hardware and lastly power delay product.

Key words: Cryptography · Error Detection and Correction · Fault Tolerant Systems · KASUMI Block Cipher · Residue Number System (RNS) · VLSI

INTRODUCTION

Kasumi is a Japanese word meaning Mist. It is a Block Cipher with Fiestel Structure developed by Mitsubishi Electric Corporation, designed for 3GPP (3rd Generation Partnership Project) [1] to be used in the cellular communications networks and safety of many wireless standards. The KASUMI block cipher was designed as a modification of the Misty algorithm and it could resistance against Linear and Differential Cryptanalysis techniques [2]. Kasumi is a widely used block cipher in synchronous wireless standards (GSM, GPRS and UMTS) [3-5]. One of the most important parts of the KASUMI is the key schedule and its processing is controlled by a 128-bit encryption key K. By using a 128-bit ciphering key K, it modifies a 64-bit Plaintext to a 64-bit Ciphertext.

The 64-bit input is divided into two 32-bit strings L_0 and R_0. The outputs of each round are produced according to the following equation:

$$R_i = L_{i-1},\ L_i = R_{i-1}\ XOR\ F(L_{i-1}, RK_i),\ 1 \le i \le 8 \quad (1)$$

Where F denotes the round function with L_{i-1} and round key RK_i as inputs. The round key RK_i comprises the subkey triplet (KL_i, KO_i, KI_i). KASUMI has a Feistel structure containing eight rounds; the produced Ciphertext is the 64-bit string derived from the concatenation of the L_8 and the R_8, which are produced at the end of the eighth round. The F itself is constructed from the FL and FO subfunctions, with associated sub-keys Kl_i (used with FL) and sub-keys Ko_i and KI_i (used in FO), followed by a bit-wise XOR operation with the previous branch. Fig. 1 shows the structure and

Corresponding Author: Hamidreza Mahyar, Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. Tel: +98 912 2499095, Fax: +98 21 88055794.

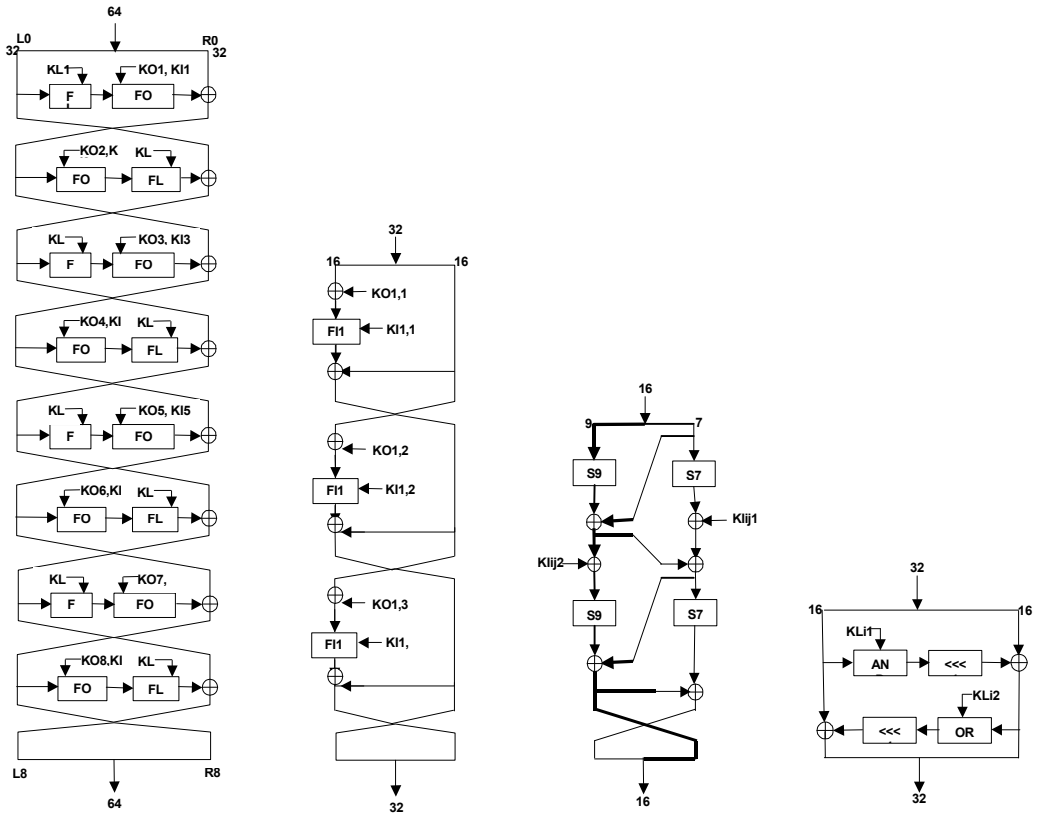

Fig. 1: The KASUMI Block Cipher

components of the KASUMI block cipher. This algorithm has two different forms. In the odd round the *FL* function is performed first and then the *FO* function as follows:

$$F_i(K_i, L_{i-1}) = FO\,(KO_i, Ki_i, FL\,(KL_i, K_{i-1})) \quad (2)$$

In the even round, the order of the functions is reversed as follows:

$$F_i(K_i, L_{i-1}) = FL\,(KL_i, FO\,(KO_i, KI_i, L_{i-1})) \quad (3)$$

The 128-bit round key, RK_i, is derived from key K. In the KASUMI the same algorithm is used both for encryption and decryption. *FL*, shown in Fig. 1(d), is a 32-bit function made up of simple AND, OR, XOR and left rotation operations. *FO*, depicted in Fig. 1(b), is also a 32-bit function having a three-round Feistel organization which contains one *FI* block per round. *FI*, see Fig. 1(c), is a non-linear 16-bit function having itself a four-round Feistel structure; it is made up of two nine-bit substitution boxes (S-boxes) and two seven bit S-boxes.

Fig. 1(c) shows that data in the *FI* function flow along two different paths: a nine-bit long path (thick lines) and a seven-bit path (thin lines). Notice that in Feistel structures, such as the one used in this algorithm, each round's output is twisted before being applied as input to the following round. After completing eight rounds, KASUMI produces a 64-bit long Ciphertext block corresponding to the plaintext input block. As we know about the KASUMI block cipher, the key schedule plays an important and essential role, so improvement of that can increase performance of the KASUMI algorithm.

The rest of this paper is organized as follows: In section 2 we recollect Residue Number System. In section 3 and 4 we present Multi-Level Residue Number System and Redundant Residue Number System, respectively. Multi-Level Redundant Residue Number System and its application in the KASUMI algorithm are represented in section 5. In section 6 we describe One-Hot Multi-Level Residue Number System and its application in the KASUMI. Simulation results and discussion with previous work are shown in section 7. Finally, in section 8 the paper conclusions are given.

Residue Number System: A Residue Number System is specified in terms of a set of relatively prime integers $\{m_i\}_{i=1,\ldots,n}$ such that $gcd\ (m_i, m_j) = 1$ for i is not equal to j, where gcd means greatest common divisor of m_i and m_j. For such a system M, $M = m_1 \times m_2 \times \ldots \times m_n$ is the dynamic range and any integer X in interval [0, M-1] can be uniquely represented by a N-tuple $(x_1,x_2,x_3,\ldots,x_n)$, where x_i is the residue of X in modulo m_i for $i = 1, 2, \ldots, n$ [6-10].

A Residue Number System is specified in terms of a set of relatively prime integers $\{m_i\}_{i=1,\ldots,n}$ such that $gcd\ (m_i, m_j) = 1$ for $i \neq j$, where gcd means greatest common divisor of m_i and m_j. For such a system M,

$M = m_1 \times m_2 \times \ldots \times m_n$, is the dynamic range and any integer $X \in [0, M\text{-}1]$ can be uniquely represented by a N-tuple $(x_1,x_2,x_3,\ldots,x_n)$, where x_i is the residue of X in modulo m_i for $i = 1, 2, \ldots, n$ [6-10].

The reconstruction of X from its residues $(x_1,x_2,x_3,\ldots,x_n)$ is based on the Chinese Remainder Theorem (CRT) shown by:

$$X = \left\langle \sum_{i=1}^{n} (x_i N_i)_{m_i} \times M_i \right\rangle_M$$

$$M = \prod_{i=1}^{n} M_i$$

$$M_i = \frac{M}{m_i} \quad , N_i = \left\langle M_i^{-1} \right\rangle_{m_i} , i = 1,2,3,\ldots,n \qquad (4)$$

The notation $\left\langle M_i^{-1} \right\rangle_{m_i}$ in (4) denotes the multiplicative inverse of M_i modulo m_i [11].

Another advantage of this system is security; for the reason that the conversion of RNS to weighted number system needs moduli which operate as a key.

The Residue Number System due to particular features has numerous applications in arithmetic functions such as Cryptosystem, Digital Signal Processing [12], Digital Filtering [13], Coding theory [14], RSA encoding algorithm [15-16], KASUMI block cipher, Digital Communications [17], Ad-Hoc networks, distributed dependable and secure Data Storage and Retrieval [18], ability of Error Detection and Correction [19-20] and Fault Tolerant Systems [21].

This system is commonly benefited in those areas where XOR, addition, subtraction and multiplication operations of numbers are being repeated. RNS substantially is fault tolerant owing to when one error occurs on one remainder, it won`t be conveyed to other remainders. Error detection and correction are wholly possible in this system [22].

Multi-Level Residue Number System: In the Residue Number System, it is conceivable to accomplish arithmetic calculations on each modulus with a new Residue Number System because RNS has a considerable notice for increasing calculation speed, reducing power consumption and increasing the security and fault tolerance. This procedure can recur in several levels until we gain very small moduli. Multi-Level Residue Number System (MLRNS) is achieved form the above mentioned procedure. In Multi-Level RNS, we should take into account the dynamic range of any sub-Residue Number Systems. For example, the Residue Number System dynamic range that is considered for i^{th} level of each $(i\text{-}1)^{th}$ moduli-level must be greater or equal to those moduli. In this article, in order to have simplicity of the representation Two-Level Residue Number System is being analyzed. It should be mentioned that this method could be generalized to more than two levels [23].

Two-Level Residue Number System has a much higher security level than the Residue Number System, for the reason that two symmetrical coding key algorithms are used inside each other. The other advantage of Two-Level RNS is the simple selection of moduli set for a large dynamic range that is by selecting a few large moduli and applying a new Residue Number System with a lower power for second level this capability is achieved. By using few moduli with higher power in the first level; first the need for moduli to be relatively pair-wise prime is eliminated and there is no obligation for the moduli to be symmetric and regular, second as the number of moduli is reduced the concerning conversion circuits, become simple and the operation is done rapidly [24].

Furthermore, in the second level since the moduli are small because of the limited propagation of carries, the internal calculations of the Residue Number System are done faster. Arithmetic computations of Two-Level Residue Number System are performed on the second level residues. Hence, two operand arithmetic operations are defined as follows:

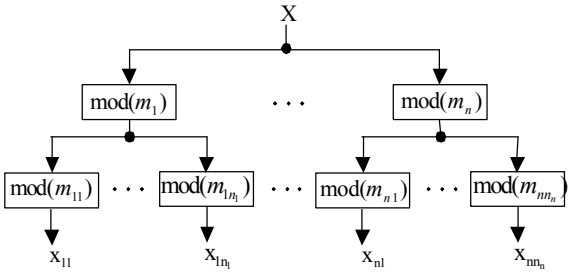

Fig. 2: Conversion from weighted number system to Two-Level Residue Number System.

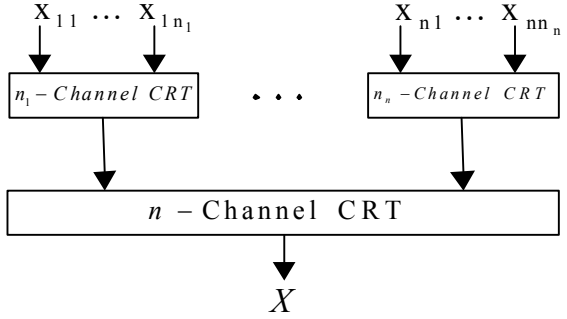

Fig. 3: Conversion from Two-Level Residue Number System to weighted number system.

$$\{z_{i1}, z_{i2}, z_{i3}, \ldots, z_{in_i}\} = \{x_{i1}, x_{i2}, x_{i3}, \ldots, x_{in_i}\} \circ \{y_{i1}, y_{i2}, y_{i3}, \ldots, y_{in_i}\} \quad (5)$$

In the place which $z_{ij} = (x_{ij} \circ y_{ij}) \bmod m_{ij}$, $i = 1,2,3,...,n_i$ and "°" could be addition, subtraction, XOR and multiplication [25-27].

The process of converting weighted number system into Two-Level Residue Number System is shown in Fig. 2. At first, the original number should be converted to the first level Residue Number System and then, the generated residues should be converted to the second level Residue Number System.

The generic schema of the reverse conversion is illustrated in Fig. 3. Firstly, it uses $n_i \times n_i$ – *channel* CRT for $i = 1,2,3,...n$, due to convert second level residues to equal residues of the first level and secondly, the acquired residues are changed into weighted number system by using an *n*-channel CRT.

Redundant Residue Number System: A Redundant Residue Number System (RRNS) is defined as a Residue Number System added with *r* additional moduli. RRNS code offers fast and built-in self-checking computation. These advantages open a new direction in fault tolerance area, especially for error detection/correction codes, to improve the reliability of KASUMI block cipher. The first *h* moduli from a set of non-redundant moduli and their product represent the legitimate range, *M* that is:

$$M = \prod_{i=1}^{h} m_i \quad (6)$$

The remaining $P - h = r$ moduli from the set of redundant moduli that allows error detection and correction where M_R is specified as follows:

$$M_R = \prod_{i=h+1}^{h+r} m_i \quad (7)$$

A residue vector $(x_1,x,...,x_h,x_{h+1},...,x_{h+r})$ is given and the corresponding integer X is the member of the interval [0, M_T - 1], where $M_T = M \times M_R$.

This interval, usually called total range, can be divided into two adjacent intervals by considering the ranges defined by the non-redundant and redundant moduli. The interval [0, M - 1] is called the *legitimate range* and the interval [M, M_T - 1] is the *illegitimate range*.

In order for RRNS to have self checking, error detection and correction properties, the information or data has to be constrained within the legitimate range. It has been shown that RRNS with *r* redundant moduli can detect *r* errors and can correct up to $\left\lfloor \frac{r}{2} \right\rfloor$ errors, where $\lfloor \rfloor$ denotes the integer part. This restriction defines the dynamic range of the system. The m_i-projection of X, denoted X_i, is specified as the residue vector $(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_P)$, *i.e.* the representation of X with the i^{th} residue digit deleted.

A single error occurs when a legitimate vector $(x_1, \ldots, x_i, \ldots, x_P)$ is changed into a different residue vector, $(x_1, \ldots, x_i, \ldots, x_P)$ by the occurrence of an error in the i_{th} digit, the number corresponding to his this vector is X.

In [28] proved that, under the hypothesis of ordered moduli (*i.e.* $m_i < m_{i+1}$ for all i), in a RRNS representation with $r = 2$ any error in a single module products an illegitimate number X. Moreover, X_i is legitimate, wherein i is the residue affected by an error, while the other projections X_j, for j is not equal to i, $i = 1, \ldots, P$ are all illegitimate. From these considerations is straightforward to detect and correct an error in the RRNS representation. The erroneous module is that characterized by his m_i-projection belonging to the legitimate range, while the correct value of the integer can be obtained by performing the reverse conversion of the X_i projection.

Multi-Level Redundant Residue Number System and its Application in the Kasumi Algorithm

Multi-level Redundant Residue Number System: In this article Multi-Level Redundant Residue Number System is presented for increasing error detection and correction and increasing security in high speed computing without carry propagation. In Multi-Level Redundant Residue Number System, redundant moduli could be used for error detection and correction. Many errors could be detected or corrected in low levels in this method. Multi-Level Redundant Residue Number System has the capability to obtain much fault tolerance for more important moduli in first level. In this method, moduli which are supposed for lower levels of Redundant Residue Number System have more redundancy or have more Hamming distance in other words. There is an important issue in this system; the error detection and correction of RRNS coding are performed on moduli not on bits (Note that moduli might be single bit or more). Therefore, if a single-bit-error occurs, other moduli will not be affected. So, the rate of error detection and correction will be increased for single-bit-error by using small moduli.

Error Control Mechanism in the KASUMI Block Cipher: KASUMI is an eight round Feistel type cipher, 64-bit block cipher with a 128-bit key. KASUMI shares with MISTY1 the design goals of having a numerical basis for its security and of being sufficiently fast when implemented in hardware. To meet the design goals, the key schedule was carefully chosen to optimize the hardware performance. Since it is made clear in the KASUMI algorithm, one of the most important and vital components are the keys. Therefore if we can improve the performance of the key schedule in terms of complexity, speed, delay, power and security; then the KASUMI block cipher will be progressed and finally we will revolutionize cellular communications networks and safety of many wireless standards. This section explains high secure and reliable method for designing the key schedule and also achieves more performance in the above features. Keep in mind that increasing efficiency of foundation units can significantly improve the characteristics of system.

KASUMI has a simple and linear key schedule in order to make the hardware significantly smaller and to reduce key set-up time. Every bit of the key is used once in every round. For the purposes of this paper it is sufficient to consider every i^{th} round key as being made up of three parts KL_i, KO_i and Ki_i.

These in turn are divided into a total of eight 16-bit parts such that $KL_i = KL_{i,1} \mid KL_{i,2}$, $KO_i = KO_{i,1} \mid KO_{i,2} \mid KO_{i,3}$ and $KI_i = KI_{i,1} \mid KI_{i,2} \mid KI_{i,3}$, where | denotes concatenation. The $KI_{i,j}$ are further divided into two parts (first one of seven bits, second of nine) with

$$Ki_{i,j} = KI_{i,j,1} \mid KI_{i,j,2}.$$

The round keys are derived by splitting the key K into eight 16-bits parts $K = K_1 \mid \bullet \bullet \bullet \mid K_8$. Each part of the key is used to derive exactly one round key part in each round. The key schedule has the property that changing one bit of K changes exactly one key bit of each round key. We want to add error detection/correction mechanism into the key schedule with high speed operation.

In this scheduler, because of inherent property of Multi-Level Residue Number System in splitting large numbers to small moduli due to increase speed, we use this property to split key K into sub-keys. This technique also uses the redundant moduli to detect likely errors as a result of detecting and correcting errors. Fig. 4 shows the secure and reliable mechanism that used in the key schedule of the KASUMI block cipher. This method can exert for the other sub-keys (KL_i, KO_i, KI_i and KI_{ij}) the same as key K.

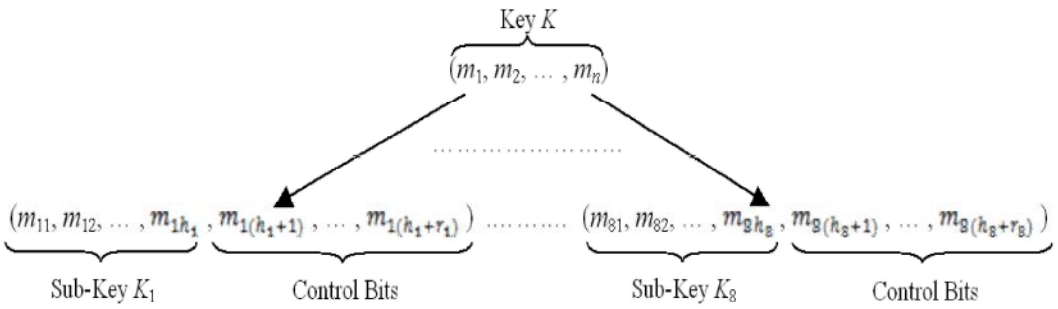

Fig. 4: Multi-Level Redundant Residue Number System mechanism applied in the key schedule of KUSUMI.

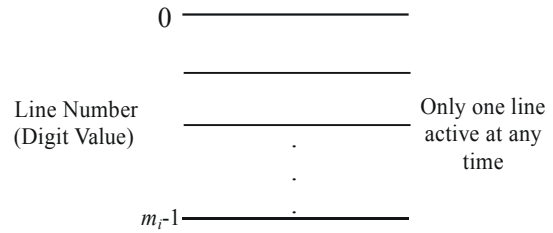

Fig. 5: One-Hot representation for m_i remainders.

One-Hot Multi-Level Residue Number System and its Application in the Kasumi Algorithm

One-Hot Multi-Level Residue Number System: One-Hot Residue Number System is a high performance technique that is derived from Residue Number System. In One-Hot RNS we discuss in m_i that is the remainder of moduli. They are shown from zero to m_i-1. In One-Hot we define a signal line dedicated for each remainder. The activity of each signal shows the similar remainder with it. One–Hot representation for m_i moduli remainders are shown in Fig. 5. In this system it is the rule that in each moment only one of the lines is ON and active but the remaining lines are inactive.

Via changing the amount of input in entrance, the amounts of two lines change at maximum level. Therefore, the power consumption dissipation is at minimum level. Because of its structure, OHRNS is so simple, high speed and low power. In One–Hot representation of remainders, operations are done by circular shifts. The function of this system is based on barrel shifter that has shift entry and data entry.

Barrel shifters are simple, regular layout of arithmetic circuits, zero-cost implementation of inverse and index calculation and moduli conversion and also contain excellent Power-Delay Product.

One of the significant features of One-Hot is its independence to the type of moduli but one of the shortcomings of One-Hot System is that it couldn't be implemented for large moduli for the reason that the number of transistors are increased. Consequently, this system is appropriate for small moduli. Notice that we can solve this problem with combining Multi-Level RNS and One-Hot RNS. As it was mentioned in this section, One-Hot Residue Number System is suitable for small moduli, but for large moduli it is not applicable because the transistors are added in arithmetic calculations. On the other hand, in Multi-Level Residue Number System the arithmetic operations are done on small moduli.

In this article these two techniques are combined with each other and the result is “One-Hot Multi-Level Residue Number System”. In combining these two techniques, first a moduli collection with large modulus is selected and then for each of these moduli one new Residue System is chosen and the procedure is repeated. Hence in the final level, one Residue Number System with small moduli is gained and it is the best condition to make a profit of OHRNS on small moduli.

High-Speed and Low-Power Key Scheduler in the Kasumi: One of the major differences between KASUMI and MISTY1 is in the key schedule. The key scheduler receives the 128-bit initial input key K and generates the round keys *KL* (32-bit long), *KO* (48-bit long) and *KI* (48-bit long) for each of the eight rounds.

Each round key is split into two or three 16-bit parts and these parts are the ones directly computed by the key scheduler. The input key *K* is split into eight 16-bit parts K_i, $1 \le i \le 8$ and then the scheduler performs left rotation operations (“<<<”) and computes the K_i' values, which are defined as follows:

$$K_i = K_i \text{ XOR } C_i,\ 1 \le i \le 8 \qquad (8)$$

Where C_i's are known and fixed constants. The constants C_i are interleaved with the key bits in order to avoid weak-key classes based on fixing key bits to be zero. Such weak keys were found in IDEA and in order ciphers as well. In each round, eight words are used as the round subkey (up to some in-word rotations). Therefore, the 128-bit subkey of each round is a linearly dependent on the secret key in a very simple way. In the previous section, we presented the new technique to achieve high-secure and reliable key schedule while in this section, we show a novel design of the key schedule with using excellent architecture based on One-Hot Multi-Level Residue Number System for gaining high-speed and low-power operation.

In Fig. 6, One-Hot RNS is represented in which two entries are named as “Data Entry” and “Shift Entry”. The shifter transfers the “Data Entry” to the same extent as “Shift Entry” and it moves toward output point.

Since the delay of this circuit equals to one transistor therefore OHRNS circuits are much faster and also the power consumption is low as a result that two signals are only active at the same time. Transistor level of this circuit is shown as bellow and the transistor delay is clear.

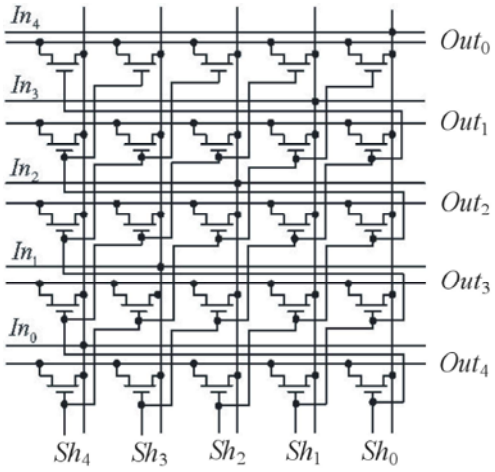

Fig. 6: One-Hot shifter for moduli 5.

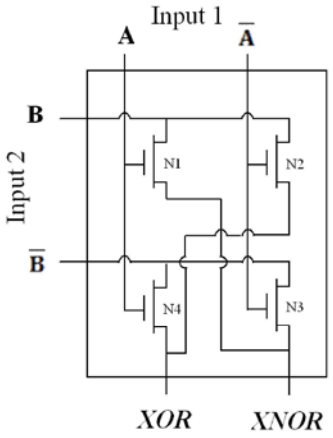

Fig. 7: The circuit of OHRNS XOR gate

Hence if we use One-Hot Multi-Level Residue Number System in the key schedule of the KASUMI block cipher, then the propagation delay and power consumption are decreased and also the speed of operation is increased.

Fig. 7 illustrates a new architecture instead of using XOR common gate in many applications that need high-speed and low-power components like KASUMI but notice that the functionality of XOR operation doesn`t change. It generates XOR and XNOR outputs simultaneously with only four transistors. This circuit that named "One-Hot Residue Number System Exclusive OR" is based on pass-transistor logic and the outputs have a good logic level for all input signals. One of the transistors will be ON when each line of two inputs are active at the same time and the result will be in the actual output. For example, when (A and $\bar{B}$) or ($\bar{A}$ and B) are driven high simultaneously then the XOR output is active.

Thus, in this implementation the propagation is only equal to one transistor delay. XNOR output is at high impedance state since N1 and N3 are OFF and it is also the same for XOR output when N2 and N4 are OFF. Hence we use this new circuit (OHRNS XOR) for computing XOR operation in the key schedule of KASUMI, then power consumption and speed of operation are improved. Notice that we can also use One-Hot Multi-Level Residue Number System on the first level of moduli for increasing efficiency and achieving more performance. We give the exact key schedule of KASUMI in Table 1 and list the values of the constants in Table 2. As we can easily see in Table 1, the key scheduler consists of thirty two 16-bit XOR, eight 1-bit cyclic left shifts, eight 5-bit cyclic left shifts, eight 8-bit cyclic left shifts and eight 13-bit cyclic left shifts. Thus, XOR operation and cyclic shift are the main and essential parts of the scheduler.

Whereas it was mentioned in the previous section, One-Hot Residue Number System is simple, rapid, low power and has regular and simple structure; moreover OHMLRNS is suitable for computing XOR logical operation and cyclic shift. So, we want to use it in the key schedule and improve performance of the KASUMI block cipher in the aforementioned terms.

Table 1: KASUMI□s Key Schedule Algorithm

Round	$KL_{i,1}$	$KL_{i,2}$	$KO_{i,1}$	$KO_{i,2}$	$KO_{i,3}$	$KI_{i,1}$	$KI_{i,2}$	$KI_{i,3}$
1	K_1 <<< 1	K_3□	K_2 <<< 5	K_6 <<< 8	K_7 <<< 13	K_5□	K_4□	K_8□
2	K_2 <<< 1	K_4□	K_3 <<< 5	K_7 <<< 8	K_8 <<< 13	K_6□	K_5□	K_1□
3	K_3 <<< 1	K_5□	K_4 <<< 5	K_8 <<< 8	K_1 <<< 13	K_7□	K_6□	K_2□
4	K_4 <<< 1	K_6□	K_5 <<< 5	K_1 <<< 8	K_2 <<< 13	K_8□	K_7□	K_3□
5	K_5 <<< 1	K_7□	K_6 <<< 5	K_2 <<< 8	K_3 <<< 13	K_1□	K_8□	K_4□
6	K_6 <<< 1	K_8□	K_7 <<< 5	K_3 <<< 8	K_4 <<< 13	K_2□	K_1□	K_5□
7	K_7 <<< 1	K_1□	K_8 <<< 5	K_4 <<< 8	K_5 <<< 13	K_3□	K_2□	K_6□
8	K_8 <<< 1	K_2□	K_1 <<< 5	K_5 <<< 8	K_6 <<< 13	K_4□	K_3□	K_7□

X <<< i - X rotated to the left by i bits

Table 2: KASUMI's Key Schedule Constants

Round	1	2	3	4	5	6	7	8
Constant	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
Value	0123_x	4567_x	$89AB_x$	$CDEF_x$	$FEDC_x$	$BA98x$	7654_x	3210_x

RESULTS AND DISCUSSION

We compare four conventional coding mechanisms by two parameters: capability of error detection/correction and burst error detection. As presented in Table 3, Multi-Level Redundant Residue Number System code is compared with parity check code, CRC code and hamming code mechanisms.

None of other error control mechanisms are capable to split the key into the small ones. One of the major advantages of MLRRNS codes is the burst error detect.

In Table 4 a comparison is made between One-Hot Multi-Level Residue Number System and other Residue Number Systems. As we can see in Table 4, with comparing to other System Numbers, we have achieved a significant improvement in terms of simplicity of moduli selection, dynamic range, calculation speed, power consumption and reliability.

In addition, we simulate proposed OHRNS XOR by using TSMC 0.90-µm technology and simulations are carried out using HSPICE. The circuits performance is evaluated in terms of worst-case delay, power consumption and power-delay product for a supply voltage 1.2V VDD at 50-MHz frequency. The delay is calculated from 50% of voltage level of input to 50% of voltage level of resulting output all the rise and fall output transitions. For the calculation of the power-delay product, worst-case delay is chosen to be the larger delay amongst the two outputs. Toward an accurate result, all the possible input combinations are considered for all the circuits.

The PDP is a quantitative measure of the efficiency of the tradeoff between power dissipation and speed. By optimizing the transistor sizes of the OHRNS XOR considered, it is possible to reduce the delay without significantly increasing the power consumption and transistor sizes can be set to achieve minimum PDP.

The proposed OHRNS XOR circuit was compared with the circuits in Goel *et al*., 2006 [29] and Hassoune *et al*., 2010 [30]. The simulation results at 1.2-V VDD and TSMC 0.90-µm technology are shown in Table 5. We consider all the possible input transitions with an output transition at every input transition.

Table 3: The comparison between error control mechanisms

Error Control mechanism	Capability	Burst error check
Parity check Code	Error detection	No
Cyclic Redundancy Check Code	Error detection	Yes
Hamming Code	Error detection and correction	No
Multi-Level Redundant Residue Number System Code	Error detection and correction	Yes

Table 4: Comparing OHMLRNS with other Residue Number Systems

Features	RNS	MLRNS	OHRNS	OHMLRNS
Select the set of moduli	Normal	Simple	Normal	Simple
Speed of calculation	Normal	Normal	High	High
Dynamic range	Normal	Large	Normal	Large
Power Consumption	Normal	Normal	Low	Low
Reliability	Medium	Good	Medium	Very High

Table 5: Simulation results for proposed OHRNS XOR-XNOR circuit in 0.90-µm technology at 50-MHz frequency and 1.2-V V_{DD}

	Goel *et al*. 2006 [29]	Hassoune *et al*. 2010 [30]	
Circuit in:	Figure 6 (c)	Figure 8 (a)	Proposed circuit
No. of Transistor	8	6	4
Power (nW)	82.91	118.26	65.70
Delay (ps)	24.85	16.8	9.31
PDP (e^{-18})	2.06	1.987	0.612
Improvement (PDP)	70%	69%	

The results indicate that the performance of the proposed circuit (OHMLRNS XOR) is better than the performance of the compared circuits. The proposed circuit is 2.7× faster than the circuit in Goel *et al*., 2006 and also consumes low power. This is due to the its structure that is inherently high power consuming but is expected to be lesser than the compared XOR circuits due to the reduced number of transistors.

The proposed circuit is 1.8× faster than circuit in Hassoune *et al*., 2010 and consumes very low power, too. Owing to the higher speed of our circuit, there is almost 69%–70% saving in PDP in this circuit. The proposed circuit uses only four transistors whereas the circuit in Goel *et al*., 2006 uses 8 and the circuit in Hassoune *et al*., 2010 uses 6 transistors.

Since it was demonstrated in the previous section, the novel OHRNS XOR circuit was utilized for the key schedule of the KASUMI block cipher and the proposed key scheduler was reached the high performance of computations with regard to speed, delay and power consumption.

When we only contemplate on the XOR gate in the key schedule and according to the 8 rounds structure that each round contains (4×16) 64 2-bit input XOR gate hardware and 1 XOR gate delay, the following results are obtained for all rounds: if it uses XOR circuit in Goel *et al*., 2006 with 8 transistors then the propagation delay is 8×1×8=64 transistors and hardware includes 8×64×8=4096 transistors; if it uses XOR circuit in Hassoune *et al*., 2010 with 6 transistors then delay is 8×1×6=48 transistors and hardware is 8×64×6=3072 transistors; whereas if it utilizes the OHRNS XOR with 4 transistors then the propagation is only equal to 8×1×4=32 transistors delay and also hardware decreases to 8×64×4=2048 transistors. So the proposed circuit is 2× and 1.5× faster than using the circuit in Goel *et al*., 2006 and Hassoune *et al*., 2010; also there is a considerable reduction in the PDP achieved by the proposed circuit. So we can easily observe the increasing of performance in the key scheduler and finally this improvement occurs for the KASUMI block cipher, too.

CONCLUSIONS

Taking everything into consideration, because of decreasing the power supply voltage of future devices, reliability will be the major challenge of VLSI design methodology. In this paper we introduced the Multi-Level Redundant Residue Number System (MLRRNS) as a numeric system and its capability of error detection and correction. We glanced at the key schedule of KASUMI block cipher architecture and then we proposed a new approach in applying MLRRNS error control mechanism into the key scheduler. The main property of Multi-Level Redundant Residue Number System is the splitting big numbers into small ones and that future is used to split keys into small ones. Another advantage of this error control mechanism is error detection/correction. Also the designers can add different redundant bits to every small key that depends on the design.

The KASUMI block cipher is used in many systems and networks of 3rd generation. The main and important section of this algorithm is the key scheduler. We also redesigned XOR gate by using One-Hot Multi-Level Residue Number System in order to increase speed and decrease power dissipation because XOR logical operation is repeatedly used in the key scheduler. This means that an efficient and high-speed key schedule of the KASUMI block cipher using OHRNS XOR was explained in this document which targets low PDP. It can outperform all the previous published designs. The proposed circuits face the needs of any manufacturer looking for a high performance ciphering circuits which is high-speed and has low power consumption. We recommend the use of OHMLRNS for the design of high-performance circuits.

ACKNOWLEDGEMENT

The author would like to thank the reviewers for their constructive comments to improve the readability and quality for this paper.

REFERENCES

1. 3rd Generation Partnership Program. 3GPP Home Page, A Global Initiative. http://www.3gpp.org.
2. Matsui, M., 1997. New Block Encryption Algorithm MISTY. *4th International Fast Software Encryption Workshop*, Haifa, Israel, pp: 54-68.
3. Kitsos, P., N. Sklavos and O. Koufopavlou, 2007. UMTS security, system architecture and hardware implementation. Wireless Communication and Mobile Computing, 7: 483-494.
4. 3rd Generation Partnership Program. Document 1: f8 and f9 Specification 35.201. Release 5. Version 5.0.0.
5. 3rd Generation Partnership Program. Document 2: KASUMI Specification. Technical Specification 35.202. Release 5. Version 5.0.0.

6. Chren, W.A., 1998. One-Hot Residue Coding for Low Delay-Power Product CMOS Design. IEEE Transactions On Circuits And Systems II: Analog And Digital Signal Processing, 45(3).
7. Hurst, S.L., 1984. Multiple-Valued Logic – Its status and its future. IEEE Transaction on Computers, pp: 1160-1179.
8. Gonzalez, A.F. and P. Mazumdar, 2000. Redundant Arithmetic, Algorithms and Implementations Integration. The VLSI Journal, 30(1): 13-53.
9. Soderstrand, M.A. and Eds, 1986. Residue Number System arithmetic: modern applications in digital signal processing. New York, *IEEE Press*.
10. Conway, R. and J. Nelson, 2004. Improved RNS FIR Filter Architectures. IEEE Transactions on Circuits and Systems II: Express Briefs, 51(1).
11. How, H.T., T.H. Liew, E.L. Kuan, L.L. Yang and L. Hanzo, 2006. A Redundant Residue Number System Coded Burst-by-Burst Adaptive Joint-Detection Based CDMA Speech Transceiver. IEEE Transactions on Vehicular Technology, 55(1): 387-396.
12. Ciet, M., M. Neve1, E. Peeters and J. Jacques, 2003. Parallel FPGA Implementation of RSA with Residue Number Systems. IEEE Proceedings of the Symposium on Circuits and Systems, 2: 806-810.
13. Bajard, J.C. and L. Imbert, 2004. A Full Implementation RSA in RNS. IEEE Transactions on Computer, 53(6): 769-774.
14. Ramirez, J., *et al*., 2002. Fast RNS FPL-Based Communications Receiver Design and Implementation. 12th Int'l Conf. Field Programmable Logic, pp: 472-481.
15. Barsi, F. and P. Maestrini, 1973. Error Correcting Properties of Redundant Residue Number Systems. IEEE Transactions on Computers, C-22(3): 307-315.
16. Mandelbaum, D., 1972. Error Correction in Residue Arithmetic. IEEE Transactions on Computers, C-21(6): 538-545.
17. Krishna, H., K.Y. Lin and J.D. Sun, 1992. A coding theory approach to error control in Redundant Residue Number Systems - Part I: theory and single error correction. IEEE Transactions on Circuits and Systems, 39: 8-17.
18. Sun, J.D. and H. Krishna, 1992. A coding theory approach to error control in Redundant Residue Number Systems -Part II: multiple error detection and correction. IEEE Transactions on Circuits and Systems, 39: 18-34.
19. Parhami, B., 2001. RNS Representation with Redundant Residues. 35th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, pp: 1651-1655.
20. Yassine, H.M., 1992. Hierachical Residue Number System suitable for VLSI Arithmetic Architectures. IEEE International Symposium on Circuits and Systems, 2: 811-814.
21. Hariri, A., K. Navi and R. Rastegar, 2005. A Simplified Modulo (2^n – 1) Squaring Scheme for Residue Number System. IEEE International Conference on Computer as a Tool, 1: 615-618.
22. Timarchi, S., K. Navi and M. Hosseinzadeh, 2006. New Design of RNS Subtractor for modulo (2^n + 1). *2th* IEEE International Conference on Information & Communication Technologies: From Theory To Applications, pp: 24-28.
23. Hosseinzadeh, M., K. Navi and S. Timarchi, 2006. Design Circuit Residue Number System in Current mode. 14th Iranian Conference of Electrical Engineering, pp: 16-18.
24. Szabo, N. and R. Tanaka, 1967. Residue Arithmetic and its Application to Computer Technology. MC-Graw-Hill. New York.
25. Etzel, M.H. and W.K. Jenkins, 1980. Redundant Residue Number Systems for Error Detection and Correction in Digital Filters. IEEE Transactions on Acoustics, Speech and Signal Processing, ASS-28(5): 538-544.
26. Hanzawa, S., T. Sakata, K. Kajigaya, R. Takemura and T. Kawahara, 2005. A Large-Scale and Low-Power CAM Architecture Featuring a One-Hot- Spot Block Code for IP-Address Lookup in a Network Router. IEEE Journal of Solid-State Circuits, 40(4).
27. Chren, W.A., 1999. Delta–Sigma Modulator with Large OSR Using the One-Hot Residue Number System. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 46(8).
28. Hosseinzadeh, M., S.J. Jassbi and K. Navi, 2007. A Novel Multiple Valued Logic OHRNS Moduli r^n Adder Circuit. International Conference on Engineering and Technology, 25: 128-132.
29. Goel, S., A. Kumar and M.A. Bayoumi, 2006. Design of Robust, Energy-Efficient Full Adders for Deep-Submicrometer Design Using Hybrid-CMOS Logic Style. IEEE Transactions on Very Large Scale Integration Systems, 14(12).
30. Hassoune, I., D. Flandre, I. O'Connor and J.D. Legat, 2010. ULPFA: A New Efficient Design of a Power-Aware Full Adder. IEEE Transactions on Circuits and Systems I: Regular Papers, 57(8).