Sürdürülebilir Mühendislik Uygulamaları ve Teknolojik Gelişmeler Dergisi
2020, 3(1): 3-11

Montgomery Modüler Çarpma Algoritması için Yeni Donanım Mimarileri

New Hardware Architectures for Montgomery Modular Multiplication Algorithm

[1]Zuhal KOCA

*[1] Tokat Gaziosmanpaşa Üniversitesi, zuhal152@gmail.com

Geliş Tarihi : 17.05.2020
Kabul Tarihi : 22.05.2020

ÖZET

İnternet uygulamalarının hız kazanması ve pek çok uygulamanın internet üzerine taşınması güvenli haberleşme ve bilgi güvenliği konusunu da beraberinde getirmiştir. Verilerin Gelişen bilgi ve iletişim teknolojisi, pek çok uygulamanın internet üzerine taşınması dolayısıyla dünyanın her yerinden erişilebilir hale gelmesine olanak vermiştir. Büyük kolaylık ve avantaj sağlayan bu durum, güvenli haberleşme ve bilgi güvenliği konusunu çeşitli sorunları beraberinde getirmiştir. Bu duruma karşı verilerin şifrelenmesi çözümleri üretilmiştir.
Verilerin şifrelenmesi ile güvenli haberleşmenin dört temel özelliğinin sağlanması amaçlanmıştır. Bunlar Gizlilik, Veri Bütünlüğü, Kimlik Doğrulama ve Reddedilmezlik' tir. Şifrelenmiş verinin güvenliği şifreleme algoritmasının gücüne ve kullanılan anahtarların gizliliğine bağlıdır. Şifreleme algoritması, şifreleme ve şifre çözme işlemini gerçekleştirmek üzere kullanılan matematiksel fonksiyondur. Bir algoritma, hem yazılımla hem de donanım bileşenleri ile gerçekleştirilebilir. Birçok algoritma, şifreleme ve şifre çözme işlemini gerçekleştirmek amacıyla, düz metin dışında "anahtar" olarak bilinen bir değer de kullanır.
Bu çalışmada, ilk olarak güvenlik amaçlı kullanılan şifreleme sistemlerinin temel işlemlerinden birisi olan Montgomery Modüler Çarpma algoritması anlatılmış, daha önce yapılan çalışmalarda kullanılan CSA toplayıcısı yerine alternatif olarak CIA,CLA ve CSIA toplayıcıları kullanılarak eleman sayısı, işlem hızı, gecikme süresi, toplama işlem sayısı gibi ortak performans kriterlerinin değişimi simülasyon sonuçları ile test edilmiştir. Sonuç olarak, üç giriş ve iki çıkışa sahip CSAtoplayıcısı ile yapılan işlemlerde kullanılan sum ve carry, iki giriş ve tek çıkışa sahip CIA, CLA ve CSIA toplayıcılarına uygulanamadığı için özellikle toplama işleminde çalışma hızı daha uzun sürmüştür. Ek olarak kullandığımız toplayıcılarda eleman sayısında artış görülmüştür.
Anahtar Kelimeler: CIA ve CLA toplayıcıları, Montogomery Modüler Çarpması, Açık Anahtar Şifreleme Sistemleri.

ABSTRACT

The acceleration of Internet applications and the transportation of many applications on the Internet has brought with it the issue of secure communication and information security.
Providing great convenience and advantage, this situation has brought with it a variety of problems on the issue of secure communication and information security. Data encryption solutions have been produced against this situation.
It is aimed to provide four basic features of secure communication by encrypting data. These are Privacy, Data Integrity, Authentication, and Disapproval. The security of encrypted data depends on the power of the cryptography algorithm and the privacy of the keys used. The encryption algorithm is the mathematical function used to perform encryption and decryption.
An algorithm can be performed with both software and hardware components. Many algorithms also use a value known as "key" other than plain text to perform encryption and decryption.
In this study, cia and cla collectors were used together as an alternative instead of the CSA adder used in the Montgomery Modular Multiplication Algorithm and aimed to accelerate the process. As a result, the speed of operation took longer, especially in the addition process, as sum and carry used in csa operations could not be applied to other adders. In addition, the number of complement has increased.

Keywords: CLA & CIA Adders, Montgomery Modular Multiplication, Public Key Cryptosystems

Sorumlu Yazar: zuhal152@gmail.com

1. GİRİŞ

Bilgisayar ağlarının ve haberleşme sistemlerinin güvenliğinin sağlanması için kullanılan en önemli işlem, verilerin şifrelenerek anlamsız hale getirilip hedefe gönderilmesi ve hedefte tersi işlem yapılarak tekrar eski hele getirilmesidir.

Kriptoloji şifre bilimidir ve amacı haberleşmek isteyen noktaların bulunduğu güvensiz ortam içerisinde, güvenli bir iletişim kanalı oluşturarak güvenliğini sağlamaktır. Böylece ortamda bulunan üçüncü şahıslar, haberleşen iki nokta arasında gidip gelen verileri elde etseler bile adlandıramazlar. Birinci nokta gönderici noktasıdır ve gönderici verisi açık veridir. Açık verinin şifreleme bloğundan geçtikten sonraki hali şifreli veri adını alır. Şifreleme bloğunda daha önceden paylaşılmış parametrelerle (anahtar ya da anahtar elde etmek için kullanılacak fonksiyonlar, değerler,...) açık veri şifrelenir ve güvensiz haberleşme ortamı üzerinden alıcıya ulaştırılır.

Şifrelenmiş verinin güvensiz ortamda üçüncü kişiler tarafından ele geçirilmesi hiçbir şey ifade etmez çünkü artık şifreli veri sadece alıcının bildiği parametrelerle çözüldükten sonra anlamlı hale gelecektir. Açık anahtar şifreleme yöntemleri kullanan çeşitli algoritmalar bulunur. Günümüzde en güvenilir algoritmalardan birisi Eliptik Eğri Şifreleme Algoritmasıdır.

Eliptik Eğri Şifrelemesi aynı güvenlik seviyesi için daha kısa anahtar uzunluğu kullanır.

Şifreleme uygulamaları hızlı aritmetik işlemler gerektirir. Bu nedenle çarpma işlemi bilgisayar aritmetiğinin en önemli işlemlerinden birisi olmuştur. Ortak anahtar şifreleme sistemlerinde sonlu bir alanda veya sonlu bir halkada işlem yaparken temel aritmetik işlem modüler çarpmaya dayanmaktadır. Modüler çarpma uzun bir işlem gerektirir. Modüler çarpımlar açık anahtar şifreleme sistemlerinde büyük modüller gerektirdiği ve çok kullanıldıklarından dolayı büyük bir öneme sahiptir ve özellikle bilgi güvenliği açısından doğrudan bir etkiye sahiptir. Bu nedenle standart algoritmalar dışında algoritmalarda geliştirilmiştir.

Geliştirilmiş bu algoritmalardan birisi olan Montgomery Modüler Çarpma algoritması ile modüler işlemlerin hem donanım hem de yazılım tabanlı sistemlerde gerçekleme zorluğu aşılmıştır. Bu algoritmada çarpma işlemi farklı bir şekilde yapıldığından algoritma sonucunda modüler işleme gerek kalmaz. Ayrıca, bu algoritma işlemleri seri olarak arka arkaya gerçekleştirdiği için donanımsal tasarımlarda büyük avantajlar sağlar.

Şifreleme sistemlerinde en önemli işlemlerden birisi aynı zamanda hem şifreleme hem de şifre çözme işlemi için gerekli ve modüler çarpma dizisi ile yapılabilen modüler üs almadır. Bu nedenle, hızlı modüler çarpma, tasarımda gerçek zamanlı şifreleme ve şifre çözmenin anahtarı haline gelir ayrıca donanımda veri güvenliği gerektiren sistemleri hızlandırarak enerji tüketimini azaltmak gibi işlemleri basitleştirir. Bu algoritma aynı zamanda Eliptik eğri Şifrelemesinin son aşaması olan Nokta Çarpma işleminin temelini oluşturur.(Yavuz, 2008) Bu çalışmada Eliptik eğri şifreleme işlemlerinde en önemli alt blok olan modüler çarpma işlemi için Montgomery Modüler çarpma algoritması kullanılarak sağladığı avantajlardan yararlanılacaktır.

2. YÖNTEM

I. MONTGOMERY MODÜLER ÇARPMA

Günümüzde en başarılı modüler çarpma yöntemlerinden birisi, 1985 yılında P. L. Montgomery tarafından önerilen ve tamsayıların etkili bir modüler çarpma işlemini gerçekleştiren MMÇ algoritmasıdır. Bu algoritma şifreleme algoritmalarının hızlandırılmasında önemli bir etki sağlar. Alan ve zaman değişimleri, yüksek sayı tabanı kullanımı sağlar.(Kuang, Wu & Lu, 2016)

Montgomery modüler çarpım, bölünmeyi çarpmalarla değiştirerek, çok sayıda hassas modüler çarpmayı hızlı bir şekilde gerçekleştirmek için kullanılan bir algoritmadır.(Josel& Abida, 2016) Özellikle A, B, N sayılarının değerleri büyük olduğunda AB mod N hesaplamasını en hızlı yapan algoritmadır.

$$C = AB \bmod N \ ,$$

çarpımını gerçekleştirmeden önce, her iki faktör de Montgomery uzayına dönüştürülmelidir.

Çarpma işleminden kalanı bulurken Montgomery uzayına geçmek için A ve B sayısı R ile çarpılır,

$$(AR)\ (BR)\ (\mathrm{mod}\ N)$$

AB mod N 'in hesaplanması istendiğinde;

Öncelikle N' den büyük pozitif bir R tamsayısı seçiyoruz. R(Radix) herhangi bir tabanda k basamaklı bir sayıdır. Bu sayıyla bilgisayarda çarpma, bölme ve modül kaydırma veya mantıksal işlemler yapılabilir.
R'nin değeri $R= 2^k$ olarak tanımlanır.

k pozitif bir tamsayıdır. N genellikle $2k-1 < N < 2k$ aralığında yer alır.

N ve R, N tek sayı olduğunda kolayca elde edilen en büyük ortak bölen (gcd- greatest common division) $gcd(R,N)= 1$ ve $0< R^{-1} < N$ ve $0< N'< R$ değerleri arasında tanımlanan R^{-1} ve N' diye iki sayı vardır ve $R^* R^{-1} - N^*N' =1 \ (\mathrm{mod}\ R)$ ve $N^*N' \ (\mathrm{mod}\ R)= 1$ olmalıdır. (N'=-1/ N)

R ve R-1 Genişletilmiş Öklit (Euclidean) algoritması kullanılarak hesaplanır. (Yavuz, 2008)

Bu algoritma belirli bir tabana göre (modulus) verilen sayının tersini bulur. Yani bir sayının bir mod da hangi sayı ile çarpıldığında 1 değerini vermesidir. (Miaoqing, Gaj, ElGhazawi, & Fellow, 2011) (Montgomery, 1985) (Farmani)

N ve R aralarında asal olmak zorundadır. A tamsayısı A < N şartını sağlamalıdır.

A ve B sayılarının modül değeri N sisteminde R' ye bağlı kalanı hesaplamak için:

$$\overline{A} = A.R \bmod N$$

$\overline{B}$ = BR modN

$\overline{A}$ ve $\overline{B}$ değerleri N sistemindeki, Montgomery kalan değerleridir.

$$\overline{C} = \overline{A}.\overline{B}.R^{-1} \bmod N$$

$\overline{A}$ ve $\overline{B}$ değerleri yerine konulduğunda

$\overline{C}$ =ARBRR^{-1} modN

$R.R^{-1} =1$ olduğu için; $\overline{C}$ =ABRmodN RR^{-1} = C olur.

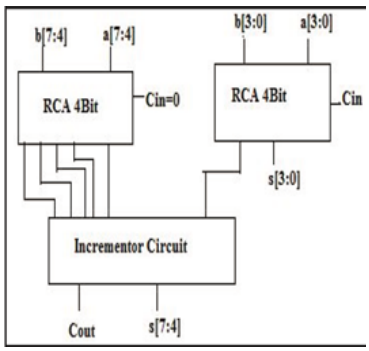
Algoritma 1 Algoritma MM: Radix-2 Montgomery Çarpma
Inputs : A, B, N (modulus)
Output : S[k]
1. S[0] = 0;
2. for i = 0 to k − 1 {
3. qi = ($S[i]_0$ + Ai × B_0) mod 2;
4. S[i+1] = (S[i] + Ai× B + qi × N)/ 2;
5. }
6. if (S[k] ≥ N) S[k] = S[k] − N;
7. return S[k];

Şekil.1 MM algoritması

II. CSA (CARRY SAVE ADDER) YERİNE KULLANILACAK TOPLAYICILAR

A. CARRY INCREMENT ADDER (CIA)

Ripple Carry Adder' lar ve artımsal devrelerden oluşur. Artımsal devreler sıralı bir dizi gibi Ripple Carry içerisindeki HA(yarım toplayıcı)' lar kullanılarak tasarlanabilir. Toplama işlemi bitlerin toplam sayısını 4 bitlik gruplara böler ve toplama işlemini 4 bitlik RCA' lar kullanarak yapar. Her grup için iki kısmi toplamı hesaplamak ve doğru olanı seçmek yerine, sadece bir kısmi toplam, girdi taşımalarına göre hesaplanır ve gerekirse arttırılır. (Kulkami, 2015).



Şekil.2 CIA blok şeması

B.CARRY LOOK AHEAD ADDER (CLA)

Bu toplayıcı, giriş sinyallerini temel alarak, taşıma sinyallerini önceden hesaplayarak taşıma (carry) yayılma süresi problemini ortadan kaldırır. (Şekil.3). Yani bir taşıma sinyalinin yayılması için gerekli kapı sayısını azaltarak taşıma gecikmesini azaltır. Bir tam toplayıcı için yayılma ve üretilme şu şekilde olur: (Singh, Kumar & Singh , 2009) (Jusbir & Lalit, 2015)

$P_i = A_i \oplus B_i$ Carry propagate
$G_i = A_i B_i$ Carry generate

Yayılma ve üretilme sinyalleri sadece giriş bitlerine bağlı olduğundan dolayı bir kapı gecikmesinden sonra geçerli olacaktır. Daha sonra aşağıda verildiği şekilde Internal Carry Generation stage hesaplanır:

$$S_i = P_i \oplus C_{i-1}$$

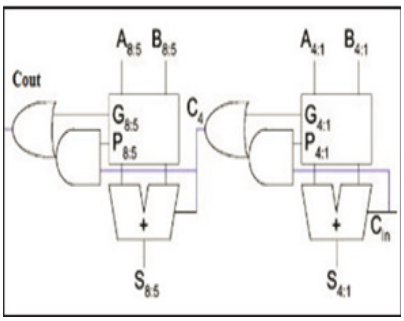
C_i : carry-in

$C_{i+1} = G_i + P_i C_i$ ve

$C(i) = G(i) + P(i).C(i-1)$ eşitliği elde edilir.

Son adımda toplama (Sum Generation Stage Sum) hesaplanır:

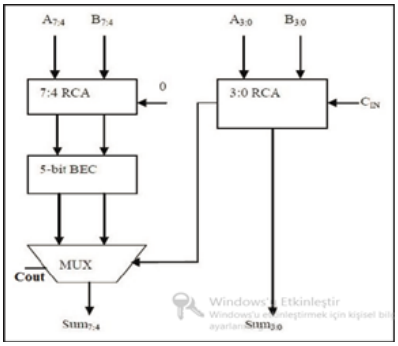
$$Sum[i] = A_i \oplus B_i \oplus C[i-1]$$



Şekil.3 CLA blok şeması

C.CARRY SELECT ADDER (CSIA)

Birçok hesaplama sisteminde, çoklu eldelerin (carry) bağımsız olarak üretilmesiyle elde gecikme sorununu azaltmak ve daha sonra toplamı üretmek için kullanılan toplayıcıdır. Bir CSIA, iki n bitlik sayının (n+1) toplamını hesaplayan bir toplayıcıdır. Toplam ve elde (sum ve carry) üretimi birbirinden bağımsızdır. Yani C_{in} =1 ve C_{in} =0 paralel olarak yürütülür. C_{in} ' e bağlı olarak harici mux'lar bir sonraki aşamaya yayılacak olan eldeyi seçer. Ayrıca eldeye bağlı olarak toplam seçilir. Böylece gecikme azaltılmış olur. Yapı Mux' ların karmaşıklığına göre artar. CSIA, diğer toplayıcılara göre basit ancak daha hızlıdır. Hızlı olmasına rağmen devre elamanı sayısı diğer toplayıcılara göre iki kat fazladır. Ek olarak diğer toplayıcılara göre enerji tüketimi ve kullandığı alanda fazladır. (Şekil.4) (Kulkami, 2015) (Jusbir & Lalit, 2015)

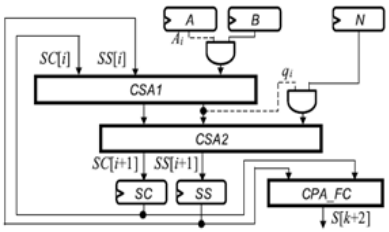


Şekil.4 CSIA blok şeması

III. VAROLAN VE ÖNERİLEN MONTGOMERY ÇARPMASI

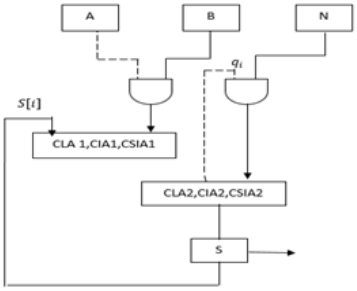
A. SCS-MM-1 ÇARPICI

Şekil.1 Montgomery Modüler Çarpım algoritmasının Radix-2 versiyonunu göstermektedir. Önerilen tasarımda öncelikle bu algoritmada toplama işlemi sırasıyla CLA, CIA ve CSIA toplayıcıları kullanılarak gerçeklenmiştir. CSA yerine kullanılacak toplayıcılar iki giriş tek çıkış olduğundan dolayı Şekil.5 te verilen temel aldığımız tasarımda SS[i] ve SC[i] çıkışları tek bir çıkış olarak alınmıştır. Toplama işleminin uzun sürmesinden dolayı S[i] çıkışı Sum ve Carry olarak ayrı ayrı tanımlanmış böylece toplama hızı artırılmıştır. A ve B değerleri sırasıyla carry save gösterimiyle tanımlanmıştır. Şekil.5 te CSA1 ve CSA2 yerine bu çalışma için sırasıyla CLA, CIA ve CSIA toplayıcıları konularak yapılan yeni tasarımda toplama hızının yavaşladığı görülmüştür.



Şekil.5 SCS-MM-1 çarpıcı

Algoritma 1 de ve Şekil.5 te verilen, SCS- tabanlı Montgomery Çarpmasında CSA adder' ının yerine Şekil.6 verilen tasarımda CLA, CIA ve CSIA toplayıcıları konularak kullanılan eleman sayısı ve hız ölçümü yapılmıştır. Tasarımda CSA için tanımlanan SS ve SC tek bir çıkış olarak alınmıştır. Şekil.1 de A,B ve N diye üç giriş bulunmaktadır. A ve B çarpma işleminde kullanılan sabit değerler, N modül değeri S[k] ise çıkış değeri için tanımlanmış ifadedir. Burada normalden farklı gösterim S[i], S' in 0. ncı anındaki değeridir. Programlamaya başlamadan önce S' in 0. ncı değeri 0 alınır. For döngüsü içerisinde MMÇ ye ait işlemler gerçekleştirilir. For döngüsü bir sayaç ile ifade edilir. Döngüden sonraki karşılaştırma S ve N arasında eğer şart sağlanmıyorsa çıkışa S[k]' yı verilir. (Tablo 1) (Kuang , Wu & Lu, 2016)



Şekil.6 SCS-MM-1 Çarpıcı

Tablo 1. SCS- MM-1 Simülasyon sonuçları

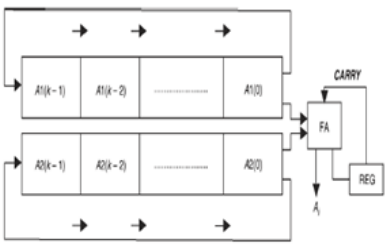
Toplayıcı	Çarpıcı	Anahtar uzunluğu	Çalışma Hızı (MHz)	LUT sayısı
CLA	SCS	1024	2.54	7752
CIA		1024	3.16	6215
CSIA		1024	2.24	8193

B. FCS- TABANLI MM

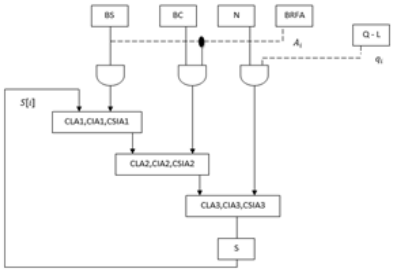
Bu tasarımda, temel aldığımız çalışmada toplayıcı olarak yine CSA toplayıcıları kullanılmıştır. A değeri sum ve carry olarak BRFA yazmacı içinde depolanır. Üç CSA kullanılarak yapılan tasarımda CSA toplayıcısının yerine CLA, CIA ve CSIA toplayıcıları konularak sonuç alınmıştır(Şekil.9). BRFA yazmacının kullanılmasındaki amaç içerisindeki depolama elemanlarının enerji tüketimini önemli ölçüde azalttığı içindir. BRFA yazmacı (Şekil.8), iki shift register , bir tam toplayıcı(FA) ve bir flip-floptan oluşur. A değerleri başlangıçta iki yazmaçta saklanır ve daha sonra her tekrarda bir bit konumu ile sağa kaydırılır. Tam toplayıcı, önceki tekrarda üretilen carry ve A1 ve A2 değerlerinin en az anlamlı bitini ekleyerek Ai değerini ardışık olarak üretir (Şekil.7). Ayrıca algoritmada 3. satırda bulunan Q_i, Q_L devresi ile hesaplanır. Yani Q_i değerleri ve toplayıcılar aynı saat çevriminde hesaplanır. (Tablo2)

Algoritma FCS-MM-1:
FCS-Tabanlı Montgomery Çarpıcı
Inputs : AS,AC, B, N (modulus)
Output : S[k+2]
1. S[0] = 0;
2. for i = 0 to k +1 {
3. qi = ($S[i]_0$ + Ai × B_0) mod 2;
4. S[i+1] = (S[i] + Ai× B + qi × N) / 2;
5. }
6. return S[k+2];

Şekil.7 FCS-MM-1 Montgomery Çarpım Algoritması



Şekil.8 BRFA Yazmacı



Şekil.9 FCS-MM-1Montgomery çarpıcı

Tablo 2. FCS- MM Simülasyon sonuçları

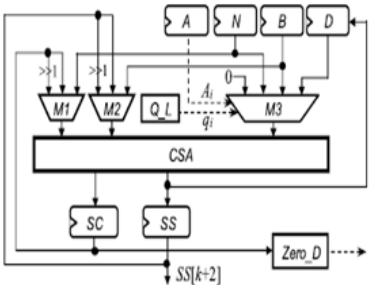
Toplayıcı	Çarpıcı	Anahtar uzunluğu	Çalışma Hızı (MHz)	LUT sayısı
CLA	FCS	1024	1.16	15180
CIA		1024	2.09	11903
CSIA		1024	2.28	16199

C. KRİTİK YOL GECİKMESİ AZALTILMASI

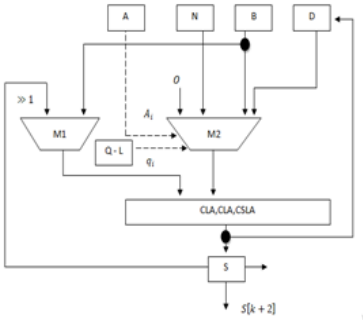
Varolan algoritmada B ve N değerlerini toplayarak, bir seviyeli CSA kullanılarak ve çoğullayıcılar eklenerek giderilmeyi amaçlanan 3. tasarımda , kritik yol gecikmesini azaltma amaçlanmıştır. Tek-seviyeli CSA yerine önerilen tasarımda üç farklı toplayıcı (CLA, CIA ve CSIA) kullanılarak yeni değerler alınmıştır. D= B+ N olarak tanımlanmıştır ve bu değer önceden hesaplanmıştır. D elde yayılma işlemi için tanımlanmıştır. Fakat bu işlem ek saat çevrimi gerektirir (Şekil.10 (b)). Önerilen tasarımda CSA toplayıcısı yerine kullanılacak üç toplayıcılar iki giriş olduğu için M1 ve M2 birleştirilmiştir. Zero- D devresi SC' nin sıfıra eşit olup olmadığını tespit etmek için kullanıldığından önerilen tasarımda carry ve sum değerleri olmayacağından dolayı kaldırılmıştır (Şekil.11). Önerilen tasarımda kullanılan farklı toplayıcılarla yapılan ölçümlerde özellikle eleman sayısında artış görülmüştür.

Değiştirilmiş SCS-MM Algoritması:
Inputs : A, B, N (modulus)
Output : S[k+2]
1. S = (B+N+0);
2. S=0;
3. for i = 0 to k +1 {
4. qi = ($S[i]_0$ + Ai × B_0) mod 2;
5. if (Ai=0 and qi =0) x = 0;
6. if (Ai=0 and qi =1) x = N;
7. if (Ai=1 and qi =0) x = B;
8. if (Ai=1 and qi =1) x = D;
9. S[i+1] = (S[i] + Ai×B + qi × N)/ 2;
10. }
11. return S[k+2];

Şekil.10(a) Değiştirilmiş SCS-MM Algoritması



Şekil.10 (b) MSCS- MM Çarpıcı



Şekil.11 MSCS- MM Çarpıcı

Tablo 3. MSCS Simülasyon sonuçları

Toplayıcı	Multiplier	Anahtar uzunluğu	Max Çalışma Hızı (MHz)	LUT sayısı
CLA	MSCS	1024	4.04	14885
CIA		1024	5.24	13610
CSIA		1024	4.12	14157

3. TARTIŞMA, SONUÇ VE ÖNERİLER

Temel alınan makalede, yapılan çalışma entegre üzerinde gerçeklenmiştir. Toplayıcı olarak üç giriş iki çıkışa sahip ve toplama hızı çok yüksek olan (carry(C) ve sum(S)) CSA (Carry Save Adder) kullanılmıştır. Bu çalışmada, CSA, yerine alternatif olarak CLA(Carry Look Ahead Adder), CIA (Carry Increment Adder) ve CSIA(-Carry Select Adder) toplayıcıları kullanılarak tasarımlar yeniden düzenlenip CLA, CIA ve CSIA için uygun hale getirilerek Xlinx Vivado 17.2 yazılım programında verilog kodları yazılıp simülasyon sonuçları alınmıştır. Bizim çalışmamızda uygulama entegre üzerinde yapılmadığından ve ayrıca bu toplayıcılar daha önce birlikte kullanılmadığından dolayı tam karşılaştırma yapılamamıştır. Farklı algoritmalara göre alınan sonuçlarda carry-save gösteriminde kullanılan ve sadece CSA ile yapılan işlemlerde kullanılan sum ve carry bu çalışmada kullanılan toplayıcılara uygulanamadığı için özellikle toplama işle minde çalışma hızı daha uzun sürmüştür. Ek olarak eleman sayısında artış görülmüştür. Algoritma ve tasarımlara göre alınan sonuçlar Tablo 1, Tablo 2 ve Tablo 3' te verilmiştir. CSA kullanılan orijinal tasarımda çalışma hızı daha iyidir. Buradaki amaç Montgomery çarpmasının farklı toplayıcılarda verdiği sonuçlara göre özellikle şifrelemenin en temel bölümünü oluşturan bu çarpmanın uygulamada bize ne sağlayacağının gözlemlenmesidir. Çalışmanın devamında Eliptik Eğri Diffie-Hellman şifrelemesi için gerekli hiyerarşinin en alt bloğu olan Montgomery çarpmasında daha fazla tercih edilen CSA yerine daha önce kullanılmayan CLA ve CIA toplayıcıları birlikte kullanılarak işlemci tasarımı için farklı bir yol izlenecek ve bu algoritmanın sağladığı avantajlardan yararlanılacaktır. Karşılaştırma için pipeline ve folding tasarımlar yapılması hedeflenmiştir. Böylece bu toplayıcıların bize sağlayacağı avantajlar ve dezavantajları daha iyi gözlemleme imkanımızın olacaktır.

4. KAYNAKLAR

Yavuz, İ. (2008).Eliptik Eğri Kriptosisteminin FPGA üzerinde Gerçeklenmesi. (Yayınlanmış Yüksek Lisans Tezi)

Shiann-Rong Kuang, Member, IEEE, Kun-Yi Wu,& Ren-Yao Lu (2016). Low-Cost High-Performance VLSI Architecture for Montgomery Modular Multiplication.

Jose1.D , Nathimugil.J , Abida. B (2016) . Implementation of Optimized Montgomery modular Multiplier on FPGA .

Miaoqing Huang, Member, IEEE, Kris Gaj, Tarek El-Ghazawi & Fellow. IEEE (2011) New Hardware Architectures for Montgomery Modular Multiplication Algorithm.

P.L. Montgomery, Math. Comput., vol. 44, no.170, pp. 519- 521, Apr. 1985. “Modular Multiplication without trial division.

Farmani. M. Montgomery Modular Arithmetic.

Prof. Rashmi Rahul Kulkarni . (2015). Comparison among Different Adders.

R- P. Pal Singh & P. Kumar & B. Singh. (November 2009) .Performance Analysis of 32-Bit Array Multiplier with a Carry Save Adder and with a Carry-Look-Ahead Adder .

Jasbir. K. & Lalit. S. (March. 2015) Comparision Between Various Types of Adder Topologies.