Implementation of an Energy Efficient Reconfigurable Authentication Unit for Software Radio

L.Thulasimani
Lecturer Department of Electronics and Communication Engineering
PSG College of Technology, Coimbatore
l_thulasi@yahoo.com; lthulasi@gmail.com

M.Madheswaran
Principal
Muthayammal Engineering College, Rasipuram
Madhesawran.dr@gmail.com

Abstract—**To promote the commercial implementation of software download for Software Defined Radio (SDR) terminals, a secure method of download is vital. Downloading of all the relevant software is performed via a public channel, and accordingly the security issue of the downloading is one of the key issues. For the purpose of security, it is necessary to ensure privacy, integrity, and authentication. The paper aims at an efficient reconfigurable hardware architecture which supports data authentication for the secured downloading of software in reconfigurable receivers. A reconfigurable authentication unit which operates in two different modes of RSA and D.H is proposed. The algorithms are to be implemented using Verilog, and their hardware utilization on the FPGA device is to be analyzed.**

***Index Terms*— DH, Hardware utilization, RSA, Reconfigurability, SDR.**

I INTRODUCTION

SDR terminals will dynamically reconfigure the baseband processing part of a wireless device. In order to provide the ability to change or update this portion of the device, a method of software download is necessary. The process of software download enables the introduction of new functionality (defined in software) into the terminal, with the aim of modifying its configuration and/or content. The major issue related to SDR development is that it requires a secure method of downloading.

Cryptography is the science of Information Security. It protects data from theft or alteration and can also be used for user authentication. Modern cryptography concerns itself with confidentiality, integrity, non-repudiation, and authentication. This paper aims at providing data authentication in reconfigurable receivers. RSA algorithm supports encryption, decryption and key exchange whereas D.H supports only key exchange. In both the algorithms, it is the main operation to compute modular exponentiation. However, the large bit modular operation makes the system low. Also it makes the hardware implementation difficult. For solving this problem, the Montgomery modular reduction

algorithm [1] is usually adopted for modular multiplication and for fast implementation.

RSA algorithm is based on the difficulty of the integer factorization problem and the security level has a close relationship with the key size of the algorithm. And the key size grows with the growth of higher security level. Till now, most of current RSA chip are fixed-precision solutions. That is, the operands cannot exceed a fixed bit-size. But in [2] and [3], scalable architectures for modular multiplication were proposed, which are based on multiple-word radix-2k Montgomery multiplication algorithm. [4] Examines various scalable implementations of elliptic curve scalar multiplication employing multiplicative inverse, focusing mainly on modular division architectures for D.H algorithm.

In this paper, section I gives the brief introduction about SDR and security issues and section II explains the RSA and D.H algorithms, Section III describes the proposed methodology, section IV elaborates the individual block and the simulation results are discussed in section V.

II RSA AND D.H ALGORITHM

A. The RSA Algorithm

The RSA algorithm was invented by Rivest, Shamir, and Adleman. Let p and q be two distinct large primes. The modulus n is the product of these two primes:

$$n=p \times q$$

Euler's totient function of n is given by

$$f(n) = (p-1)\times(q-1)$$

Now, select a number $1 < e < f(n)$ such that

$$\gcd(e,f(n)) = 1$$

and compute d with

$$d = e\text{-}1 \pmod{f(n)},$$

using the Extended Euclidian algorithm . Here e is the public exponent and d is the private exponent. The modulus n and the public exponent e are published. The values of d and the prime numbers p and q are kept secret. Encryption is performed by computing,

$$C = Me \pmod{n}$$

where M is the plaintext such that $0 \leq M < n$.The number C is the cipher text from which the plaintext M can be computed using,

$M = Cd \pmod{n}$

B. The D.H algorithm

Diffie–Hellman establishes a shared secret that can be used for secret communications by exchanging data over a public network. For this scheme, there are two publicly known numbers: a prime number q and an integer α that is a primitive root of q. If users A and B wish to exchange a key,
User A selects a random integer,

$X_A < q$ and computes,

$Y_A = \alpha^{X_A} \bmod q$.

Similarly user B selects a random integer,

$X_B < q$ and computes,

$Y_B = \alpha^{X_B} \bmod q$.

Each side keeps the X value private and makes the Y value publicly to the other side. User A computes the key as,

$K = (Y_B)^{X_A} \bmod q$.

And user B computes the key as,

$K = (Y_A)^{X_B} \bmod q$.

These two calculations produce identical results.

III PROPOSED METHODOLOGY

In many public-key encryption schemes (here RSA, D.H key exchange), Modular inverse and Modular exponentiation are the basic arithmetic operations heavily used. Modular inverse is computed from extended euclidean algorithm.Fig.1 shows the general hardware structure.

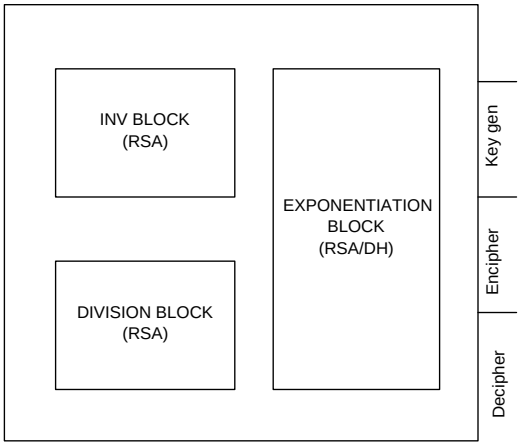

Figure.1.Overall hardware layout

IV IMPLEMENTATION OF INDIVIDUAL ARCHITECTURES

A. Divider Architecture

A Non-Restoring divider is s implemented as in Fig.2.In normal restoring division, Subtraction takes place until there is a sign changes and then again another addition is done in order to revert back the sign. This is not the case in non-restoring division. Here the only operation required is either addition or subtraction. Successively right-shifted versions of the divisor are subtracted from or added to the dividend, resulting in partial remainders. The sign of the partial remainder determines the quotient bit and, further, determines whether to add or subtract the shifted divisor.

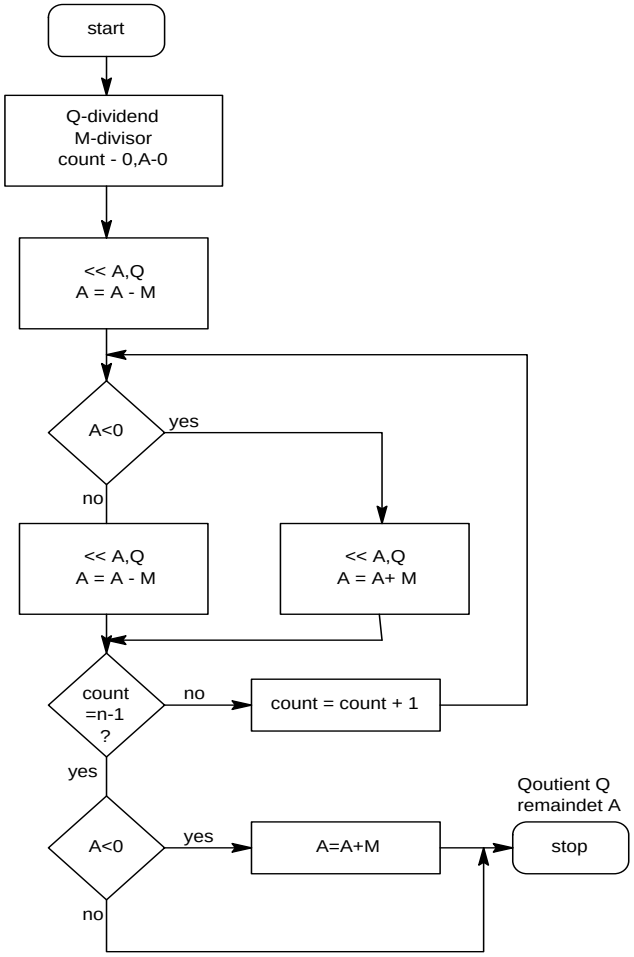

Figure.2 Flow Chart for .Non-Restoring division

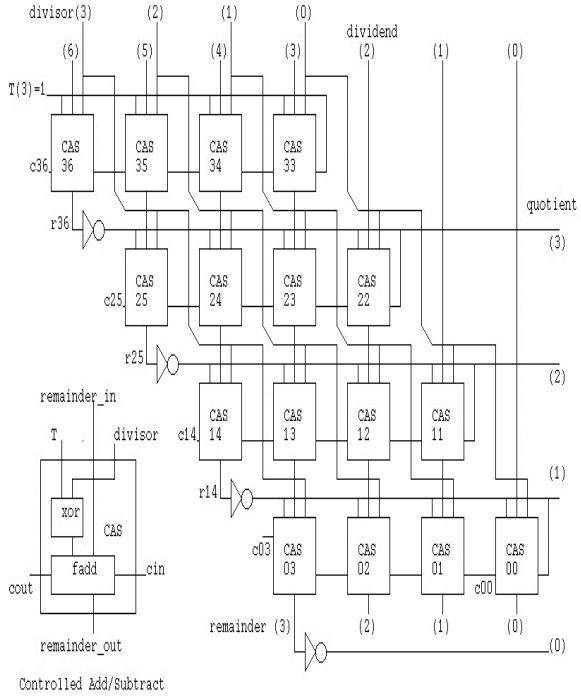

Figure.3Fast Divider Architecture

B.Modular inverse block

In order to compute the private key in RSA algorithm, there is need to find modular inverse. The flow of algorithm is,

function extended_gcd(a, b)

```
x := 0    lastx := 1
y := 1    lasty := 0
while b ≠ 0
    quotient := a div b
          temp := b
   b := a mod b
   a := temp
          temp := x
   x := lastx-quotient*x
   lastx := temp
          temp := y
   y := lasty-quotient*y
   lasty := temp
return {lastx, lasty, a}
```

If a=1, Then inverse exists and if 'lasty' is negative then, y = y + b and y is the inverse. Else inverse does not exist. The above algorithm computes a^{-1} mod b. Fig.3 shows the RTL schematic of the inverse block implemented.

C.Modular exponentiation

Modular exponentiation is performed by repeated modular multiplications. Modular Multiplication is C = A * B mod M where A, B < M. In security system where we have very large operand size, the hardware implementation is too expensive. Straightforward Method of calculation includes Multiplication then modulus division. Speeding the modular multiplication will have a great impact on the speed of public key algorithms [5].

Algorithm 1:

Method to perform A*B mod n.
Montgomery multiplication (A, B, n)
S[0];
for i in 0 to k - 1 loop
qi = (S[i]0 + Ai * B0) mod 2;
S[i+1] = (S[i] + Ai * B + qi * n)div 2;(critical delay equation)
end loop;
return S[k];

Hence the critical delay equation is replaced by carry save representation [6], where the sum of the bit vectors SUM and CARRY is equal to the sum of the four input bit vectors X1, X2, X3, and X4 as shown in Fig.4. The carry save representation of the four input operands is output from the register after only one clock cycle. Note that the input operands A and B and the output product S are now in a carry save representation (CSR) denoted by A1 and A2, B1 and B2, and S1 and S2 respectively.

Algorithm 1 can be rewritten as that written below replacing the critical delay equation with four to two CSA logic,

qi = (S1[i]0 + S2[i]0)+ (Ai * (B10 + B20)) mod 2;
S1[i+1],S2[i + 1] = CSR(S1[i] + S2[i] + Ai * (B1+ B2)+qi* n)div2; which has Critical delay of 3FA + 2XORs+1AND--8XORs+1AND

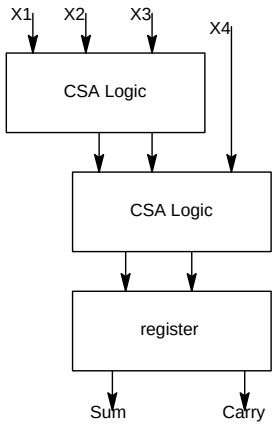

Figure 4.Block diagram of 4 to 2 CSA

These improved versions can be used for the calculation of modular exponentiation, as in algorithm 2 below

Algorithm 2:
Method to perform C^d mod n.
FOUR-to-two multiplier modular exponentiation (C, d, n)
$K = 2^{2k} \pmod n$; (computed externally)
P1[0], P2[0] = 4to2 MontMult (K, 0, C, 0, n);
R1[0]; R2[0] = 4to2 MontMult(K, 0, 1, 0, n);
for i in 0 to k_d -1 loop
P1[i+1], P2[i+1] = 4to2 MontMult(P1[i], P2[i], P1[i],P2[i],n];
if d[i] = 1 then
R1[i+1], R2[i+1] = 4to2 MontMult(R1[i], R2[i],P1[i], P2[i], n];
end if;
end loop;
M1,M2 = 4to2 MontMult(1, 0, R1[k], R2[k], n);
M = M1+ M2;
return M;

V RESULTS AND DISCUSSION

The hardware architecture is implemented in Verilog, and synthesis is performed with Xilinx ISE 9.2i.Virtex II kit is chosen for downloading the synthesized code. The hardware utilization of the divider and modular inverse algorithm are tabulated.

TABLE 1 HARDWARE UTILIZATION OF DIVIDER.

FPGA DEVICE : 2V4000BF957-6		
Allocated area	***Used/Available***	***Utilization***
I/Os	42/684	6%
Fun.Generators	234/46080	0%
CLB Slices	133 / 23040	0%
Dffs and Latches	17	-
frequency	9.87MHz	

TABLE 2 HARDWARE UTILIZATION OF INVERSE ALGORITHM.

FPGA DEVICE : 2V4000BF957-6		
Allocated area	***Used/Available***	***Utilization***
I/Os	33/684	4%
Fun.Generators	789/46080	1%
CLB Slices	406 / 23040	1%
Dffs and Latches	133/46080	0%
Frequency	31.017MHz	

Table 1 and 2 shows the hardware utilization summary of divider and modular inverse algorithm. Table 3 shows the comparison of current work with that of previous one, and it could be seen that area utilization is reduced.

TABLE 3COMPARISON WITH PREVIOUS WORK

Architectu	**Work**	**FPGA**	**CLB Slices Used/available**
Divider	[9]	Virtex II pro	455/13,696
	Current work	2v4000bf957-6	133 / 23040
Inverse Algorithm	[9]	Virtex II pro	688/13,696
	Current work	2v4000bf957-6	406 / 23040

TABLE 4 DEVICE UTILISATION OF MODULAR MULTIPLICATION WITH AND WITHOUT CSA

FPGA DEVICE : 2V4000BF957-6				
Allocated area	***Used/Available***		***Utilization***	
	Without CSA	**With CSA**	**Without CSA**	**With CSA**
I/Os	36/684	49/684	5%	7%
Fun.Generators	78/46080	310 / 23040	0%	0%
CLB Slices	53/ 23040	292/ 23040	0%	1%
Flip Flops	42/46080	456/ 23040	0%	0%
Delay	13.38ns(without CSA)		11.20ns(with CSA)	

Similar implementation of modular exponentiation algorithm is done and the individual blocks are integrated to get the overall hardware unit. Since single hardware is implemented for both algorithms, improved power and area results are expected with reduced delay[8].

TABLE 5:DEVICE UTILIZATION SUMMARY OF MODULAR EXPONENTIATION

FPGA DEVICE : 2V4000BF957-6		
Allocated area	***Used/Available***	***Utilization***
I/Os	33/684	4%
Fun.Generators	2493/46080	5%
CLB Slices	2361/23040	10%
IO flip flops	3633/46080	7%
Frequency	110.43MHz	

TABLE 6:DEVICE UTILIZATION SUMMARY OF RSA AND DH ALGORITHMS

FPGA DEVICE : 2V4000BF957-6				
Allocated area	***Used/Available***		***Utilization***	
	RSA With CSA	DH With CSA	RSA With CSA	DH With CSA
I/Os	40/684	33/684	5%	4%
Fun.Generators	14265/46080	6774/46080	30%	14%
CLB Slices	16611 / 23040	5680 / 23040	50%	24%
Flip Flops	14512/46080	7262/46080	31%	15%
Frequency	32.475MHz(RSA)		80.048MHz	

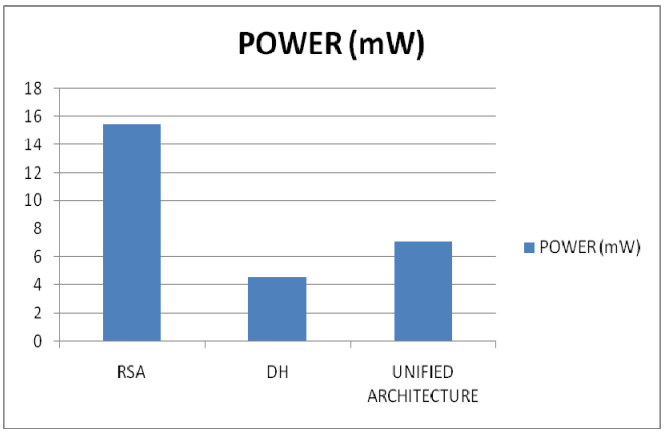

Figure 5Power Consumption Comparison

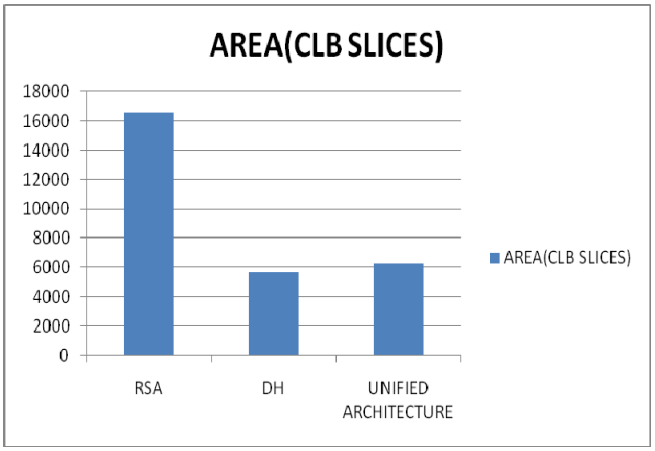

Figure 6 Area Utilization Comparison

Fig 5 and 6 shows the area and power consumption of the unified architecture of RSA and DH algorithms.

VII CONCLUSION

Thus in this work, a VLSI architecture of the authentication unit for the reconfigurable receiver is presented. The architecture is reconfigurable in the sense that, it operates either in RSA mode to perform encryption, decryption and key exchange or in D.H mode to perform key exchange only. It guarantees security level in reconfigurable receivers requiring data authentication. The synthesis results are compared with that of individual implementation of the algorithms, thus applicable for the reconfigurable receiver terminals.

VIII REFERENCES

[1] P. L. Montgomery, "Modular multiplication without trial division," Math. Computation, vol. 44, pp.519-521,1985.
[2] A. F. Tenca and . K. Ko,c, " A scalable archi-tecture for multiplication", Cryp-tographic Hardware and Systems 1999, CHES'99, pp.94-108, 1999.
[3] A. F. Tenca, G. Todorov, and . K. Ko,9, "High-radix design of a scalable modular mul-tiplier", Cryptographic Hardware and Embedded Systems2001,CHES'2001, pp.185-201,2001.
[4]C. McIvor, M. McLoone and J.V. McCanny,"Modified Montgomery modular multiplication and RSA exponentiation techniques", IEE Proc.-Comput. Digit.Tech., Vol. 151, No. 6, November 2004.
[5] Himanshu Thapliyal,anvesh,vivek " modified montgomery moduar mltiplication using 4:2 compressor andCSA adder",centre for VLSI and embedded system technologies,2005.
[6]Richa Garg ,Renu vig "An efficient montgomery modular multilication algorithm and RSA cryptographic processor" international conferenc on computational intelligence and multimedia applicaions,2007.
[7 N.sklavos,P p. kitsos k. Papadopoulos o. koufopavlou "Design, Architecture and Performance Evaluation of the Wireless Transport Layer Security" *The Journal of Supercomputing, 36, 33–50, 2006 C_ 2006 Springer Science + Business Media*, Inc. Manufactured in The Netherlands.
[8]. Zerene Sangma,"Hardware Implementation of Elliptic Curve Diffie-Hellman Key Agreement Scheme in GF(p)" Rochester Institute of Technology Rochester, New York,October 2008
[9]L. Hars, "Modular inverse algorithms without multiplications for cryptographic applications," EURASIP Journal on Embedded System, vol. 2006, January 2006.
[10]R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120–126, February 1978.
[11]T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469–472, July 1985.
[12]M. D. Shieh, J. H. Chen, H. H. Wu, and W. C. Lin, "A new modular exponentiation architecture for efficient design of rsa cryptosystem," IEEE Transactions on Very Large Scale Integration (VLSI) Systems archive, vol. 16, no. 9, pp. 1151–1161, September 2008.
[13]M. A. Hasan, "Efficient computation of multiplicative inverses for cryptographic applications," in 15th IEEE Symposium on Computer Arithmetic. Vail,Colorado, U.S.A.: IEEE, 2001.
[14]N. I. for Standards and Technology, "Digital signature standard (dss)," August 1991.
[15] William Stallings, "Cryptography and Network Security: Principles and Practices." 3rd edition, 2003.
[16] http://www.arl.wustl.edu/~jl1/education/cs502/course_project.htm
[17] Bruce Schneier, "Applied Cryptocraphy: Protocols, Algorithms, and Source Code in C." 2nd edition, 1996.

AUTHORS PROFILE

L. Thulasimani has obtained her BE and ME degree from Coimbatore Institute of Technology, India in 1998 and 2001 respectively. She has started her teaching profession in the year 2001 in PSNA Engineering College, Dindigul. At present she is an Lecturer in department of Electronic and Communication Engineering in PSG college of Technology, Coimbatore .She has published 4 research papers in International and National conferences. She is a part time Ph.D research scalar in Anna University Chennai. Her areas of interest are Wireless security, Networking and signal processing. She is a life member of ISTE and IEEE.

Dr. M. Madheswaran has obtained his Ph.D. degree in Electronics Engineering from Institute of Technology, Banaras Hindu University, Varanasi in 1999 and M.E degree in Microwave Engineering from Birla Institute of Technology, Ranchi, India. He has started his teaching profession in the year 1991 to serve his parent Institution Mohd. Sathak Engineering College, Kilakarai where he obtained his Bachelor Degree in ECE. He has served KSR college of Technology from 1999 to 2001 and PSNA College of Engineering and Technology, Dindigul from 2001 to 2006. He has been awarded Young Scientist Fellowship by the Tamil Nadu State Council for Science and Technology and Senior Research Fellowship by Council for Scientific and Industrial Research, New Delhi in the year 1994 and 1996 respectively. He has published 120 research papers in International and National Journals as well as conferences. His field of interest includes semiconductor devices, microwave electronics, optoelectronics and signal processing. He is a Senior member of IEEE, Fellow of IETE, and IE and member of ISTE