Pipelined Sms4 Cipher Design for Fast Encryption Using Twisted BDD S-Box Architecture

Babu M[#1] Mukuntharaj C[#2] Saranya S[#3]
Assistant Professor,PSRR College of Engineering for Women,Sivaksi[#1,2]
Programmer, TATA Consultancy Services, Chennai[#3]

ABSTRACT: ***In this current fast moving world, getting the information faster is more important. My project makes it happen. SMS4 cipher based on Pipelined Twisted BDD (Binary Decision Diagram) S-box architecture can convert the plain text into cipher text as fast as other S-box architecture. SMS4 is a 128-bit block cipher used in the WAPI standard for protecting data packets in WLAN. In this project S-box architecture using Look-Up Table (LUT), Twisted BDD and Pipelined Twisted BDD were compared and proved that Encryption using S-box Pipelined Twisted BDD architecture is about 3.5 – 4 times faster than other S-box architectures. SMS4 is a symmetric key cipher, in which both the sender and the receiver use a single key for encryption and decryption. The decryption procedure of SMS4 can be done in the same way as the encryption procedure by reversing the order of the round keys.***

Keywords-SMS4 cipher; S-box; Twisted BDD architecture; Pipelined Twisted BDD architecture

I INTRODUCTION

In 2006, the Office of State Commercial Cipher Administration of China (OSCCA) released the specification of the SMS4 block cipher, which was employed in the Wide Authentication and Privacy Infrastructure (WAPI) standard to provide the data confidentiality in wireless networks. The Architectures of LUT S-box and Twisted BDD S-box were compared in this paper. The fast Encryption of plain text to cypher text is done with Twisted BDD S-box architecture.

S box: The S (substitution) box takes in 8 bits and outputs 8 bits. It is written Sbox (:).

Fundamental Operations

The two fundamental operations used by this algorithm are:

^ the bitwise XOR of two 32-bit vectors, <<< i the circular shift of a 32-bit word, with i bits shifted left.

Input and output blocks, and key

The 128-bit input block consists of four 32-bit words MK = (MK_0 , MK_1 , MK_2 , MK_3) or MK_i(i = 0, 1, 2, 3). The round key schedule, derived from the encryption key, is represented by (rk0, rk1, … , rk31), where each rk_i(i = 0, 1,….., 31) is 32 bits long. The 128-bit output block consists of four 32-bit words FK = (FK0, FK1, FK2, FK3). For decryption, the round key schedule is represented by CK = (CK0, CK1, …., CK31) or

FK_i(i = 0, …, 3), CK_i(i = 0, ….., 31).

The round function F

This algorithm uses a nonlinear substitution structure, encrypting 32 bits at a time. This is called a one-round exchange. Consider a one-round-substitution:

Let the 128-bit input block be the four 32-bit elements (X_0, X_1, X_2, X_3) ∈ $(Z^{32}{}_2)^4$, with rk ∈ $Z^{32}{}_2$, then F is given by F(X_0, X_1, X_2, X_3, rk) = X_0 ^ T(X_1 ^ X_2 ^ X_3 ^ rk)

Mixer-substitution T

T is a substitution that generates 32 bits from 32 bits T: $Z^{32}{}_2$ $Z^{32}{}_2$. This substitution is a reversible process. It consists of a non-linear substitution, τ, and a linear substitution L,

i.e., T(:) = L(τ (:)).

Non-linear substitution τ

τ applies 4 S-boxes in parallel.

Let a 32-bit input word be $A = (a_0, a_1, a_2, a_3) \in (GF(2^8))^4$, where each a_i is an 8-bit character. Let the 32-bit output word be $B = (b_0 ,b_1 ,b_2 ,b_3) \in (GF(2^8))^4$, given by $B = (b_0, b_1, b_2, b_3) = \tau (A) = (Sbox(a_0), Sbox(a_1), Sbox(a_2), Sbox(a_3))$

Linear substitution L

B ∈ $Z^{32}{}_2$, the 32-bit output word of the non-linear substitution τ will be the input word of the linear substitution L. Let C ∈ $Z^{32}{}_2$ be the 32-bit output word generated by L. Then C = L(B) = B ^ (B<<<2) ^ (B<<<10) ^ (B<<<18) ^ (B<<<24)

II SMS4 BLOCK CIPHER

SMS4 is a Chinese block cipher standard, mandated for use in protecting wireless net-works. The input, output, and key of SMS4 are each 128 bits. The algorithm has 32 rounds, each of which modifies one of the four 32-bit words that make up the block by x-oring it with a keyed function of the other three words. Encryption and decryption have the same structure except that the round key schedule for decryption is the reverse of the round key schedule for encryption. SMS4 is a 32-round iterative algorithm, and both the data block and the key size are fixed to 128 bits. The encryption flow of the SMS4 cipher is shown below

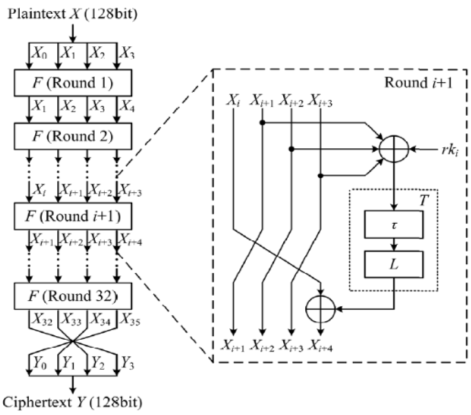

Plaintext X (128bit)
X_0 X_1 X_2 X_3
F (Round 1)
X_1 X_2 X_3 X_4
F (Round 2)
X_i X_{i+1} X_{i+2} X_{i+3}
F (Round i+1)
X_{i+1} X_{i+2} X_{i+3} X_{i+4}
F (Round 32)
X_{32} X_{33} X_{34} X_{35}
Y_0 Y_1 Y_2 Y_3
Ciphertext Y (128bit)
Round i+1
X_i X_{i+1} X_{i+2} X_{i+3}
rk_i
T
τ
L
X_{i+1} X_{i+2} X_{i+3} X_{i+4}

Fig.1 SMS4 Cipher Encryption process

SMS4 Encryption Algorithm

Let $X = (X_0, X_1, X_2, X_3) \in (GF(2^{32}))^4$ be the plaintext and $Y = (Y_0, Y_1, Y_2, Y_3) \in (GF(2^{32}))^4$ be the cipher text. Let denoted by $rk_i \in GF(2^{32})$ the round keys and by $(X_i, X_{i+1}, X_{i+2}, X_{i+3})$ the $(i+1)$-th round inputs, $i \in \{0,1,...,31\}$. Then the SMS4 scheme can be written as

$$X_{i+4} = F(X_i, X_{i+1}, X_{i+2}, X_{i+3}, rk_i)$$

$$= X_i \wedge T(X_{i+1} \wedge X_{i+2} \wedge X_{i+3} \wedge rk_i)$$

And $(Y_0, Y_1, Y_2, Y_3) = (X_{35}, X_{34}, X_{33}, X_{32})$ where $i \in \{0,1,...,31\}$, F is the round function and T is the composite transformation. The transformation T: $GF(232) \rightarrow GF(232)$ is composed of the nonlinear transformation τ and the linear transformation L: $T\ (.) = L\ (\tau\ (.))$ The transformation τ includes four 8-bit nonlinear S-boxes in parallel. Let denoted by $A = (a_0, a_1, a_2, a_3) \in (GF(2^8))^4$ the input of and τ by $B = (b_0, b_1, b_2, b_3) \in (GF(2^8))^4$ the output. Then can be defined as

$$B = (b_0, b_1, b_2, b_3) = \tau\ (A)$$

$$= (Sbox(a_0), Sbox(a_1), Sbox(a_2), Sbox(a_3))$$

where $Sbox(.)$ is the S-box byte substitution.

The output of τ, B, is also the input of the linear transformation L. Let denoted by $C \in GF(2\ 32\ )$ the output of L. Then L can be defined as

C=L(B)=B^(B<<<2)^(B<<<10)^(B<<<18)^

(B<<<24)

where << i denotes a 32-bit cyclic left shift by i positions. SMS4 is a symmetric key cipher, in which both the sender and the receiver use a single key for encryption and decryption. The decryption procedure of SMS4 can be done in the same way as the encryption procedure by reversing the order of the round keys.

III Key Schedule

The key schedule process of SMS4 cipher has the same structure as that in the encryption process except for L function. Let $MK = (MK_0, MK_1, MK_2, MK_3) \in (GF(2^{32}))^4$ denote the cipher key, $rk_i \in GF(2^{32})$, $i \in \{0,1,...,31\}$ denote the round keys, and $K_i \in GF(2^{32})$, $i \in \{0,1,...,35\}$. Then key schedule algorithm is defined as

$(K_0,K_1,K_2,K_3)=(MK_0 \wedge FK_0, MK_1 \wedge FK_1, MK_2 \wedge FK_2, MK_3 \wedge FK_3)$ and

$rk_i = K_{i+4} = K_i \wedge T'\ (K_{i+1}, K_{i+2}, K_{i+3}, CK_i)$

where FK_i , $i \in \{0,1,2,3\}$ are system parameters, CK_i , $i \in \{0,1,2,3\}$ are key constants, and T' is a transformation similar to T in the encryption process. The only difference between T and T' is the linear transformation. Instead of L, the following transformation L' is used in T':

$L'(B) = B \wedge (B<<<13) \wedge (B<<<23)$

The system parameters FK_i are defined in hexadecimal as

FK_0 = 0xa3b1bac6, FK_1 = 0x56AA3350

FK_2 = 0x677D9197, FK_3 = 0xB27022DC

The key constants $CK_i = (ck_{i,0}, ck_{i,1}, ck_{i,2}, ck_{i,3}) \in (GF(2^8))^4$ can be computed as follows:

$$ck_{i,j} = (4 * i + j) * 7 (\text{mod } 256)$$

where $i \in \{0,1,...,31\}$, and $j \in \{0,1,2,3\}$.

IV S – box

The algebraic structure of the S-box can be described as $Sbox(a) = I\ (a.\mathbf{A_1} + C_1\ )\mathbf{A_2} + C_2$ where $I(.)$ is the patched inversion over $GF(2^8)$, the matrices $\mathbf{A_1}, \mathbf{A_2} \in GL(8,2)$, and the vectors $C_1, C_2 \in GF(2)^8$. The cyclic matrices and the row vectors in:

$$\mathbf{A_1} = \mathbf{A_2} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

$C\ 1 = C\ 2 = (1, 1, 0, 0, 1, 0, 1, 1)$

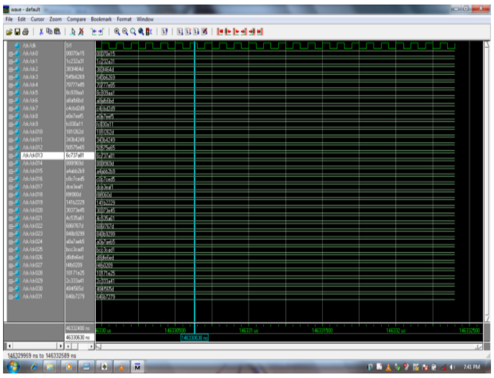
The irreducible polynomial is

$f\ (x) = x^8 + x^7 + x^6 + x^5 + x^4 + x^2 + 1$

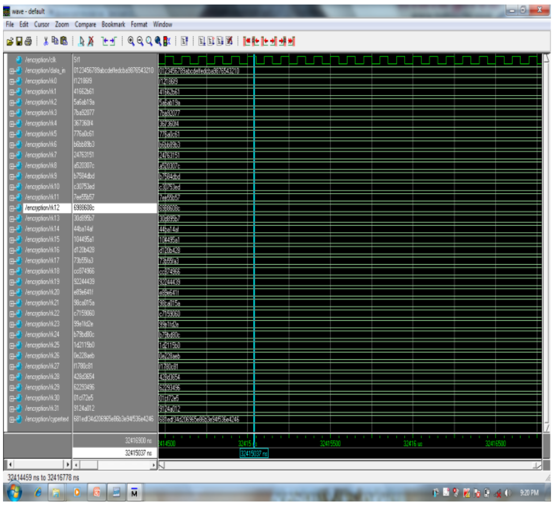
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	D6	90	E9	FE	CC	E1	3D	B7	16	B6	14	C2	28	FB	2C	05
1	2B	67	9A	76	2A	BE	04	C3	AA	44	13	26	49	86	06	99
2	9C	42	50	F4	91	EF	98	7A	33	54	0B	43	ED	CF	AC	62
3	E4	B3	1C	A9	C9	08	E8	95	80	DF	94	FA	75	8F	3F	A6
4	47	07	A7	FC	F3	73	17	BA	83	59	3C	19	E6	85	4F	A8
5	68	6B	81	B2	71	64	DA	8B	F8	EB	0F	4B	70	56	9D	35
6	1E	24	0E	5E	63	58	D1	A2	25	22	7C	3B	01	21	78	87
7	D4	00	46	57	9F	D3	27	52	4C	36	02	E7	A0	C4	C8	9E
8	EA	BF	8A	D2	40	C7	38	B5	A3	F7	F2	CE	F9	61	15	A1
9	E0	AE	5D	A4	9B	34	1A	55	AD	93	32	30	F5	8C	B1	E3
A	1D	F6	E2	2E	82	66	CA	60	C0	29	23	AB	0D	53	4E	6F
B	D5	DB	37	45	DE	FD	8E	2F	03	FF	6A	72	6D	6C	5B	51
C	8D	1B	AF	92	BB	DD	BC	7F	11	D9	5C	41	1F	10	5A	D8
D	0A	C1	31	88	A5	CD	7B	BD	2D	74	D0	12	B8	E5	B4	B0
E	89	69	97	4A	0C	96	77	7E	65	B9	F1	09	C5	6E	C6	84
F	18	F0	7D	EC	3A	DC	4D	20	79	EE	5F	3E	D7	CB	39	48

Table.1 SMS4 S-box Look-Up Table

Simulation result of Encryption using LUT S-box Architecture Timing Diagram of key Constant Generation



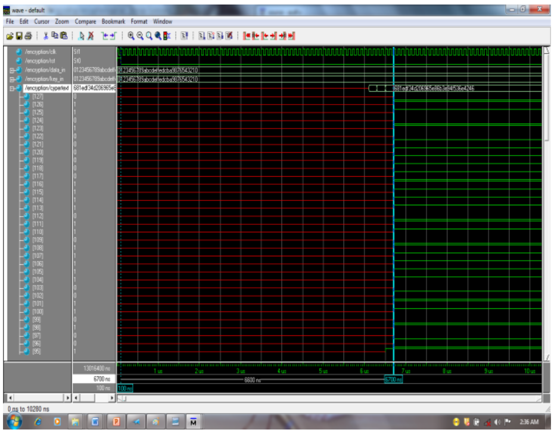
Timing Diagram of Round Key Generation



```
/encryption/clk        St1
/encryption/data_in    0123456789abcdeffedcba9876543210
/encryption/rk0        f1218689
/encryption/rk1        41662b61
/encryption/rk2        5a6ab19a
/encryption/rk3        7ba92077
/encryption/rk4        367360f4
/encryption/rk5        776a0c61
/encryption/rk6        b6bb89b3
/encryption/rk7        24763151
/encryption/rk8        a520307c
/encryption/rk9        b7584dbd
/encryption/rk10       c30753ed
/encryption/rk11       7ee55b57
/encryption/rk12       6988608c
/encryption/rk13       30d895b7
/encryption/rk14       44ba14af
/encryption/rk15       104495a1
/encryption/rk16       d120b428
/encryption/rk17       73b55fa3
/encryption/rk18       cc874966
/encryption/rk19       92244439
/encryption/rk20       e89e641f
/encryption/rk21       98ca015a
/encryption/rk22       c7159060
/encryption/rk23       99e1fd2e
/encryption/rk24       b79bd80c
/encryption/rk25       1d2115b0
/encryption/rk26       0e228aeb
/encryption/rk27       f1780c81
/encryption/rk28       428d3654
/encryption/rk29       62293496
/encryption/rk30       01cf72e5
/encryption/rk31       9124a012
/encryption/cypertext  681edf34d206965e86b3e94f536e4246

32416900 ns
32415037 ns
32414459 ns to 32416778 ns
```

Timing diagram of Encryption using LUT S-box Architecture



```
/encryption/clk        St1
/encryption/rst        St0
/encryption/data_in    0123456789abcdeff
/encryption/key_in     0123456789abcdeff
/encryption/cypertext  681edf34d206965e6
  [127]  0
  [126]  1
  [125]  1
  [124]  0
  [123]  1
  [122]  0
  [121]  0
  [120]  0
  [119]  0
  [118]  0
  [117]  0
  [116]  1
  [115]  1
  [114]  1
  [113]  1
  [112]  0
  [111]  1
  [110]  1
  [109]  0
  [108]  1
  [107]  1
  [106]  1
  [105]  1
  [104]  1
  [103]  0
  [102]  0
  [101]  1
  [100]  1
  [99]   0
  [98]   1
  [97]   0
  [96]   0
  [95]   1

13016400 ns
6700 ns
100 ns
0 ns to 10280 ns
```

ENCRYPTION USING TWISTED BDD S-BOX ARCHITECTURE

The encryption using Twisted BDD S-box Architecture is also same as that of the LUT S-box architecture. Only difference is the structure of transformation box τ.

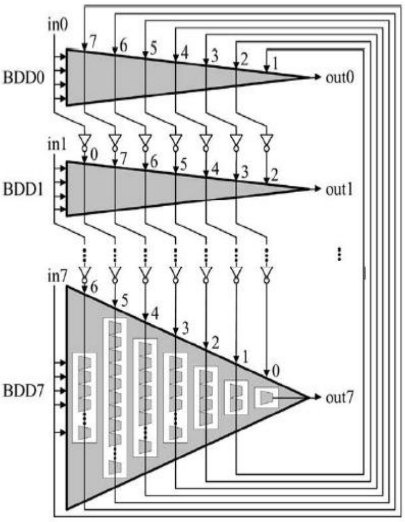


Fig.2 Twisted BDD Architecture

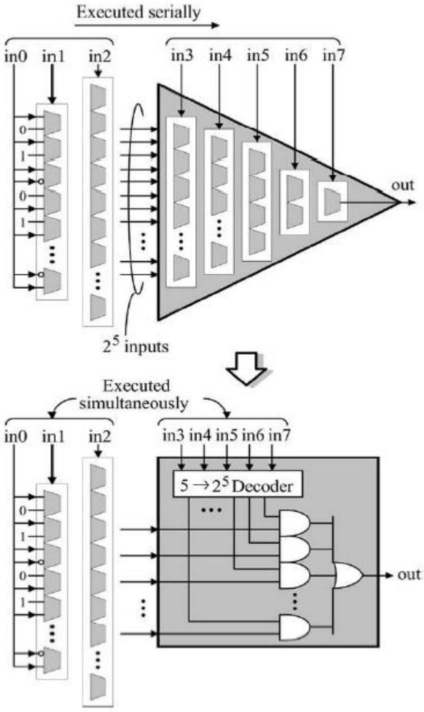


Fig.3 Speedup by parallel decoding of selector control signal

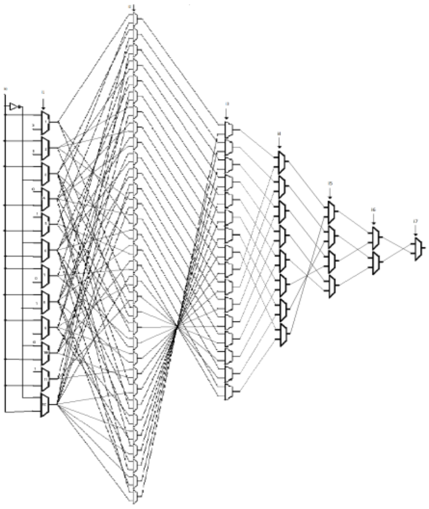


Fig.4 Twisted BDD S-Box Architecture with m=0

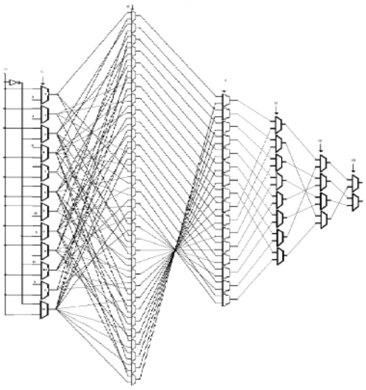


Fig.5 Twisted BDD S-Box Architecture with m=1

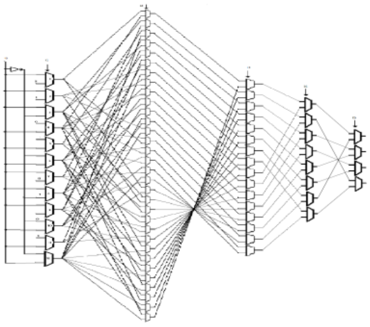


Fig.6 Twisted BDD S-Box Architecture with m=2

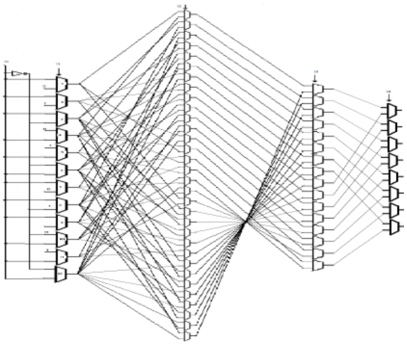


Fig.7 Twisted BDD S-Box Architecture with m=3

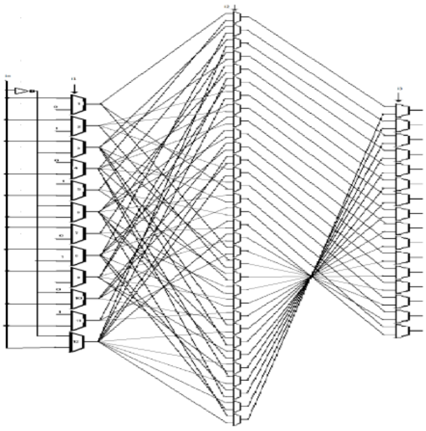


Fig.8 Twisted BDD S-Box Architecture with m=4

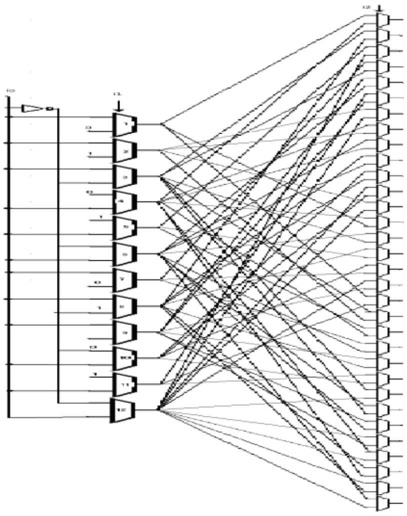
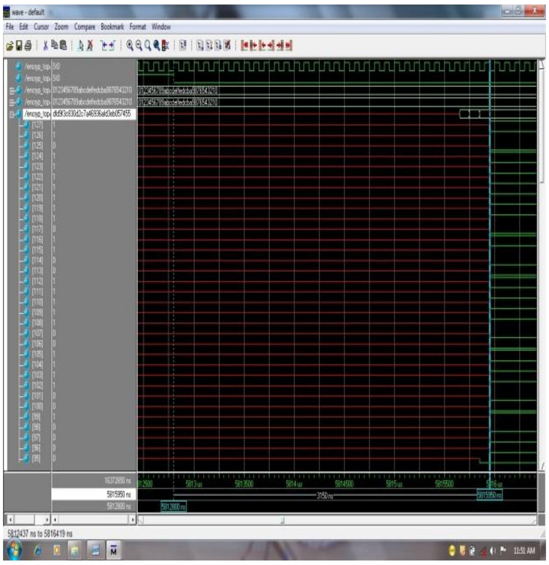


Fig.8 Twisted BDD S-Box Architecture with m=4

Simulation Result of Encryption using Twisted BDD S-box Architecture


wave - default
File Edit Cursor Zoom Compare Bookmark Format Window
/encryp_top
/encryp_top
/encryp_top 0123456789abcdeffedcba9876543210
/encryp_top 0123456789abcdeffedcba9876543210
/encryp_top dfd9f3c830d2c7a46936afd3eb057455

ENCRYPTION USING PIPELINED TWISTED BDD S-BOX ARCHITECTURE

Pipelining is the process of adding delay element on the flow of signal path. This will improve the speed of the circuit. Thereby we can achieve better performance. D-FF is used as delay element in this circuitry.

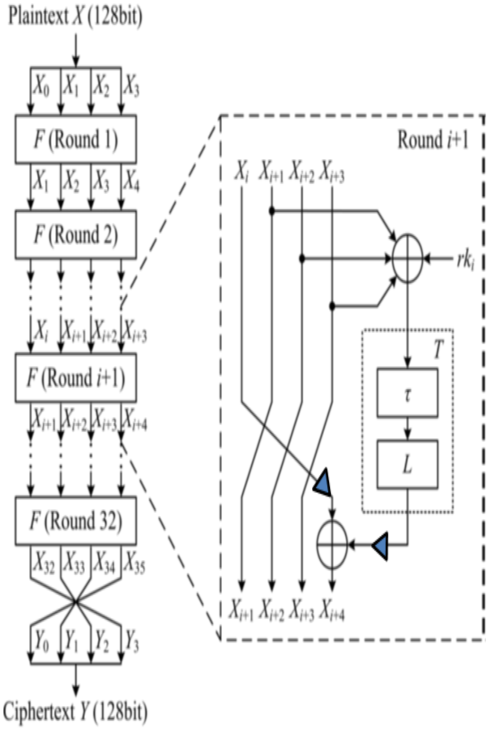
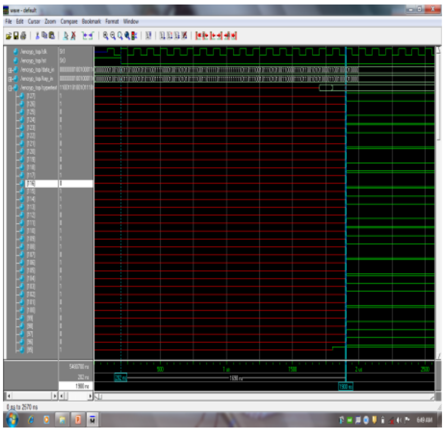


Fig.9 Pipelined SMS4 Cipher Encryption process

Simulation Result of Encryption using Twisted BDD S-box Architecture



V Performance Comparison

ARCHITECTURE	MEMORY NEEDED	DELAY
S-Box LUT	1.03Gb	6600ns
Twisted BDD with m = 0	4.31Gb	3350ns
Twisted BDD with m = 1	3.12Gb	5100ns
Twisted BDD with m = 2	4.26Gb	3365ns
Twisted BDD with m = 3	4.37Gb	3300ns
Twisted BDD with m = 4	4.81Gb	3296ns
Twisted BDD with m = 5	4.37Gb	3300ns
Pipelined SMS4 Cipher	6.18Gb	1698ns

Table.2 Performance Comparison

VI CONCLUSION

This paper purely deals about, how fast we can convert the plain text into cipher text. The SMS4 cipher design using LUT S-box architecture, Twisted BDD S-box architecture and using Pipelined Twisted BDD S-box architecture are designed and proved that Encryption using Pipelined Twisted BDD S-box architecture is about 3.5 – 4 times faster than all other S-box Architecture.

Future work

The Pipelined Twisted BDD architecture improved the operation speed of the S-box circuit. On the other hand, we can find that much of the critical path delay is used by other operations other than S-box, including XORs, multiplexors and setup time required by the technology. So, future works on improving the SMS4 circuit speed can be focused on the fast circuit architecture design of other parts in SMS4 cipher.

REFERENCES

[1] X. Bai, L. Guo, and T. Li, “Differential power analysis attack on SMS4 block cipher,” in Proceedings of 4th IEEE International Conference on Circuits and Systems for Communications, ICCSC 2008, Shanghai, China, May 2008, pp. 613–617.

[2] L. Zhang and W. Wu, “Differential fault analysis on SMS4 (in Chinese),” Chinese Journal of Computers, vol. 29, no. 9, pp. 1596–1602, 2006.

[3] W. Li and D. Gu, “An improved method of differential fault analysis on the SMS4 cryptosystem,” in Proceedings of 1st International Symposium on Data, Privacy, and E-Commerce, ISDPE 2007, Chengdu, China, Nov. 2007, pp. 175–180.

[4] S. Morioka and A. Satoh, “A 10-Gbps full-AES crypto design with a twisted BDD S-box architecture,” IEEE Trans. VLSI Syst., vol. 12, no. 7, pp. 686–691, Jul. 2004.