International Journal of Modern Engineering Research (IJMER)
www.ijmer.com Vol. 3, Issue. 5, Sep - Oct. 2013 pp-3020-3025 ISSN: 2249-6645

Implementation of Low Power and High Speed encryption Using Crypto-Hardware

G.VijayaBharathi, *K.VasuBabu, M.tech,*
Asst.Prof, VVIT, Nambur, Guntur.

ABSTRACT: *Cryptographic algorithms such as International Data Encryption Algorithm (IDEA) have found various applications in secure transmission of the data in networked instrumentation and distributed measurement systems. Modulo 2n +1 multiplier and squarer play a pivotal role in the implementation of such crypto-algorithms. In this work, an efficient hardware design of the IDEA (International Data Encryption Algorithm) using novel modulo 2n + 1 multiplier and squarer as the basic modules is proposed for faster, smaller and low-power IDEA hardware circuits. Novel hardware implementation of the modulo 2n + 1 multiplier is shown by using the efficient compressors and sparse tree based inverted end around carry adders is given. The novel modules are applied on IDEA algorithm and the resulting implementation is compared both qualitatively and quantitatively with the IDEA implementation using the existing multiplier/squarer implementations. Experimental measurement results show that the proposed design is faster and smaller and also consume less power than similar hardware implementations making it a viable option for efficient hardware designs.Yet,despiteitssophistication,manyfutureattemptsat crackingDESshowedsignificant signsofsuccess.Forexample, thedistributive computing approachofspreadingcracking computationpoweroverthe Internet earnedRockeVerserandMichael Sandersthe prize ofthe1997DESChallenge.DESChallenge II wasalso crackedthefollowingyear.WiththeinventionoftheElectronic FrontierFoundationDESCracker, itwasshownthat a 56-bit keyprotectionisinsufficient againstexhaustivesearchemployedwithtoday'stechnology.Therefore,there wasanurgentcallforastrongersecret-keyencryptionalgorithm.IDEAwasone ofthe algorithmstoanswerthatcall.*

Key words —modulo 2*n* + 1 multiplier; International Data Encryption Algorithm (IDEA); Sparse-tree adder; Power/area/speed measurement;

I. INTRODUCTION

Thedemand for highsecurityincommunicationschannels,networkedInstrumentationanddistributedmeasurement systemsisevergrowing rapidly.Theconfidentiality andsecurityrequirementsarebecomingmoreandmoreimportant to protectthedata transmittedandreceived. Thisleadsto the need for efficient design of cryptographic algorithms whichofferdataintegrity, authentication, non-repudiationand confidentialityof the encrypteddataacrossthe communication channels.Variouscryptographic algorithmshavebeenstudied andimplemented toensuresecurityofthesesystems.Inthis paper,modulo2n+1multiplier hasbeenmuchfocusasithas found itsimportant roleinIDEAalgorithm. Forexample,the threemajoroperationsthatdecidetheoverallperformanceand delayoftheIDEA [1,4,15]aremodulo2^{16} addition,bitwise XOR and modulo2^{16}+1multiplicationand the GF(2^n) Montgomerymultiplicationandmodularexponentiationcan beimplementedusingrepeatedmultiplication andsquaringof thevectors.Amongtheseoperations, improvingthedelayandpowerefficiencyofthemodulo2^n +1multiplicationoperation leadstosignificant increaseintheperformanceoftheentirecryptographiccipher.

Numeroushardware implementationsofthe IDEAalgorithmareproposedintheliteratureusingdifferentmodulo 2^{16}+1multiplierarchitectures.TheIDEAalgorithmhasbeen implementedinsoftware[3]onIntelPentiumII445MHzwith encryptionrateof23.53Mb/Sec.Later,IDEAwasrealized on hardwarechipbyCurigeret al [1]withencryptionratesup to 177Mb/sec.Byusingabit-serialimplementation [4],which enablestheIDEAtobefully,pipelinedtheencryption rates reached500Mb/sec with125MHzclockrate.Theefficiency oftheIDEAciphercanstillbeimprovedifefficient basic modules suchasmodulomultipliers and addersareused. Theefficient implementationofthemodulo2^n+1multiplier basedonnovelcompressorsandsparsetreebasedinverted end around carry adders is presentedin [7]. Even though thearchitecture ofthemodulomultiplierisveryefficiently proposedin[6],thehardwareimplementationandoptimization areconsiderablyimprovedin[7].Thisisresulted byreplacing thefulladder arrayswiththenovelcompressors andthefinal stageadderwiththesparsetreebasedinvertedendaround carryadder.

Thepaperisorganizedasfollows;Section II-A introducesmultiplexer-basedcompressors. InSectionII-B,thehardware implementationofmodulo2^n+1multiplierisgiven.Section IIIdiscussestheproposedimplementationoftheidEAcipher whichusesmodulo2^{16} +1multiplier.Acomparisonofourimplementationtoa recentlyproposedimplementationismade inSectionIV.Ourconclusionsaredrawninsection V.

II. PRELIMINARIES AND REVIEW

Novelmultiplexor(MUX)basedcompressorsand2^n +1 multiplierdesignhavebeenreported in[7]andarebriefly reviewedinthissectionasfollows:

A. Compressors

1) MUXvs.XOR:

ExistingCMOSdesignsof2-1MUX and2-inputXORareshowninFig.1.According to[8], the CMOSimplementation ofMUXperformsbetterintermsofpoweranddelaycomparedtoXOR.Suppose,XandY are

inputstotheXORgate,theoutputisXY+XY.Thesame XORcanbeimplementedusingMUXwithinputsX,Xand selectbitY.Theefficientimplementationofcompressors[9] isachievedbyusingbothoutputanditscomplement ofthese gates.Thisalsoreducesthetotalnumberofgarbageoutputs.

2) DescriptionofCompressors:

A(p,2)compressorwith pinputsX1,X2 ...Xp andtwooutputbitsSumandCarry alongwithcarryinputbitsandcarryoutputbitsis governedbytheequation:

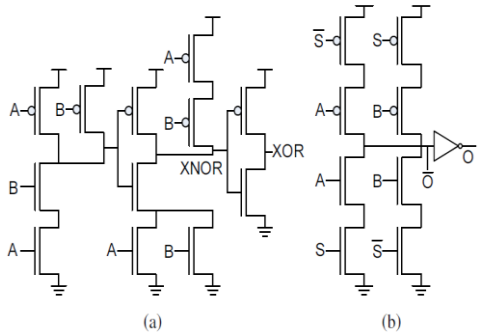

Fig. 1: CMOS implementation of 2-input (a) XOR (b) MUX

$$\sum_{i=1}^{p} X_i + \sum_{i=1}^{t}(C_{in})_i = Sum + 2(Carry + \sum_{i=1}^{p}(C_{out})_i)$$

Blockdiagramofa5:2compressorisshowninFig.2.Efficient designoftheexistingXOR-based 5:2compressor [10,11], whichtakes5inputs and2carryinputs, isshowninFig.3(a). Thecriticalpathdelayofthisexistingcompressoris4Δ-XOR (delaydenotedbyΔ).

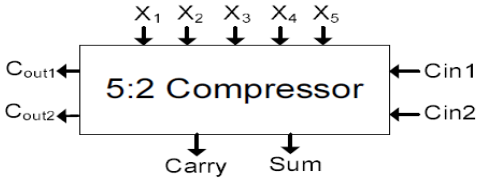

Fig. 2: Block Diagram of a 5:2 compressor

Thenewlydesignedcompressorsusemultiplexersinplace ofXORgates,resultinginhighspeedarithmetic. Also,as showninFig.3(a)inalltheexistingCMOSimplementations oftheXORandMUX gatesboththeoutput anditscomplementareavailablebutthedesignsofcompressors availablein literaturedonotusetheseoutputsefficiently. IntheCMOS implementation of the MUX if both the selectbit and its complementaregeneratedinthepreviousstage thenitsoutput isgeneratedwithmuchless delaybecausetheswitchingofthe transistorisalreadycompleted. Andalsoifboththeselectbit anditscomplementaregeneratedinthepreviousstagethentheadditionalstage oftheinverteriseliminatedwhichreducesthe overalldelayinthecriticalpath. ThenewMUX-based design of5:2compressors[9] isshownin Fig.3(b),the delayof which isΔ-XOR+3Δ-MUX CGEN blockusedinFig.3(b) canbe obtainedfromtheequationCout1=(x_1+x_2)·x_3+x_1·$x_2$andtheCMOSimplementationisgiveninFig.4.

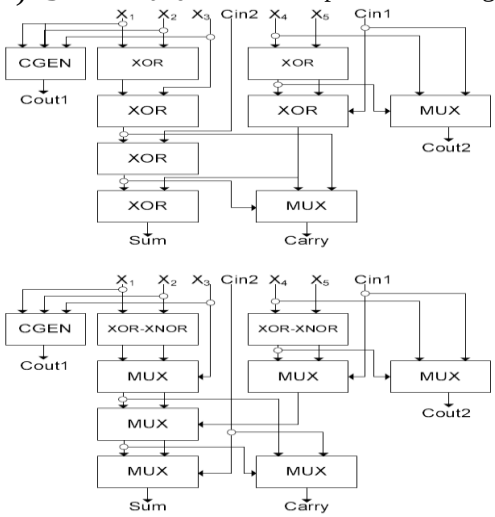

Fig. 3: 5:2 compressors: (a) existing XOR-based design; (b) new MUX-based design

B.Hardware Implementation of the mod 2^n+1 Multi plier/Squarer

The hardwareimplementationofthe modulomultiplier consistsofthreemodules.Firstmoduleistogeneratepartial products,second module is to reduce the partial products to two final operands and the last module is to add the Sum and Carry operands from partial products reduction to get the final result.

1) Partial products generation: The n × n partial products matrix is obtained from the n + 1-bitinput vectors. This partialproduct matrix is generated after repositioning the bits of the initial partial product matrix based on several observations presented in [6]. The final partial products matrix after applying all the observations is shown in Fig. 5. The partialproductbits can becomputedfrom AND,OR andNOT gates. The most complexfunctionof partialproduct generationmoduleis$p_{n-1,n-1}V_{q_{n-2}}$,where$p_{i,j}=a_ib_j$and$q_i=p_{n,i}Vp_{i,n}$

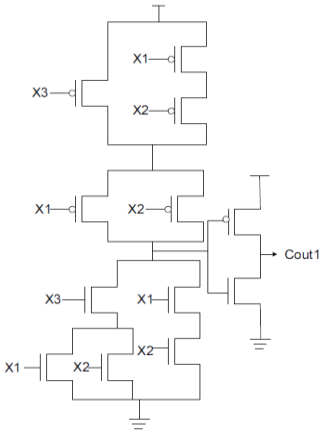

Fig. 4: CMOS implementation of carry generator block (CGEN) for the proposed design.

2^{n-1}	2^{n-2}	2^{n-3}
$PP_0 = p_{n-1,0} \vee q_{n-1}$	$p_{n-2,0}$	$p_{n-3,0}$
$PP_1 = p_{n-2,1}$	$p_{n-3,1}$	$p_{n-4,1}$
$PP_2 = p_{n-3,1}$	$p_{n-4,2}$	$p_{n-5,2}$
....	....	....
$PP_{n-2} = p_{1,n-2}$	$p_{0,n-2}$	$\overline{p_{n-1,n-2} \vee q_{n-3}}$
$PP_{n-1} = p_{0,n-1}$	$\overline{p_{n-1,n-1} \vee q_{n-2}}$	$p_{n-2,n-1}$

2^2	2^1	2^0
$p_{2,0}$	$p_{1,0}$	$p_{0,0} \vee q_{n-1} \vee p_{n,n}$
$p_{1,1}$	$p_{0,1}$	$\overline{p_{n-1,1} \vee q_0}$
$p_{0,2}$	$\overline{p_{n-1,2} \vee q_1}$	$p_{n-2,2}$
....	....	....
$\overline{p_{4,n-2}}$	$\overline{p_{3,n-2}}$	$\overline{p_{2,n-2}}$
$p_{3,n-1}$	$p_{2,n-1}$	$p_{1,n-1}$

Fig. 5: Final n×n partial product matrix showing 2^{n-1}, 2^{n-2}, 2^{n-3} ... 2^2, 2^1 and 2^0 columns.

2) Partialproductsreduction: Thisisthemostimportant modulewhichlargelydeterminesthecriticalpathdelayand theoverallperformance ofthemultiplier.Hencethismodule needsto be designedso as to get minimumdelayandconsume lesspower. Theimplementations fromtheliterature[5,6,13]usefull adders(FA)andhalfadders(HA)toconstructthismodule. Theseriesoffulladdersinanycolumncanbereplacedby thenovelcompressorsthattakethesamenumberofinputs.In theproposedimplementationuseofsuggestedcompressorsis done which not only reduces the delay and power consumptionbut also the area of the circuit. For a modulo 2^{16} + 1multiplier in IDEA cipher the Carry Save Adder (CSA) arrayimplementation using Full Adders requires fifteen full addersin series in any column, these fifteen full adders can bereplaced by two 7:2 compressors, one 5:2 compressor and two3:2 compressors.

Correction factor computation is an important step while generating the partial products matrix. The full adder implementation [6] and the compressor based implementations [7]result in the same value. Because of the space constraints, computation of the correction factor COR for full adder implementation [6] is not given in this paper. COR computation for compressor implementation involves computingonly COR2, because COR1 is obtained based on repositioning of the partial productterm, which issame for both implementations. The correctionfactor COR2 computation for FA implementation which has n-1 stages of additions is shown in [6]. And the COR2 computation for the proposed multiplier implementation using the compressors also yields the same result. Since, any (p, 2) compressor can be primarily designed using *p*- 2 FAs which give *p* - 2 carry outs with 2^nweight. Hence, the overall correction factor COR computation for CSA array FA implementation and compressor implementation yield the same result i.e., 3 as shown in [5].

3) Final Stage addition: The partial product reduction module gives one n-bit carry vector and one n-bit sum vector which need to be added in the final stage addition module. Very efficient parallel prefix adders are designed to do this operation [2].

Suppose S and C are sum and carry vectors produced after the partial product reduction section. As it is shown in the work of Zimmerman [2] that

$$|S + C + 1|_{2^n+1} = |S + C + \overline{Cout}|_{2^n} \quad (1)$$

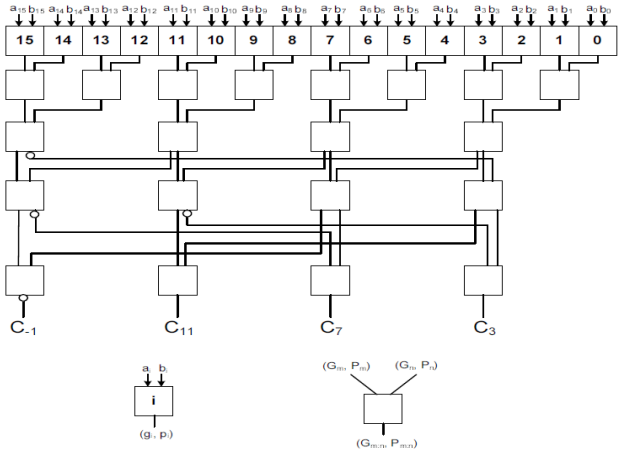

Fig. 6: Inverted EAC adder implemented using sparse tree structure

Equation (1) can be implemented using an invertedEnd-Around-Carry adder [2, 5, 6]. Even though the propagation delay of this adder is in the order of *log2n*, it has a drawback of high interconnect complexity and high fan-out. This can be overcome by sparse tree adder [12, 16] based on the prefix network logic. The sparse tree adder generates the carry for every four bits instead of generating it at every stage and using a carry select block for selecting the final carry after the prefix network. This sparse tree adder was proven to be much more efficient in terms of both delay and power when compared with the existing prefix tree based adders [14]. Hence this sparse tree can be used to design Inverted-End-Around-Carry adder. The newly designed Inverted-End-Around-Carry adder using sparse tree adder structure is shown in Fig. 6. This Inverted-EAC adder is used in the final stage addition of the modulo 2*n*+1 multiplier. Theproposed implementation of the modulo 216 + 1 multiplier for IDEA cipher is shown in Fig. 7 and $R_{16}R_{15}:::R_2R_1R_0$ represents the final product of the modulo 216 + 1 multiplier.

III. NOVEL IMPLEMENTATION OF IDEA CIRCUIT USINGTHE PROPOSED MODULO 2^N+ 1 MULTIPLIER/SQUARER

The modulo 2*n* + 1 computation is an integral part of the International Data Encryption Algorithm (IDEA) where *n* =16 [1, 4, 15]. Three major operations that decide the overall delay and performance of IDEA cipher are:

1) Modulo 216 addition;
2) Bitwise-XOR;
3)Modulo 216 + 1 multiplication/squaring.

As the first two operations take less time and are easy to implement, the delay and power efficiency of the entire IDEA cipher depends significantly on the modulo 216 + 1 multiplication/squaring operation. Hence, the IDEA cipher is implemented using the proposed modulo multiplier and compared with the existing implementations.

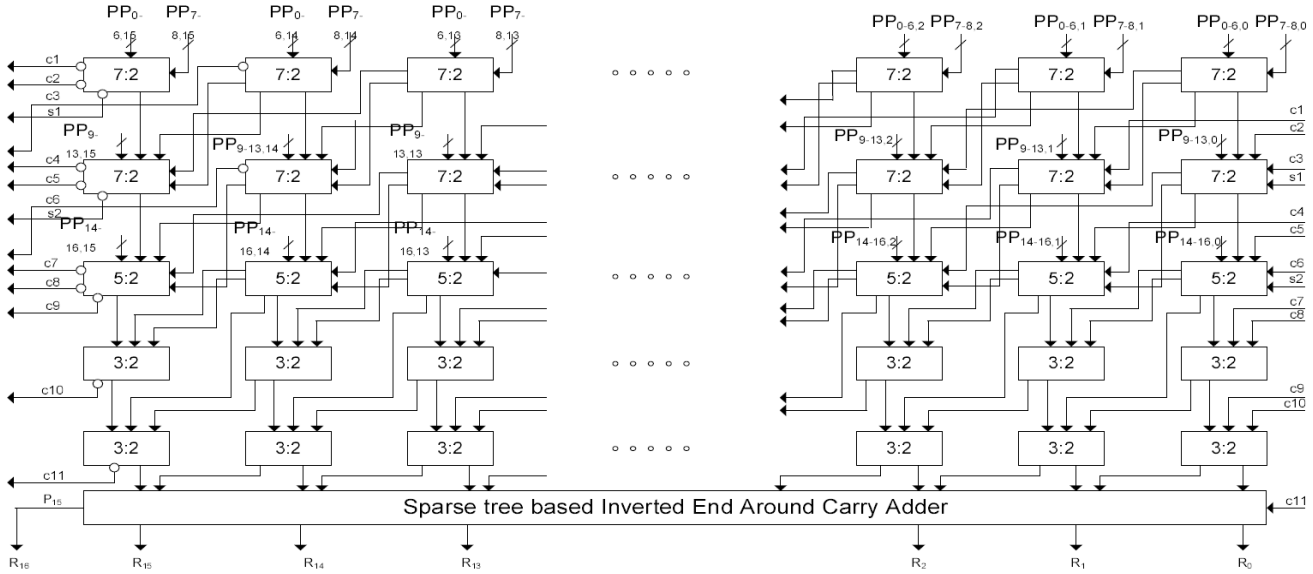

Fig. 7: Novel implementation of the modulo 2^{16}+1 multiplier using efficient compressors

Toencrypt adatablockusingIDEAcipher,thedatashould be processed through three modulo multiplication operationsin a single round and the manipulated data again shouldPass through seven such rounds iteratively and a final outputtransformation to produce the final encrypted output. TheIDEA cipher takes 64-bit input data and produces a 64-bitcipher text with a 128-bit key. The encryption and decryptionalgorithms in IDEA are almost identical except they utilizetwo different sets of sub key generated by the same key with different processes. The IDEA encryption and decryptionprocesses consist of eight rounds of data manipulation usingsubkeys and a final output transformation stage. In this cipher,all the operations are carried out on 16-bit sub-blocks. In theencryption process, the input data block of 64-bits is dividedinto 4 sub blocks of 16-bits each (*X1;X2;X3;X4*). 52 sub-keys for the encryption process are generated from the original128-bit key by shifting a part of it. Out of the 52 subkeys, sixdifferent subkeys (i.e., $Z_1^{(r)};Z_2^{(r)};Z_3^{(r)};Z_4^{(r)};Z_5^{(r)}$ and $Z_6^{(r)}$,where *r* is the roundnumber) are used for each round and theremaining 4 subkeys are used in the final output transformationstage. The 16-bit outputs at each round are represented as$Y_1^{(r)}$; $Y_2^{(r)}$; $Y_3^{(r)}$; $Y_4^{(r)}$and W_1; *W2; W3; W4* are the outputs of the final output stage transformation. The 52 subkeys used for the decryption process are obtained using a different algorithm[17]. As shown in Fig. 8, the critical path consists ofthreemodulo 216 + 1 multiplication operations, two modulo 216addition operations and two 16-bit XOR operations in eachround. In the final output transformation stage, critical path consists of a single modulo 216 + 1 multiplication operation.The throughput of the IDEA cipher can be improved, if thedelay of the modulo 216+1 multiplication operation is reduced in the pipelined implementation of the IDEA cipher. Fig. 8shows the data path of encryption process of the IDEA cipherand datapath of a single round with 4 pipeline stages with the proposed modulo multiplier.

IV. EXPERIMENTAL SIMULATIONAND RESULTS

The proposed design of the IDEA cipher with four pipelinestages using novel modulo 2*n*+1 multipliers is used to analyse and compare with the well-known IDEA cipher implementations. The use of the novel modulo multiplier improves thethroughput and performance of the IDEA cipher significantly.

A. Simulation environment

All the simulations have been carried out using MentorGraphics ASIC (Application-Specific IC) design suite. Theproposed IDEA cipher design is specified using Verilog HDLand the multiplier descriptions are mapped on a 0.18 _mCMOS standard cell library usingLeonardo Spectrum synthesis tool from Mentor Graphics.

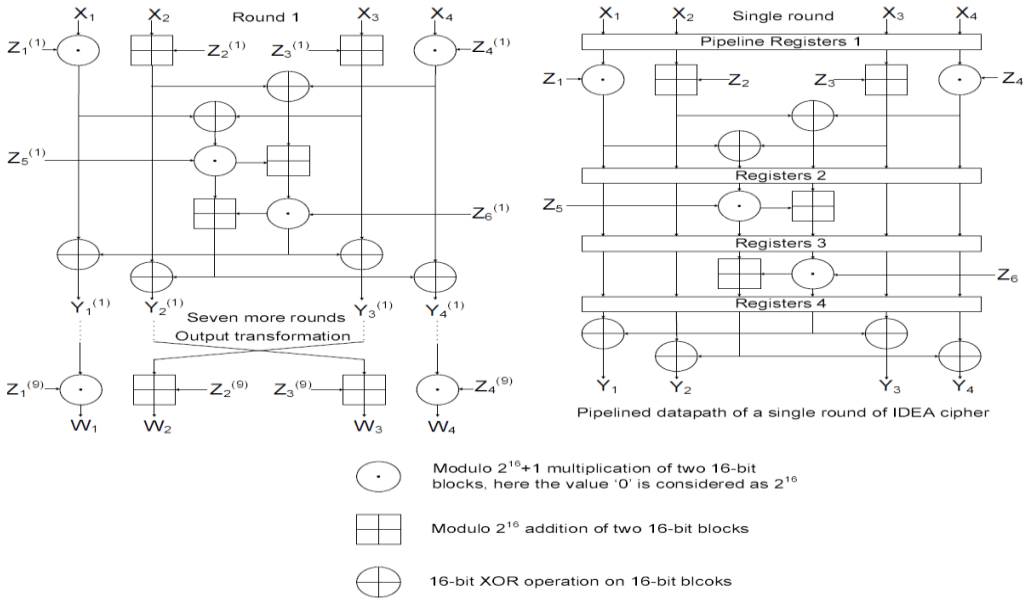

Fig. 8: Datapath of IDEA cipher with 4 pipeline stages

The design is optimized forhigh speed performance. Netlists generated from synthesis toolare passed on to standard route and place tool; the layouts areiteratively generated to get the circuits with minimum area.The calculation of power and delay are carried out using theEldosimulation tool. The proposed experimental simulationhas been performed at 1.8V with all inputs fed at a frequency of 25MHz.

B. Simulation results

The IDEA cipher is implemented using both the proposedmultiplier and the multipliers presented in [6]. Various performancemeasurements includingencryptionrate,delayand areafortheIDEAcipherusingboththeproposedmultiplier and theexisting multiplierareparametricallyobtainedand listed inTableI. As expected, theproposedIDEA circuit implementationachievessignificantimprovementsintermsof throughput(i.e.,encryption rate),latency(i.e.,criticalpath delay)andarea(i.e.,circuitarea).

TABLE I: Comparison of the performance measurements for IDEA cipher

Performance Measurement	Using proposed multipliers	Using the multipliers in [6]	% Improvement
Encryption Rate (Mb/sec)	460.25	412.15	11.25
Critical path delay (nS)	4.372	5.168	15.4
Area of the cipher (mm^2)	3.68	4.22	12.79

V. CONCLUSION

A hardware implementation of the IDEA cipher using novelmodulo 2^n+1multipliers ispresented in this paper. It isshown that the proposed modulo $2n$+1 multiplier improvestheperformance of the variouscryptographic algorithms used insecure communication systems of networked instrumentationand distributed measurement systems. Efficient compressorsandsparsetreebasedinvertedendaroundcarryaddersare usedtoreducethedelayandcomplexity ofthemultiplier. Simulations areperformedontheknownimplementation and theproposedimplementation.Thepresentedimplementation isproven toperform betterthantheexistingoneinvarious aspects,(i.e., throughputandcriticalpathdelay).

REFERENCES

[1] R. Zimmermann, A. Curiger, H. Bonnenberg, H. Kaeslin, N. Felber andW. Fichtner, A 177 Mb/s VLSI implementation of the international dataencryption algorithm, IEEE J. Solid-State Circuits, 1994, 29, (3), pp.303-307
[2] R. Zimmerman, Efficient VLSI implementation of modulo (2^n _ 1)addition and multiplication, IEEE trans. Comput., 2002, 51, pp. 1389-1399.
[3] O. Cheung, K. Tsoi, P. Leong and M. Leong, Tradeoffs in parallel andserial implementations of the international data encryption algorithmIDEA, Lecture Notes in Computer Science, vol. 2162, pp. 333-340,2001.
[4] M. Leong, O. Cheung, K. Tsoi, and P. Leong, A bit-serial implementation of the international data encryption algorithm IDEA, 2000 IEEESymposium on Field-Programmable Custom Computing Machines, pp.122-131, 2000.
[5] C. Efstathiou, H. Vergos, S. Dimitrakopoulos and D. Nikolos, Efficientdiminished-1 modulo 2^n+ 1 multipliers, IEEE Trans. Comput., 2005,54, pp. 491-496.
[6] H. Vergos and C. Efstathiou, Design of efficient modulo 2^n+ 1multipliers, IETComput. Digit. Tech., 2007, 1, (1), pp. 49-57.
[7] R. Modugu, N. Park and M. Choi, A Fast Low-Power Modulo2^n+ 1 Multiplier Design, 2009 IEEE International Instrumentationand Measurement Technology Conference, pp.951-956, May 2009.
[8] R. Zimmermann and W. Fichtner., Low-power logic styles: CMOSversus pass-transistor logic, IEEE J. Solid- State Circuits, vol. 32, pp.1079-1090, July 1997.
[9] S. Veeramachaneni, L. Avinash, M. Rajashekhar and M. Srinivas,Efficient Modulo (2k _1) Binary to Residue Converters System-on-Chip for Real-Time Applications, The 6th International Workshop onDec. 2006 pp.195 - 200.
[10] C. Chang, J. Gu and M. Zhang, Ultra low-voltage lowpower CMOS4-2 and 5-2 compressors for fast arithmetic circuits, IEEE J. Circuitsand Systems I, Vol. 51, No. 10, pp. 1985- 1997, 2004
[11] M. Rouholamini, O. Kavehie, A. Mirbaha, S. Jasbi and K. Navi, ANew Design for 7:2 Compressors, Computer Systems and Applications,2007. AICCSA '07. IEEE/ACS International Conference on 13-16 May2007 Page(s):474 - 478.
[12] S. Mathew, M. Anders, R. Krishnamurthy and S. Borkar, A 4-GHz130-nm address generation unit with 32-bit sparse-tree adder core,IEEE Journal of Solid-State Circuits, Volume 38, Issue 5, May 2003Page(s):689 - 695.
[13] Zhongde Wang, Graham A. Jullien, William C. Miller., An efficient treearchitecture for modulo 2n +1 multiplication, VLSI Signal Processing14(3): 241-248 (1996).
[14] P. Kogge and H. S. Stone., A parallel algorithm for the efficient solutionof a general class of recurrence equations, IEEE Trans. Comput., vol.C-22, pp. 786-793, Aug 1973.
[15] A. Curigeret. al., VINCI: VLSI Implementation of the New Secret-keyBlock Cipher IDEA, Proc. of the Custom Integrated CircuitsConference, San Diego, USA, May 1993
[16] Yan Sun, DongyuZheng, Minxuan Zhang, and Shaoqing Li HighPerformance Low-Power Sparse-Tree Binary Adders, 8*th* InternationalConference on Solid state and Integrated Circuit Technology, ICSICT2006.
[17] Yi-Jung Chen, Dyi-Rong Duh, Yunghsian Sam Han: Improved Modulo(2^n+1) Multiplier for IDEA, J. Inf. Sci. Eng. 23(3):911-923(2007)