YARDIMCI İŞLEMCİLİ GÖMÜLÜ ELİPTİK EĞRİ ŞİFRELEME SİSTEMİ UYGULAMASI

AN EMBEDDED ELLIPTIC CURVE CRYPTO SYSTEM IMPLEMENTATION WITH COPROCESSOR

Mehmet Sinan EROĞLU

Hacettepe Üniversitesi

Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin

ELEKTRİK ve ELEKTRONİK MÜHENDİSLİĞİ

Anabilim Dalı İçin Öngördüğü

YÜKSEK LİSANS TEZİ

olarak hazırlanmıştır.

2007

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından **ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI 'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan :….....
Prof. Dr. Selçuk GEÇİM

Üye (Danışman) : ….....
Yrd. Doç. Dr. Ali Ziya ALKAR

Üye : ….....
Yrd. Doç. Dr. Derya ALTUNAY

Üye : ….....
Yrd. Doç. Dr. Mehmet DEMİRER

Üye : ….....
Yrd. Doç. Dr. Harun ARTUNER

ONAY

Bu tez ...../...../2007 tarihinde Enstitü Yönetim Kurulunca kabul edilmiştir.

...../...../2007

Prof. Dr. Erdem YAZGAN
FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRÜ

YARDIMCI İŞLEMCİLİ GÖMÜLÜ ELİPTİK EĞRİ ŞİFRELEME SİSTEMİ UYGULAMASI

Mehmet Sinan Eroğlu

ÖZ

Bu çalışmada, gömülü şifreleme sistemleri uygulaması anlatılmaktadır. Buradaki şifreleme sistemleri eliptik eğri tabanlı olup tasarlanan donanım bir yongada sistemdir (SoC). Burada bahsedilen mühendislik yaklaşımları sadece gömülü şifreleme sistemleri için değil genel gömülü sistem uygulamaları için de kullanılabilir. Çalışmada uygulanılan sistemler eliptik eğri tabanlı olması rağmen, tasarlanan sistemin genel bir yongada sistem olması sayesinde, kullanılan metotlar eliptik eğri şifreleme sistemlerinden farklı şifreleme sistemleri için de kullanılabilinir.

Tasarlanan donanım bir yongada sistemdir. Bu donanım fiziksel olarak FPGA kullanılarak gerçekleştirilmiştir. Yongada sistemde yazılımsal işlemci çekirdeği kullanılmış, bu işlemcinin çevre birimleri FPGA kaynakları kullanılarak tasarlanmıştır.

Yongada sistem bilgisayarla dolayısıyla da dış dünya ile USB vasıtasıyla haberleşmektedir. C++ Builder ile yazılmış olan bir bilgisayar programı, USB üzerinden yongada sisteme komut göndermekte ve yongada sistemin yaptığı işlemlerin sonuçlarını USB üzerinden alarak kullanıcıya göstermektedir. Tüm şifreleme işlemleri yongada sistem tarafından yapılmaktadır. Bilgisayar programı ise sadece komut göndermek ve işlem sonuçlarını göstermek için kullanılmaktadır.

Anahtar Kelimeler: SoC, FPGA, USB, C++ Builder

Danışman: Yrd. Doç. Dr. Ali Ziya Aklar, Hacettepe Üniversitesi, Elektrik ve Elektronik Mühendisliği Bölümü

AN EMBEDDED ELLIPTIC CURVE CRYPTO SYSTEM IMPLEMENTATION WITH COPROCESSOR

Mehmet Sinan Eroğlu

ABSTRACT

In this work, embedded crypto system implementations are presented. Implemented crypto systems are based on elliptic curves and the designed hardware is a system on chip (SoC). Engineering approaches presented in this work are generic and they are not only for embedded crypto implementations, but also applicable for general purpose embedded systems. Although our implementations are based on elliptic curves, because the designed system on chip is generic, the methods can be used in other crypto systems.

Designed hardware is a system on chip and this system is physically implemented using FPGAs. We have used a softcore processor in the system on chip and we have designed peripherals for this processor using the FPGA resources.

The system on chip uses USB for communicating with the outside world. A PC program that has been written in C++ Builder sends commands to the system on chip and reads the results of the processes that are running on the system on chip using USB. All the crypto work has been done by the system on chip, the PC program just sends the commands to the system on chip and reads the results of the processes and displays them.

Keywords: SoC, FPGA, USB, C++ Builder

Advisor: Asst. Prof. Dr. Ali Ziya Aklar, Hacettepe University, Department of Electrical and Electronics Engineering

TEŞEKKÜR

Bu çalışmada bana yol göstermiş olan tez danışmanım Sayın Yrd. Doç. Dr. Ali Ziya ALKAR'a ve tezim için uzaktan desteğini esirgemeyen Sayın Dr. Levent ERTAUL'a içtenlikle teşekkür ederim

İÇİNDEKİLER DİZİNİ

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

SİMGELER ve KISALTMALAR DİZİNİ

ASIC : Application Specific Integrated Circuit (Uygulamaya Özel Tümleşik Devre)

BRAM : Block RAM (Blok RAM)

CISC : Complex Instruction Set Computer (Karmaşık Komut Kümeli Bilgisayar)

DCM : Dynamic Clock Manager (Dinamik Saat Yönetici)

DCR : Device Control Register Bus (Cihaz Denetim Yazmacı Veri Yolu)

DLL : Dynamic Link Library (Dinamik Bağ Kütüphanesi)

DLMB : Data LMB (Veri LMB'si)

DOPB : Data OPB (Veri OPB'si)

EDK : Embeded Development Kit (Gömülü Sistem Geliştirme Aracı)

FPGA : Field Programmable Gate Array (Alanda Programlanabilir Kapı Dizisi)

GCLK : General Clock (Genel Saat)

GNU : GNU is Not Unix

HDL : Hardware Descriptive Languages(Donanım Tanımlama Dilleri)

ILMB : Instruction LMB(Komut LMB'si)

IO : Input – Output (Giriş - Çıkış)

IOPB : Instruction OPB (Komut OPB'si)

IPSEC : Internet Protocol Security (IP Security)

JTAG : Joint Test Action Group (Birleşmiş Deneme Faaliyet Grubu)

Libgen : Library Generater (Kütüphane Oluşturucu)

LMB : Local Memory Bus(Yerel Bellek Yolu)

LUT : Look Up Table (Bakma Çizelgesi)

MD2 : Message Digest 2 (Mesaj Özeti 2)

MD5 : Message Digest 5 (Mesaj Özeti 5)

MHS : Microblaze Hardware Specification(Microblaze Donanım Şartnamesi)

MSS : Microblaze Software Specification(Microblaze Yazılım Şartnamesi)

MULT : Multiplier (Çarpıcı)

OPB : On-chip Peripheral Bus (Yongadaki Çevre Birimleri Veri Yolu)

Platgen : Platform Generater (Platform Oluşturucu)

PLB : Processor Local Bus (İşlemci Yerel Veri Yolu)

PLD : Programmable Logic Devices (Programlanabilir Mantık Cihazları)

PLL : Phase Locked Loop (Faza Kilitlenen Döngü)

PPC : Power PC

RISC : Reduced Instruction Set Computer (Azaltılmış Komut Kümeli Bilgisayar)

SHA-1 : Secure Hash Algorithm-1 (Güvenli Kıyım Algoritması)

SoC : System on Chip (Yongada Sistem)

VHDL : Very High Speed Integrated Circuit HDL (Çok Hızlı Tümleşik Devre HDL'i)

VOIP : Voice Over IP (IP Üzerinden Ses)

UART : Universal Asynchronous Receiver Transmitter (Evrensel Eşzamansız Alıcı Verici)

USB : Universal Serial Bus (Evrensel Seri Veri Yolu)

1.GİRİŞ

Yeniden yapılandırılabilir donanım teknolojisindeki gelişmeler, programlanabilir mantık devrelerinin birçok uygulamada kullanılabilmesine olanak sağlamıştır. Günümüzün programlanabilir cihazları yüksek hızlı ve yüksek başarımlı karmaşık sayısal tasarımların yapılabilmesine olanak sağlamaktadır. Ayrıca, internete rağbet internet üzerinden ses transferi (VOIP), elektronik ticaret, elektronik bankacılık gibi uygulamalarla büyük bir hızla artmaktadır. Ancak, internet gibi herkese açık bir ortam, kullanıcılarının bilgilerinin güvenliğini tehlikeye atmaktadır. Dolayısıyla güvenlik gereksinimi şifreleme protokol ve algoritmalarının geliştirilme zorunluluğunu doğurmuştur. İnternetin çok çeşitli sistemlerden oluşan bir ortam olması, tüm internet kullanıcılarının gereksinimlerini karşılayacak tek bir algoritmanın geliştirilmesini engellemektedir. Dolayısıyla şifreleme hizmetleri için değişken veya hızlı ve kolay değiştirilebilir bir yaklaşım izlemenin gerekliliği açıktır. Programlanabilir mantık devreleri, değişken veya kolay değiştirilebilir şifreleme uygulamalarının gerçekleştirilmesine olanak sağlamaktadır.

ASIC teknolojisine göre çok daha esnek tasarım yapabilme olanağı sağlayan programlanabilir mantık devreleri, bu özellikleri sayesinde şifreleme uygulamalarında kullanılmaya uygundur. IP Security (IPSEC) gibi çağdaş şifreleme protokolleri algoritmadan bağımsız olarak tanımlanmıştır. Bu, çeşitli algoritmaların aynı güvenlik hizmeti için kullanılabileceği anlamına gelir. Örneğin, bir protokolde RSA veya eliptik eğri sayısal imza algoritması olarak kullanılabilir. Ayrıca, günümüzün şifreleme sistemleri açık ve kapalı anahtarlı şifreleme sistemlerini birleştirmektedir. Programlanabilir mantık devreleri sayesinde aynı cihazı bu işlemlerin ikisini birden uygulamaya geçirmek için yeniden kullanmak mümkün olabilmektedir.

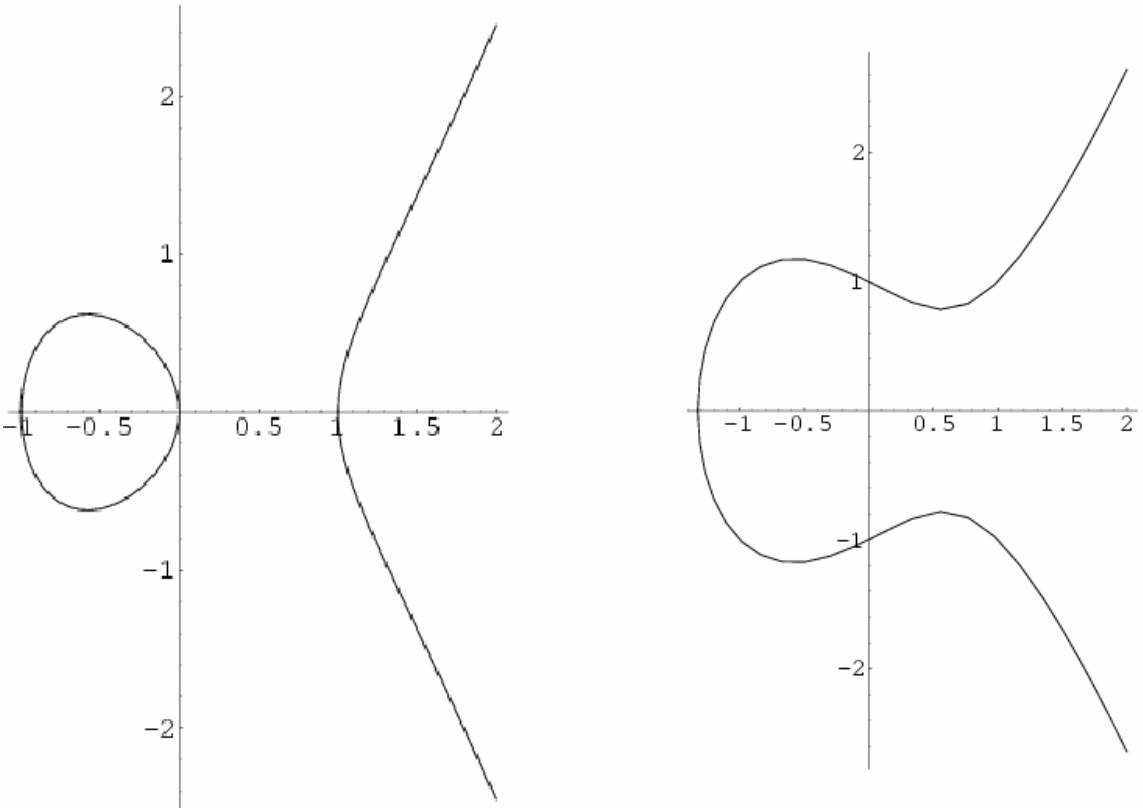
Buradaki çalışma, bir alanda programlanabilir kapı dizisi (FPGA) kullanılarak uygulamaya geçirilmiş eliptik eğri şifreleme (ECC) sistemidir. Eliptik eğri şifreleme sistemleri üzerinde seksenlerin ortasından beri çalışılmaktadır. Bu sistemlerin diğer şifreleme sistemlerinden üstün birçok özellikleri vardır. Bu sistemlerde güvenlikten ödün vermeden daha kısa anahtarlar kullanılabilmektedir. RSA gibi daha çok kullanılan 1024 bitlik bir anahtarın sağladığı güvenliği eliptik eğri sisteminde 160 bit karşılamaktadır. Ayrıca eliptik eğri şifreleme sistemleri için ayrık logaritma probleminin çözümü daha zordur.

2. ELİPTİK EĞRİLER

Eliptik eğri, E: $y^2 = x^3 + ax + b$ denkleminin grafiğidir. Buradaki a ve b sayıları amaca uygun biçimde, keyfi olarak seçilebilir. Yani a ve b sayıları istenirse rasyonel sayı, istenirse karmaşık sayı istenirse de tam sayı olabilir. Ayrıca denklemin grafiğe ∞ simgesi ile gösterilen bir noktayı da eklemek gerekmektedir. En basit biçimde ∞, y-ekseninin en tepesindeki nokta olarak tanımlanabilir. Y-ekseninin en altındaki nokta ve en üstündeki nokta özdeş olduğundan, ∞ aynı zamanda y-ekseninin en altındaki noktadır.

Gerçel sayılarla çalışırken, E'nin grafiği üç veya tek reel kökü olmasına göre iki biçimde olabilir.

Örneğin üç reel köklü $y^2 = x(x-1)(x+1)$ denkleminin ve tek reel köklü $y^2 = x^3 - x + 1$ eğrisinin grafikleri Şekil 2.1'deki gibidir.



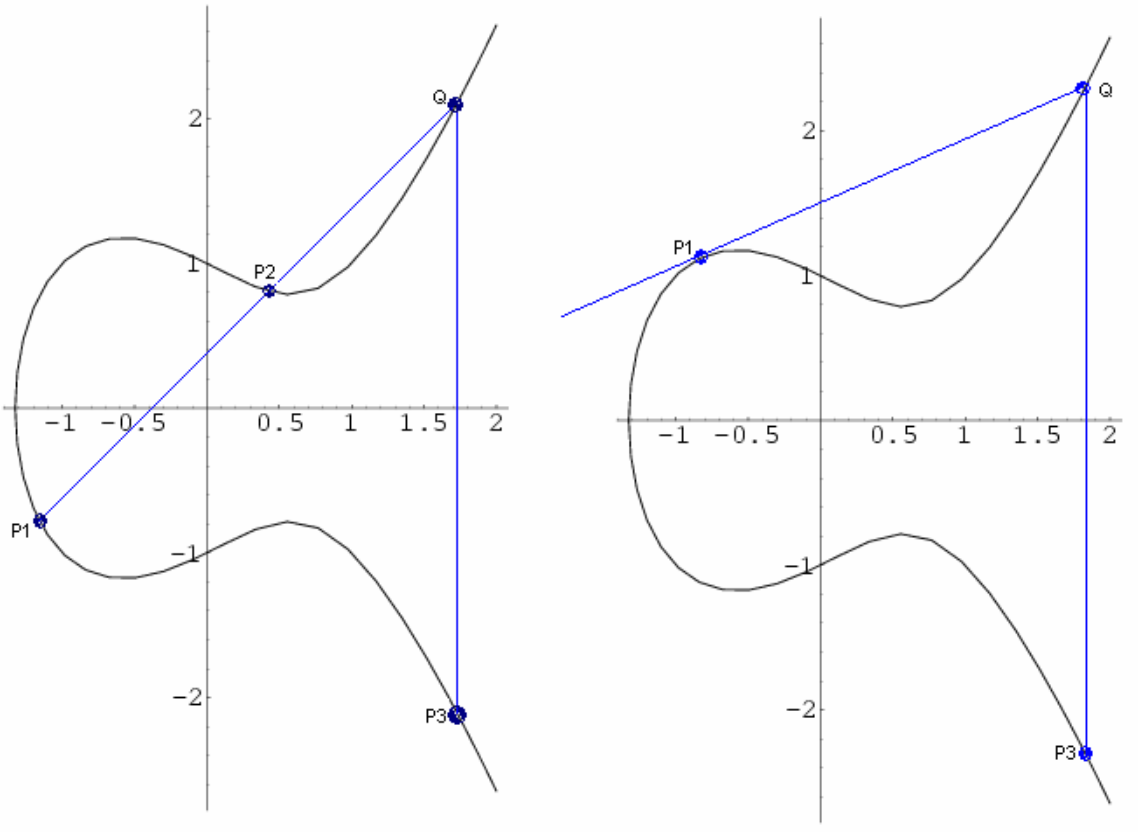
Şekil 2. 1 $y^2 = x(x-1)(x+1)$ eğrisinin ve $y^2 = x^3 - x + 1$ eğrisinin grafikleri

2.1. Eliptik Eğrilerde Toplama

E eliptik eğrisi üzerinde P_1 ve P_2 noktaları seçilmiş olsun. $P_1 \neq P_2$ ise Şekil 2.2'nin sol yarısındaki gibi P_1 ve P_2'den geçen doğruyu, $P_1 = P_2$ ise Şekil 2.2'nin sağ yarısındaki gibi P_1'e teğet doğruyu çizelim ve bu doğrunun eğriyi kestiği nokta olan Q noktasını bulalım. Q'nun x-eksenine göre simetriği P_3 olsun. Buradaki P_3 noktası P_1 ve P_2'nin E eğrisi üzerinde toplamı olarak tanımlanmaktadır. Yani

$$P_3 = P_1 + P_2 \text{ 'dir.} \tag{2.1}$$

Eğer P ile ∞'u toplamak istersek, P + ∞ = P'yi elde ederiz. P ile ∞'dan geçen doğru, y-eksenine paralel bir doğrudur ve bu doğru eliptik eğriyi P = (x, y) ve Q = (x, -y)'de keser. Q'nun x-eksenine göre simetriğini alırsak tekrar P'yi elde ederiz.



Şekil 2. 2 $P_1 \neq P_2$ durumunda ve $P_1 = P_2$ durumunda toplama

Eliptik eğri üzerinde çıkarma yapmak da mümkündür. Öncelikle, (x, y) noktası ile (x, -y) noktası üzerinden geçen doğru, y-eksenine paralel bir doğrudur ve eliptik eğriyi

∞'da keser. ∞'un x-eksenine göre simetriği yine sonsuzdur. Bu nedenle daha önce y ekseninin en üstü ve altının özdeş olduğu daha önce belirtilmiştir. Dolayısıyla,

$$(x, y) + (x, -y) = \infty \tag{2.2}$$

eşitliğini yazabiliriz.

Buradan da anlaşıldığı gibi ∞, eliptik eğri üzerinde toplama işlemine göre etkisizdir ve

$$-(x, y) = (x, -y) \text{ 'dir.} \tag{2.3}$$

Dolayısıyla, P noktasından Q noktası çıkarılmak istenildiğinde P noktasıyla –Q noktasını toplanır.

Şimdi de eliptik eğri üzerindeki toplama işlemini cebirsel olarak tanımlayalım. Eliptik eğri E,

$y^2 = x^3 + ax + b$ biçiminde olsun ve $P_1 = (x_1, y_1)$ ve $P_2 = (x_2, y_2)$ Sonra P_1 ve P_2'nin toplamı olan P_3'ü

$$P_1 + P_2 = P_3 = (x_3, y_3) \tag{2.4}$$

olarak tanımlayalım. Buradaki x_3 ve y_3 ise aşağıdaki gibidir:

$$x_3 = m^2 - x_1 - x_2 \tag{2.5}$$

$$y_3 = m(x_1 - x_3) - y_1 \tag{2.6}$$

m ise aşağıdaki gibidir:

$$P_1 \neq P_2 \quad \Rightarrow \quad m = (y_2 - y_1) / (x_2 - x_1) \tag{2.7}$$

$$P_1 = P_2 \quad \Rightarrow \quad m = (3x_1^2 + a) / (2y_1) \tag{2.8}$$

Ayrıca, eliptik eğri üzerindeki toplam işleminin birleşme özelliği,

$$(P + Q) + R = P + (Q + R) \tag{2.9}$$

ve değişme özelliği,

$$P + Q = Q + P \tag{2.10}$$

vardır.

2.2. Mod n'de Eliptik Eğri

Tamsayı bir n için, mod n'de eliptik eğrileri incelerken, daha önceden bahsedilen eliptik eğri yaklaşımlarını kullanılabilir. Örneğin,

$$E: y^2 \equiv x^3 + 2x + 3 \pmod 5 \qquad (2.11)$$

denkliğini göz önüne alalım. E üzerindeki noktalar, E denkliğini sağlayan (x, y) mod 5 ikilileridir. Bu ikililer şöyle sıralanabilir. Öncelikle x mod 5'ler 0, 1, 2, 3, 4'tür. Bunları denklikte yerine koyarsak,

$x \equiv 0 \pmod 5 \Rightarrow y^2 \equiv 3 \pmod 5 \Rightarrow$ çözümsüz

$x \equiv 1 \pmod 5 \Rightarrow y^2 \equiv 6 \equiv 1 \pmod 5 \Rightarrow y \equiv 1, 4 \pmod 5$

$x \equiv 2 \pmod 5 \Rightarrow y^2 \equiv 15 \equiv 0 \pmod 5 \Rightarrow y \equiv 0 \pmod 5$

$x \equiv 3 \pmod 5 \Rightarrow y^2 \equiv 36 \equiv 1 \pmod 5 \Rightarrow y \equiv 1,4 \pmod 5$

$x \equiv 4 \pmod 5 \Rightarrow y^2 \equiv 75 \equiv 0 \pmod 5 \Rightarrow y \equiv 0 \pmod 5$

$x \equiv \infty \pmod 5 \Rightarrow y \equiv \infty \pmod 5$

O halde E üzerindeki noktaları (1, 1), (1, 4), (2, 0), (3, 1), (3, 4), (4, 0), (∞, ∞) olarak sıralayabiliriz.

Ancak bir rasyonel sayı a/b, ab^{-1} olarak düşünülmeli ve $b^{-1}b \equiv 1 \pmod m$ denkliği unutulmamalıdır.

Şimdi Bölüm 2.1'de verdiğimiz eliptik eğri formüllerini mod n'de eliptik eğriler için verelim.

$y^2 = x^3 + ax + b$ biçiminde olsun ve $P_1 = (x_1, y_1)$ ve $P_2 = (x_2, y_2)$ Sonra P_1 ve P_2'nin toplamı olan P_3'ü

$$P_1 + P_2 = P_3 = (x_3, y_3) \qquad (2.12)$$

olarak tanımlayalım. Buradaki x_3 ve y_3 'ü hesaplamak için aşağıdaki işlemler yapılabilir:

$P_1 = P_2$ ise:

$$m = 3x^2 + a \tag{2.13}$$

$$x_3 = 2y(m^2 - 8xy^2) \tag{2.14}$$

$$y_3 = -m^3 + 12mxy^2 - 8y^4 \tag{2.15}$$

$P_1 \neq P_2$ ise:

$$u = y_1 - y_2 \tag{2.16}$$

$$v = x_1 - x_2 \tag{2.17}$$

$$x_3 = 2v(u^2 - (x_1 + x_2)v^2) \tag{2.18}$$

$$y_3 = u(3(x_1 + x_2)v^2 - 2u^2) - (y_1 + y_2)v^3 \tag{2.19}$$

2.3 Eliptik Eğrilerde Ayrık Logaritma

Klasik ayrık logaritma problemi,

$$x \equiv g^k \pmod p \tag{2.20}$$

denkliğinde x, p, ve g bilinenleriyle k'nın ne olduğunu bulma problemidir.

Ayrık logaritma problemi eliptik eğride de benzerdir. Eliptik eğri üzerinde A, B olmak üzere iki nokta düşünelim ve k da bir tam sayı olsun.

$$B = kA\ (A + A + A \ldots\ldots\ldots\ldots\ldots\ldots + A) \tag{2.21}$$

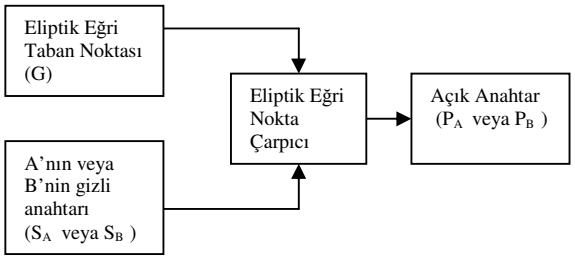
olsun. Bu eşitlikteki A ve B bilinenleriyle k'yı bulma problemine eliptik eğride ayrık logaritma problemi denir.

2.4. Eliptik Eğri Şifreleme Sitemleri ve Diffie-Hellman Anahtar Değişimi

A kişisinin B kişisine şifreli mesaj göndermek istediğini ve B'nin şifreli mesajı çözmek istediğini düşünelim.

G, eliptik eğri üzerindeki taban noktası olsun. M, şifrelenecek mesaj olsun; E, şifrelenmiş mesaj; D ise şifresi çözülmüş mesaj olsun. S_A, A'nın gizli anahtarı, S_B de B'nin gizli anahtarı olsun. P_A A'nın, P_B ise B'nin açık anahtarı olsun. K, A'nın şifreleme için kullanacağı anahtar, N de B'nin şifre çözme için kullanacağı anahtar olsun.

2.4.1. Açık Anahtar Oluşturma



Şekil 2. 3 Açık Anahtar Oluşturma Öbek Şeması

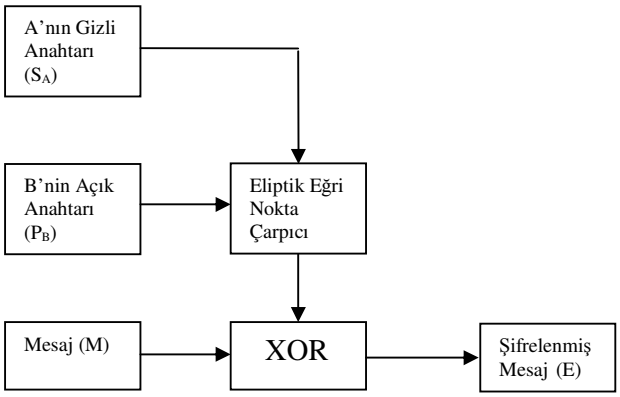
Öncelikle A ve B kişileri kendilerine ait gizli anahtarları Şekil 2.3'deki gibi oluştururlar

$$P_A = S_A * G \tag{2.22}$$

$$P_B = S_B * G \tag{2.23}$$

P_A, P_B ve G'nin açık olmasına rağmen, gizli anahtarlar S_A ve S_B , daha önceden bahsettiğimiz ayrık logaritma probleminden dolayı bulunamaz.

2.4.2. Eliptik Eğride Diffie-Hellman Anahtar Değişimi ile Şifreleme



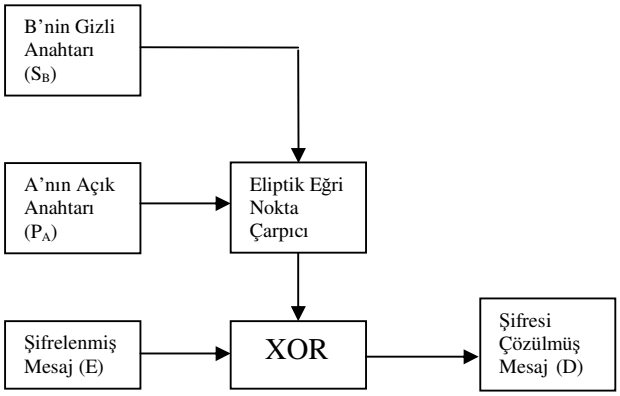
Şekil 2. 4 Eliptik Eğride Diffie-Hellman Anahtar Değişimi ile Şifreleme Öbek Şeması

Şekil 2.4'den de anlaşıldığı gibi A, mesajı aşağıdaki işlemleri gerçekleştirerek şifreler:

$$K = S_A * P_B \tag{2.24}$$

$$E = M \text{ xor } K \tag{2.25}$$

2.4.3 Eliptik Eğride Diffie-Hellman Anahtar Değişimi ile Şifre Çözme



Şekil 2. 5 Eliptik Eğride Diffie-Hellman Anahtar Değişimi ile Şifre Çözme Öbek Şeması

Şekil 2.5.'de de görüldüğü gibi B, şifreli mesajın şifresini aşağıdaki işlemleri gerçekleştirerek çözer:

$$N = S_B * P_A \quad (2.26)$$

$$D = E \text{ xor } N \quad (2.27)$$

Şifresi çözülmüş mesaj D ile M mesajının aynı olduğunu aşağıdaki gibi gösterebiliriz:

$$K = S_A * P_B = S_A * S_B * G \quad (2.28)$$

$$N = S_B * P_A = S_B * S_A * G \quad (2.29)$$

dolayısıyla N = K' dır. Buradan da anlaşıldığı gibi, A' nın şifrelemede kullandığı anahtar ile B'nin şifre çözmede kullandığı anahtar aynıdır. Buna Diffie-Hellman anahtar değişimi denir.

D'yi açarsak:

$$\begin{aligned} D &= E \text{ xor } N \\ &= M \text{ xor } K \text{ xor } N \\ &= M \text{ xor } K \text{ xor } K \\ &= M \end{aligned} \quad (2.30)$$

2.4.4. Eliptik Eğri Nokta Çarpımı

Eliptik eğri nokta çarpımını daha önce belirttiği gibi eliptik eğride toplama kuralı kullanılarak yapılabilir. Örneğin, W = 5 * P işlemini yapmak için aşağıdaki sıra takip edilebilir:

$$Q = P + P = 2 * P \quad (2.31)$$

$$R = Q + Q = 4 * P \quad (2.32)$$

$$W = R + P = 4*P + P = 5 * P \quad (2.33)$$

2.5. Kıyım (Hash) Fonksiyonları

Kıyım fonksiyonları, herhangi uzunluktaki bir dosya, resim, metin, mesaj gibi veri bloklarını belirli bir kıyım algoritmasına tabi tutarak, sabit uzunlukta sayısal bir kıyım değeri üreten fonksiyonlardır. Herhangi uzunluktaki bir dosya, resim, metin, mesaj

gibi veri kümesi kıyım fonksiyonu girdisi olabilir. Ancak çıktı kıyım değeri sabit uzunluktadır.

Kıyım algoritmaları sonucu ortaya çıkan kıyım değeri, mesajın sayısal bir parmak izidir. “Şifreleme algoritmaları” gibi bir ifadenin kıyım değeri “h7tfd8Fr” olabileceği gibi, bilgisayarımızdaki “kripto_cikti.txt” gibi bir dosyasının kıyım değeri de “a8jkdİ0” gibi bir değer olabilir. Kıyım fonksiyonu ile oluşan kıyım değeri anlamsız bir bilgidir. Kıyım fonksiyonları tek yönlüdür. Bir diğer deyişle girdi verildiğinde çıktı tektir ve kolayca bulunur. Ancak bir kıyım değeri verildiğinde fonksiyon ters çalıştırılıp onu üretmiş olan girdi bulunamaz. Ayrıca girdideki en küçük bir değişiklik çıktıyı olduğu gibi değiştirir.

Kıyım fonksiyonlarının öncelikli kullanım alanı, kasıt olmadan oluşmuş hataların farkına varmaktır. İletişim veya depolama hataları gibi doğal nedenlerden veya yetkisiz bir işlem sonucu oluşmuş hatalardan dolayı içeriği değişmiş bir veri setinin doğrulaması önceden ve sonradan alınacak kıyım değerlerinin karşılaştırması ile yapılabilir.

Bir veri bloğunu doğrulamanın, yani iletim sonucu, yetkisiz erişim ve depolama hataları vb. gibi sebeplerden dolayı içeriğin değişmediğinin kanıtlanması, veri bloğunun önceden bir kıyım değerini yaratmakla başlar. Mesajın kıyım değeri, alıcı tarafından tekrar üretilir. Eğer iki değer eşitse doğrulama sağlanmıştır.

İyi bir kıyım fonksiyonu, herhangi bir uzunluktaki veri kümesinde çalışabilmelidir. Kıyım fonksiyonunun çıktısı yani kıyım değeri sabit uzunlukta olmalıdır ve verilen herhangi bir girdi değeri x için, kıyım değeri H(x) kolayca hesaplanabilmelidir. Kıyım algoritmalarından elde edilen kıyım değerinden, orijinal veriyi elde etmek mümkün olmamalıdır. Kıyım fonksiyonlarının bu özelliğine tek yön özelliği denir.

Aynı kıyım değerini veren iki ayrı mesaj bulunamaz. Aynı biçimde bunun algoritmik bir yolu yoktur. Tek yolu olan kaba kuvvetin de yeterli zamanda yanıt vereceği şüphelidir.

Biraz önce bahsettiğimiz özellikleri yerine getiren bazı algoritmalar ortaya çıkarılmıştır. Bunlara SHA-1[5], MD2[20], MD5[19] gibi algoritmalar örnek olarak gösterilebilir.

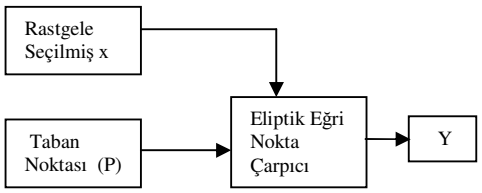
2.6 ERTAUL Şifreleme Sistemi

ERTAUL şifreleme sistemi, bir eliptik eğri şifreleme metodudur[2]. Mesajı şifrelemeden veya mesajın şifresini çözmeden önce, eliptik eğri nokta çarpıcının sonuçları kıyım (hash) fonksiyonundan geçirilir.

A kişisinin B kişisine şifreli mesaj göndermek istediğini ve B'nin şifreli mesajı çözmek istediğini düşünelim. Bu sistem için de bir eliptik eğri belirlenir ve üzerinde bir taban noktası seçilir.

P, eliptik eğri üzerindeki taban noktası olsun. M, şifrelencek mesaj olsun; E, şifrelenmiş mesaj; D ise şifresi çözülmüş mesaj olsun.

Öncelikle bir rastgele (random) sayı seçilir. Bu sayı x olsun. Bu x sayısını kullanarak Şekil 2.6'daki gibi Y oluşturulur.



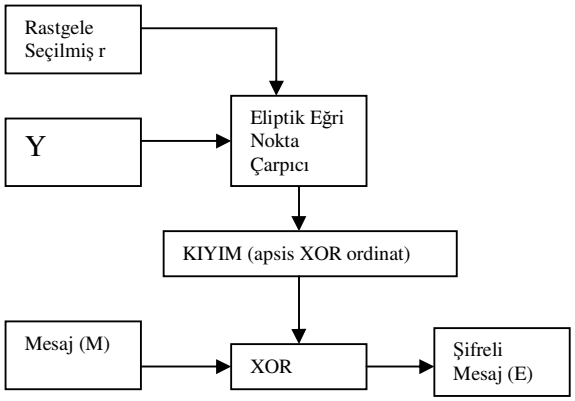
Şekil 2. 6 Y Oluşturma Öbek Şeması

Yani,

$$Y = x * P \tag{2.34}$$

eşitliğini yazılabilir.

Şimdi Y kullanılarak Şekil 2.7'deki gibi mesaj şifreleme yapılabilir.



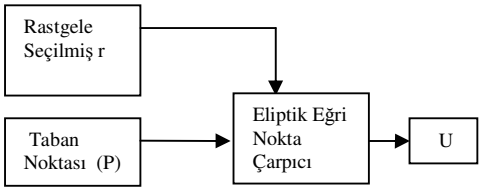
Şekil 2. 7 ERTAUL Mesaj Şifreleme Öbek Şeması

A'nın mesaj şifrelemede yaptığı işlemler aşağıdaki gibidir.

$$K = KIYIM(r * Y) \tag{2.35}$$

$$E = M \text{ xor } K \tag{2.36}$$

Daha sonra A, Şekil 2.8'deki gibi U'yu oluşturur.

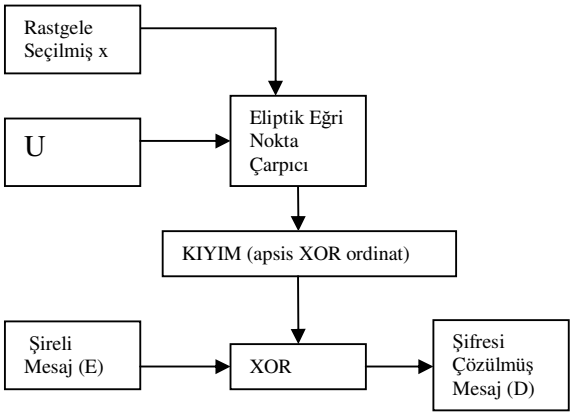


Şekil 2. 8 U Oluşturma Öbek Şeması

Şekil 2.8'den de anlaşıldığı gibi aşağıdaki eşitliği yazabiliriz.

$$U = r * P \tag{2.37}$$

Sonra A, U ve şifrelenmiş mesajı B'ye gönderir.



Şekil 2. 9 ERTAUL Şifreli Mesaj Çözme Öbek Şeması

B de Şekil 2.9'daki gibi, aşağıdaki işlemleri yaparak, şifreli mesajı çözer.

$$N = H(x * U) \tag{2.38}$$

$$D = E \text{ xor } N \tag{2.39}$$

Şifresi çözülmüş mesaj, D ile M mesajı aynıdır. Bu aşağıdaki gibi gösterilebilir:

$$r * Y = r * x * P \tag{2.40}$$

$$x * U = x * r * P \tag{2.41}$$

Dolayısıyla, $H(r * Y) = H(x * U) \Rightarrow N = K.$ (2.42)

D'yi açarsak

$$\begin{aligned} D &= E \text{ xor } N \\ &= M \text{ xor } K \text{ xor } N \\ &= M \text{ xor } K \text{ xor } K \\ &= M \end{aligned} \tag{2.43}$$

Buradan da anladığımız gibi ERTAUL sistemi kıyım fonksiyonu sayesinde çözülmesi daha zor bir şifreleme sistemidir.

3. TEKNOLOJİ ALTYAPISI

3.1. Yongada Sistem (SoC)

Yongada sistemler, bir bilgisayar sisteminin tüm parçalarının tek bir tümleşik devrede birleştiği devrelerdir. Yongada sistemler sayesinde, güç tüketimi, yongalar arası bağlantılar ve cihaz boyutları azalır.

Bir yongada sistem, bir veya daha fazla mikroişlemci, uygulamanın ihtiyaçlarını karşılayabilecek kadar çevre birimi, giriş-çıkış birimi, yongaya tümleşik bellek ve tüm bu birimlerin birbirleri ile iletişimini sağlayacak bir veri yolu içerir. Yongada sistem tasarımı, bir uygulama için yalnızca tek bir yonga kullanmayı amaçlar.

3.1.1. Yongada Sistemler ve Mikrodenetleyiciler

Bir mikrodenetleyici, bir mikroişlemciyi, biraz belleği(RAM) ve kalıcı belleği(ROM), sayaçları ve giriş-çıkış portlarını tek bir tümleşik devrede birleştiren cihazlardır. Örnek olarak Intel 8051'i verebiliriz.

Yongada sistemlerle mikrodenetleyicileri birbirinden ayırmak zordur. Bir yongada sistem, uygulamaya, ihtiyaçlara göre özelleştirilebilen sistemdir. Mikrodenetleyiciler ise daha genel amaçlı sistemlerdir.

3.2 Uygulamaya Özel Tümleşik Devreler (ASIC)

Uygulamaya özel tümleşik devreler (ASIC), en yaygın yonga türlerinden biridir. Bunlar basit tasarımlar olabilecekleri gibi, yongada sistemler gibi karmaşık tasarımların uygulamaları da olabilirler

Bir ASIC tasarımı, özel bir uygulama için yapılır. Ayrıca ASIC'ler, düşük güç tüketimi, düşük yonga alanı, yüksek saat hızlarında çalışma gibi özelliklerle donatılabilirler. Eğer çok sayıda üretim yapılacaksa, ASIC maliyetleri düşüktür.

Ancak, ASIC tasarım süreci uzundur ve üretiminden sonra yeniden yapılandırılabilmesi olanaksızdır. Ayrıca, tasarıma başlangıç maliyetleri de fazladır.

ASIC'in, çok iyi tanımlanmış ve çok sayıda üretilecek sistemler için kullanılması uygun olabilir.

3.3. Programlanabilir Mantık Cihazları (PLD)

Programlanabilir Mantık Cihazları (PLD), keyfi bir tasarımı uygulamaya geçirmek için programlanan yongalardır. PLD'ler mantık birleşimli (combinational logic) tasarımlar gibi basit tasarımları uygulamaya geçirebildikleri gibi genelde yongada sistemleri uygulamaya geçirmek için kullanılırlar.

3.3.1. Alanda Programlanabilir Kapı Dizileri (FPGA)

Alanda Programlanabilir Kapı Dizileri (FPGA), PLD'lerin bir çeşididir. FPGA'lerin genel mimarisi, yeniden yapılandırılabilir mantıksal parçalar ve programlanabilir bağlantılardan oluşur.

FPGA'ler genel amaçlı cihazlardır ve özel bir uygulama yapmak için kullanılabilirler. ASIC'e göre daha fazla güç tüketimi yaparlar ve ASIC'e göre bir tasarımı daha az etkinlikte yapılmasına izin verirler. Ayrıca yonga başına maliyeti de, ASIC'e göre pahalıdır. Ancak, FPGA'leri yeniden programlamak mümkündür bu da tasarım döngülerini kısaltır ve gerçek uygulama üzerinde deneme olanağı sağlar. FPGA'ler bu yüzden düşük sayıda üretim için uygundur ve sıkça kullanılmaktadır.

3.4 Donanım Tanımlama Dilleri (HDL)

Donanım tanımlama dilleri (HDL), donanım oluşturmakta kullanılan dillerdir ve sayısal sistem modelleme, sayısal devre sentezleme gibi amaçlarla kullanılırlar.

Donanım tanımlama dilleri yaygın olarak sayısal devre sentezlemede kullanılırlar. Tanımlanan donanım, sentez aracı tarafından teknolojiye bağımlı olarak değişen bağlantı listesine (netlist) çevrilir.

Yaygın olarak kullanılan iki HDL dili Verilog ve VHDL'dir. VHDL daha yagın olarak kullanılmakta ve daha çok araç tarafından desteklenmektedir.

VHDL, askeri ve uzay uygulamaları için tasarlanmıştır ancak günümüzde ticari ve akademik donanım projelerinde de kullanılmaktadır.

3.5 İşlemci Çekirdekleri

İşlemci çekirdeği, bir gömülü sistemde, tüm çevre birimleri çıkarıldıktan sonra kalan işlemciye denir. Yongada sistemlerde, bir veya daha fazla işlemci çekirdeği tümleşik olarak yongaya gömülür.

Yazılımsal çekirdek, FPGA'e gömülebilen HDL kodu veya HDL'in sentez aracından geçmiş bağlantı listeleridir.

Donanımsal çekirdek ise, sabit fiziksel yapıya sahiptir ve sisteme sabit hücre olarak gömülür.

3.5.1 Komut Kümesi Mimarileri

Komut kümesi mimarisi, bir işlemcinin işlemleri nasıl yürüteceğini tanımlar. Bir komut, işlemci için, X ve Y'yi toplayıp Z'ye yaz veya X'i Z'den yükle gibi basit ve temel bir yönlendirmedir.

Komut kümesi mimarileri Azaltılmış Komut Kümeli Bilgisayar Mimarisi (RISC) ve Karmaşık Komut Kümeli Bilgisayar Mimarisi (CISC) olarak ikiye ayrılır. Günümüz mikrodenetleyicilerinin ve yongada sistemlerinin işlemcilerinin mimarilerinin çoğu RISC'dir. RISC mimarili işlemciler basit komutları çok hızlı işleme işleme sokarlar. Komut uzunlukları düzgün dağılmıştır ve basit adresleme yapılarına sahiptirler.

3.5.2 Yazılımsal Çekirdekler

Yazılımsal işlemci çekirdekleri, HDL kodu veya sentez aracından geçmiş, teknolojiye bağımlı olarak değişen bağlantı listeleridir. Yazılımsal çekirdekler, öncelikle FPGA kullanıcıları arasında yaygın olarak kullanılmaktadır. Yazılımsal çekirdekler, tasarım süresini kısaltma, tasarım masraflarını azaltma, esneklik sağlama gibi avantajlarından dolayı kullanılmaktadır.

Hemen hemen tüm sayısal uygulamalar, mikroişlemci kullanmadan, uygulamaya özel mantık devreleriyle (ASIC gibi) veya programlanabilir mantık devreleriyle (FPGA gibi) hayata geçirilebilir. Sayısal uygulamalarda mikroişlemci kullanıp üzerinde gömülü yazılım koşturma yönteminin seçilmesinin iki temel nedeni vardır. Bunlar, tasarımı basitleştirme ve güncellenebilirlik olarak sıralanabilir.

Yazılım genel olarak tasarım süresini kısaltır ve tasarımın basitleştirilmesini sağlar. Aynı zamanda yazılım, uygulamada çalıştırılmadan önce, başka bir ortamda test edilebilir.

Yazılım güncellemek basit bir işlemdir hatta sistem çalışırken uzaktan bağlantı yapılarak yazılım güncellemesi yapılabilir. Ancak, programlanabilir mantık ve uygulamaya özel mantık devrelerinin güncellemesi karmaşıktır.

FPGA gibi programlanabilir mantık devrelerini güncellemek milisaniyeler mertebesinde zaman almaktadır. Ayrıca uzaktan güncelleme yapmak bu teknolojide mümkün değildir.

ASIC güncellemesi yapmak hemen hemen imkansızdır. ASIC'te güncelleme yapmak için tasarım yeniden yapılmalı ve yonga yeniden bastırılmalıdır.

Esneklik, tasarım aşamalarının basitleştirilmesi ve kısaltılması, hızlı yeniden yapılandırma ve hata ayıklama metotlarının geniş olması gibi kazanımlardan dolayı, sayısal tasarımlarda mikroişlemci yaygın olarak kullanılmaktadır.

3.6 IBM Çekirdek Birleştirici (CoreConnect) Veri yolu Mimarisi

IBM Çekirdek Birleştirici (CoreConnect) veri yolu mimarisi, yongada sistemler için tasarlanmış bir veri yolları kümesidir. Bu mimaride değişik veri yolları çeşitleri kullanılarak, yongada sistemin başarımını artırmak amaçlanmaktadır.

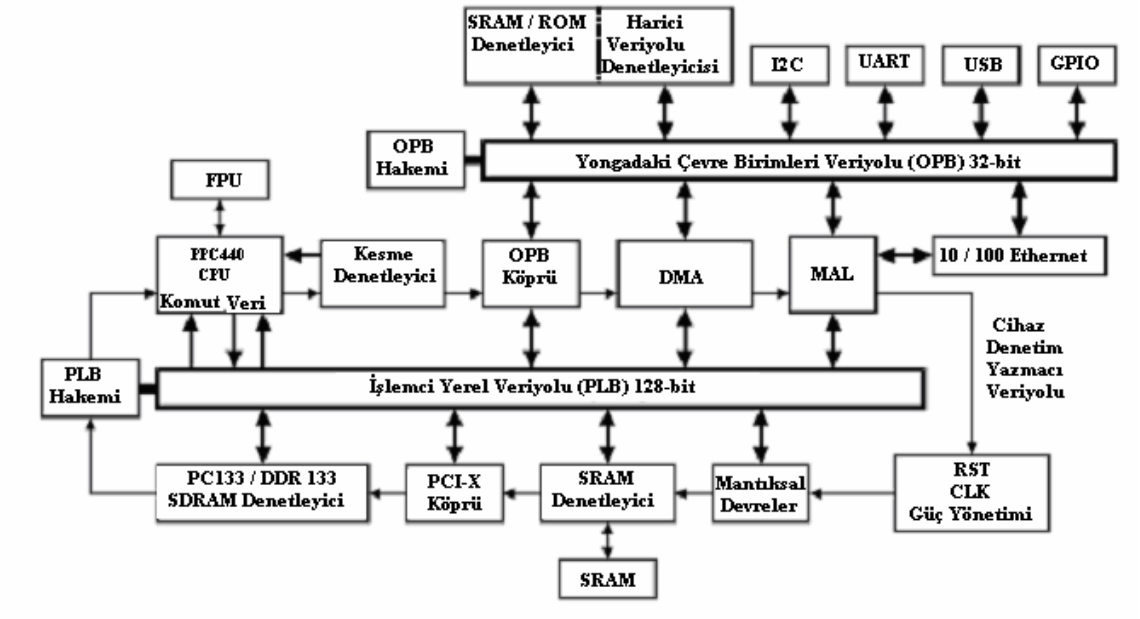
Bu mimari, üç parçadan oluşur. Bunlar, İşlemci Yerel Veri Yolu (PLB), Yongadaki Çevre birimleri Veri Yolu (OPB) ve Cihaz Denetim Yazmacı Veri Yoludur (DCR).

İşlemci Yerel Veri Yolu, yüksek başarımlı bir veri yolu olup, birden fazla işlemciyi birbirine bağlamak için veya yüksek hızlı çevre birimlerini işlemciye bağlamak için kullanılır.

Yongadaki Çevrebirimleri Veri Yolu, çok hızlı olmayan çevre birimlerini işlemciye bağlamak için kullanılır.

Cihaz Denetim Yazmacı Veri Yolu ise, yazmaç değerlerini etkin biçimde dağıtmayı sağlar.

Çekirdek Birleştirici Veri Yolu, bu saydığımız veri yollarının tekini, ikisini veya üçünü birden içeren yongada sistem tasarımlarının yapımına izin verir. Şekil 3.1, üç veri yolunun da kullanıldığı bir sistemi göstermektedir.



Şekil 3. 1 IBM CoreConnect kullanan bir yongada sistem[21]

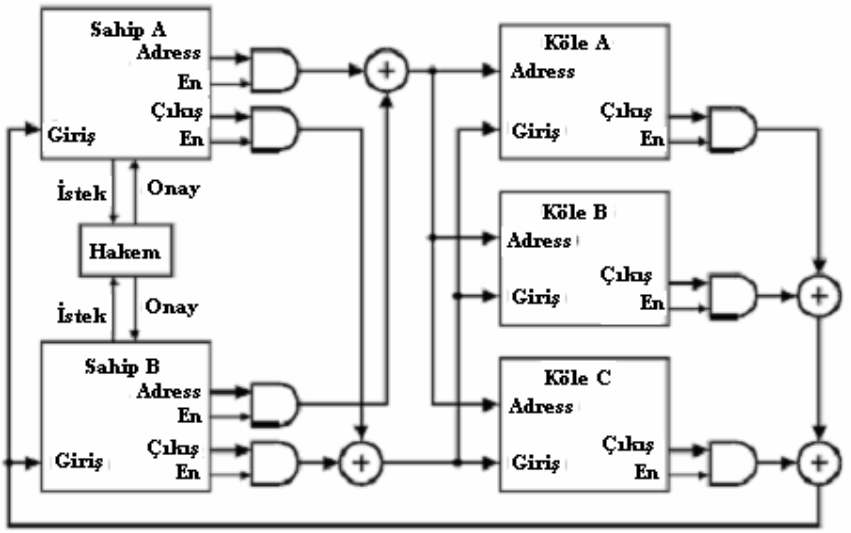
Bu saydığımız üç veri yolu, farklı saat sinyalleri ile çalışabilirler. Genellikle, OPB'nin saat sinyali, PLB'nin sinyalinden yavaştır. PLB ve OPB birbirlerine, PLB-OPB köprüsü veya OPB-PLB köprüsü ile bağlanabilirler.

IBM'in bu veri yolu mimarisi, çok esnek bir biçimde özelleştirilebilir ve uygulamadan uygulamaya çok farklılaşabilir.

ASIC uygulamalarında kullanılmak üzere IBM tarafından sağlanan Mavi Mantıksal Çekirdek (IBM Blue Logic Core) kütüphanesi ve Xilinx'in FPGA'lerde yongada tasarım yapmak için sağladığı Xilinx OPB sıkça kullanılan araçlardır.

3.6.1 Yongadaki Çevre Birimleri Veri Yolu (OPB)

OPB, kullanımı kolay olan bir veri yoludur. OPB, keyfi sayıdaki sahipten(master), keyfi sayıdaki köleye(slave) yazma ve okuma olanağı sağlar. Şekil 3.2'de küçük bir OPB sistemini görebiliriz.



Şekil 3. 2 Bir OPB uygulaması[21]

Bu veri yolları genellikle 32 bit, 64 bit veya 128 bitlik olurlar. Genellikle OPB'ye erişim tek saat döngüsünde gerçekleşir ancak yavaş OPB cihazları için daha fazla sayıda saat döngüsü de erişimde kullanılabilir.

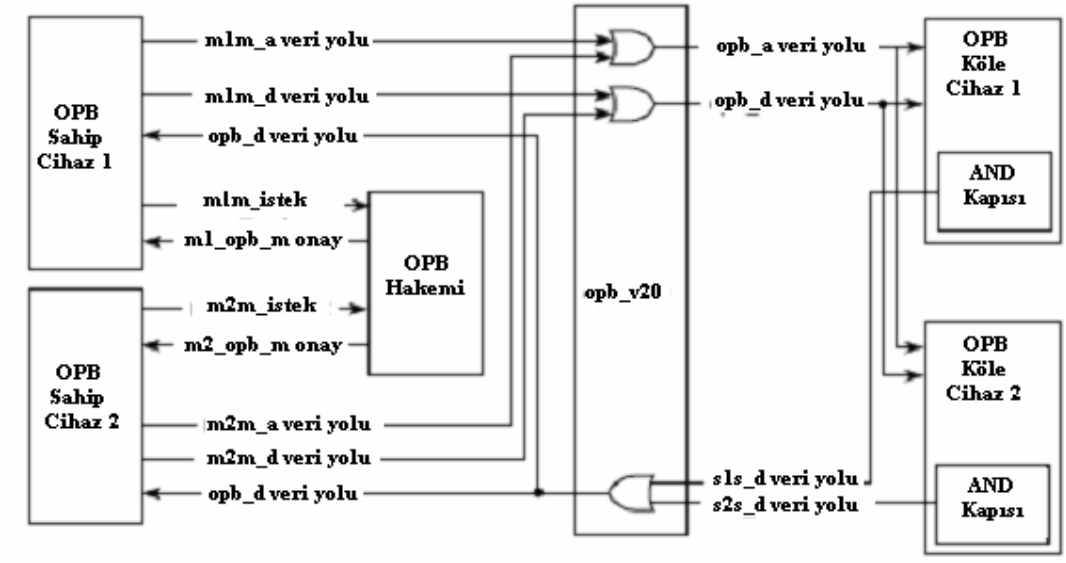
Eğer birden fazla OPB sahip veri yoluna erişebiliyorsa, erişim düzenlemek için OPB hakemi kullanılır. Böyle bir durumda, veri yoluna erişecek olan sahip cihaz, veri yolu durgun hale gelinceye kadar beklemelidir. Ayrıca, hakem de veri yoluna erişecek olan cihazın erişimini kısa bir süreliğine geciktirebilir. Ayrıca OPB sahip cihazın, veri yolu kilitleme özelliği olmalıdır. Bu özellik, sahip cihazın köle cihaza arkası arkasına erişebilmesini sağlar.

3.6.2. Xilinx OPB

Xilinx, IBM'in OPB yapısını uygulamaya geçirmiştir. Xilinx OPB'de, adreste veride 32 bit genişliğindedir. 1, 2 ve 4 baytlık erişimler gerçekleştirilebilmektedir. Xilinx OPB

hakemi 16 tane sahip cihaza izin verir ve yavaş köle cihazlar en fazla 16 saat döngüsünde sisteme cevap vermelidirler.

Şekil 3.3'te basit bir Xilinx OPB sistemi görmekteyiz.



Şekil 3. 3 Bir Xilinx OPB uygulaması[22]

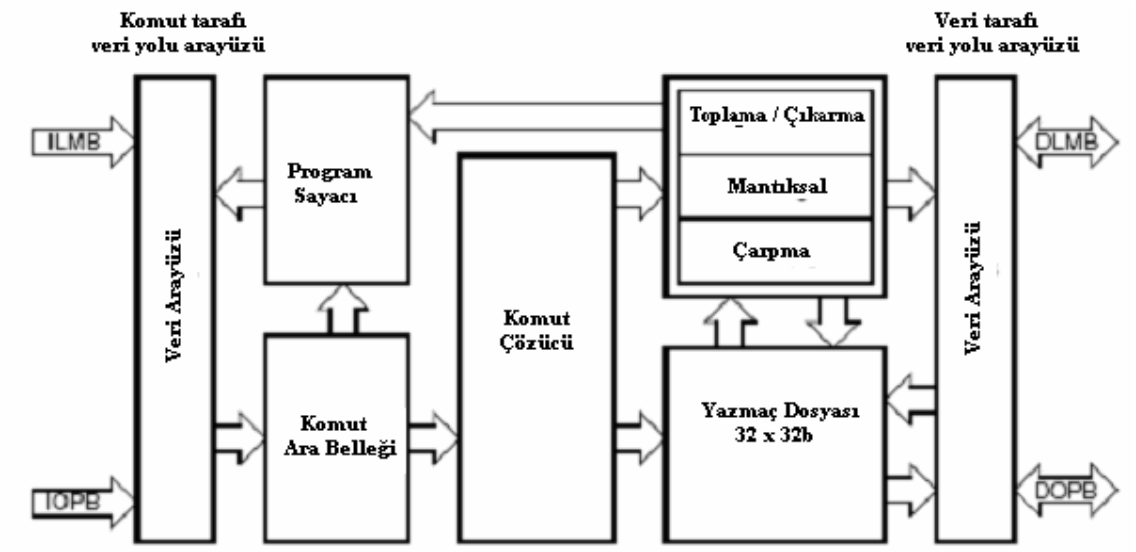
Xilinx MicroBlaze ve PicoBlaze yazılımsal işlemci çekirdekleri OPB uyumludur. Ayrıca Xilinx, Virtex II Pro serisinde bulunan IBM PPC 440 işlemcileri, OPB cihazlara bir köprü ile bağlanabilmektedir.

4. XILINX MICROBLAZE

4.1 Xilinx Microblaze Çekirdeği

Microblaze, Xilinx FPGA'lerinde uygulamaya geçirilecek gömülü sistemler için tasarlanmış bir işlemci çekirdeğidir. Microblaze ile uygulamaya geçirilecek bir yongada sistem, bir veya daha fazla microblaze, çevrebirimleri ve yonga içindeki parçaları birbirine bağlayan OPB veri yolunu içerir.

Microblaze, 32 bit RISC yapıya sahip bir yazılımsal çekirdekli işlemcidir. İşlemci üç aşamalı ardışık düzene (pipeline) ve ayrık komut ve veri cep belleğine (Harvard mimarisi) sahiptir. Microblaze çekirdeği Şekil 4.1 deki gibi açıklanabilir.



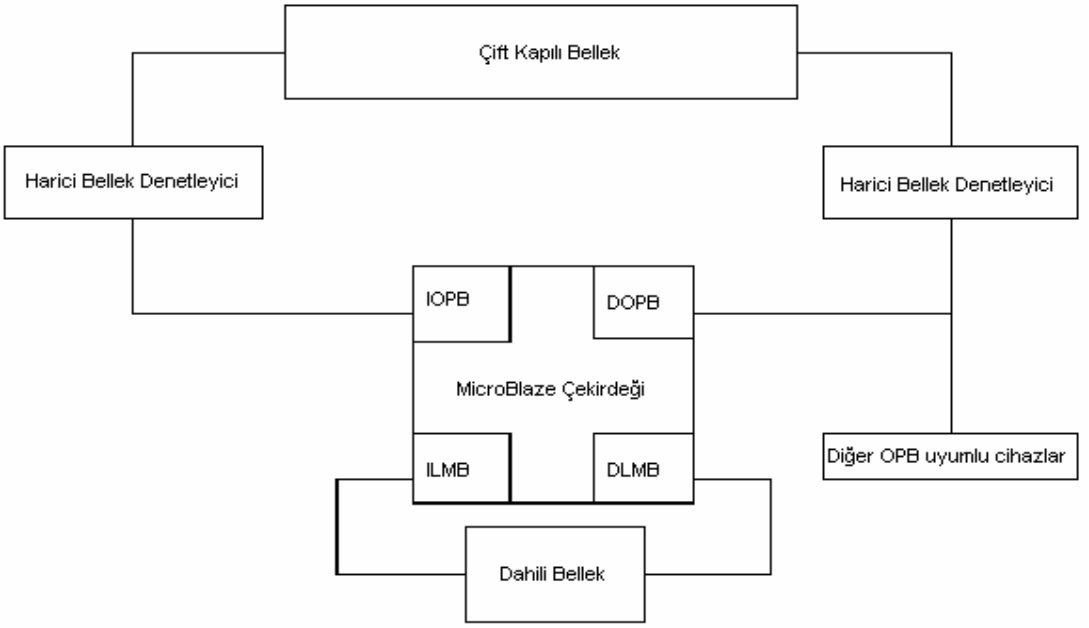
Şekil 4. 1 Microblaze çekirdeğinin öbek şeması[22]

Microblaze çekirdeği iki temel OPB sahip arayüzüne sahiptir. Bunlardan IOPB komut bölümünün yolu, DOPB ise veri bölümünün yoludur. Ayrıca çekirdek, iki yerel bellek yolu (LMB) arayüzüne sahiptir. Bunlardan ILMB komut bölümünün yolu, DLMB veri bölümünün yoludur.

Microblaze kullanılarak oluşturulan tasarımın veri yolu arayüzlerinin bir komut bir de veri bölümü olmalıdır.

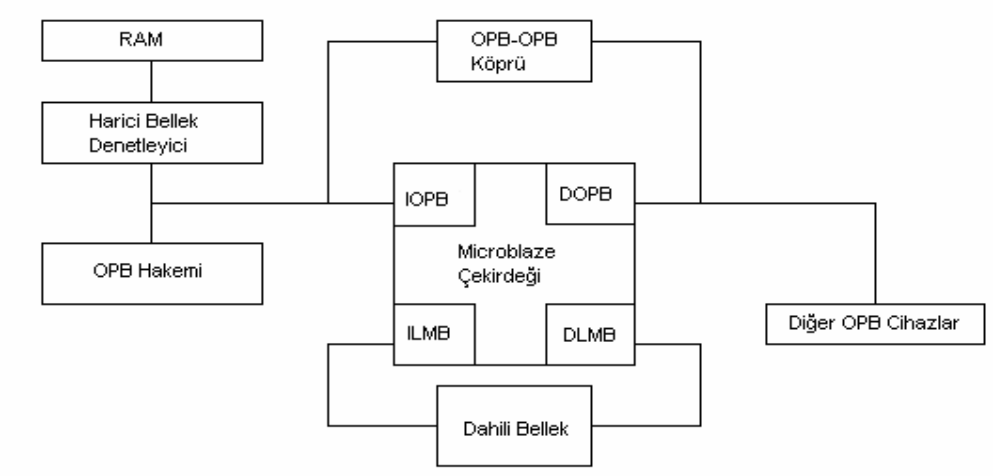
Yerel Bellek Yolu (LMB), yüksek hızlı ve eniyilenmiş bir mimariye sahip olup, Xilinx'in çift kapılı kütük belleği ile microblaze'in birbirine bağlanmasını sağlar.

Çoğu FPGA, birkaç kilobaytlık kütük belleğe (blok RAM) sahiptir. Microblaze'in üzerinde koşan yazılım bu bellekte tutulur. Geniş yastık alanı kullanılacak olan uygulamalarda, yazılım belleği yetmeyebilir. Bu durumda dışardan bellek (RAM) veya kalıcı bellek (ROM) bağlamak gerekmektedir. Dışardan bellek, OPB ile microblaze'e bağlanır. Aşağıdaki Şekillerde değişik türlerdeki belleklerin microblaze'e nasıl bağlandığını görmekteyiz.



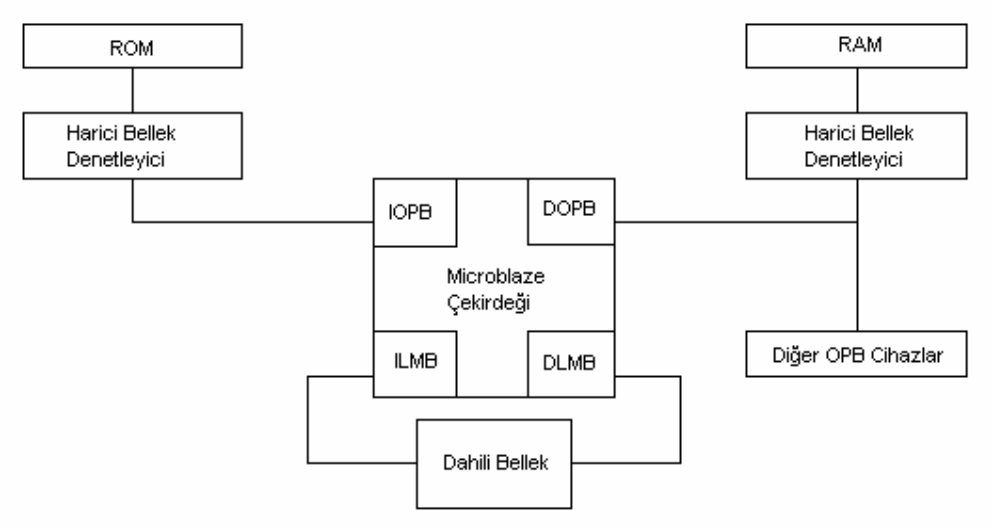
Şekil 4. 2 Microblaze'e çift kapılı bellek bağlama gösterimi

Şekil 4.2'de, çift kapılı belleğe microblaze'in nasıl bağlandığını görülmektedir.

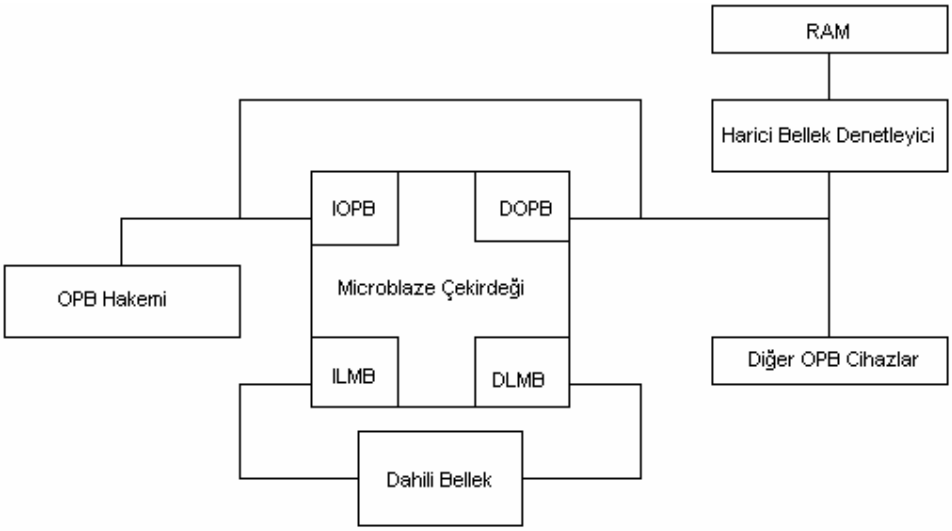


Şekil 4. 3 Microblaze'e tek kapılı bellek bağlama gösterimi

Şekil 4.3'te, microblaze'e bağlanmış bellek paylaşımlı olup, OPB hakemi, OPB çevrebirimlerinin ve microblaze'in bellek denetleyicisine erişimini düzenler.



Şekil 4. 4 Microblaze'e tek kapılı bellek ve silinmez bellek bağlama gösterimi



Şekil 4. 5 Microblaze'e paylaşımlı bellek bağlama gösterimi

Şekil 4.5'te de bellek paylaşımlıdır ve gene OPB hakemi bellek denetleyicisine erişimi düzenler.

4.2 Gömülü Sistem Geliştirme Aracı (EDK)

Xilinx'in gömülü sistem tasarımı ve yongada sistem tasarımı için sağladığı temel araç EDK'dır. EDK geniş bir OPB çevrebirimleri kümesi ve microblaze ve picoblaze yazılımsal işlemci çekirdeklerini içeren güçlü bir geliştirme aracıdır.

EDK birçok OPB çevrebirimi içerir. Bunlar arasında bellek denetleyici, UART, JTAG UART, kesme denetleyicisi sayılabilir. Bu çevrebirimleri arasında OPB hakemi de sayılabilir. Bu hakem sayesinde OPB'ye 16 sahip cihaz bağlanabilmektedir.

Microblaze içeren bir sistem, Microblaze Donanım Şartnamesi (MHS) dosyası ile tanımlanır. Bu dosya, nasıl bir veri yolu yapılandırmasının kullanıldığı ve hangi çevrebirimlerinin kullanıldığı gibi bilgilerin tanımlandığı dosyadır.

EDK içindeki platgen[23] yazılımı bu MHS dosyasından donanımın birleştirilmesini sağlar.

EDK ayrıca derleyici içermektedir ve bu derleyici GNU tabanlıdır.

EDK'nın libgen programı microblaze için yazılım sistemi oluşturur. Libgen, standart kütüphanelerin ve çevrebirimlerinin sürücülerini oluşturmada kullanılır. Libgen, Microblaze Yazılım Şartnamesi (MSS) dosyasını ve MHS dosyasını okur. MSS, sistemin hangi sürücüleri içereceğinin tanımlandığı dosyadır. Libgen koşulduğunda, gerekli kütüphane dosyalarını ve başlık (header) dosyalarını oluşturur.

Daha sonra, libgenin oluşturduğu kütüphane dosyalarından ve başlıklardan yaralanılarak oluşturulan yazılım GNU C derleyicisinden geçerek sisteme gömülür.

5. UYGULAMA

Gömülü uygulamalarda mikrodenetleyici kullanımına çok sık rastlanır hatta geleneksel olarak mikrodenetleyiciler gömülü uygulamalarda kullanılmaktadır. FPGA'ler esneklikleri sayesinde tasarımın daha uygun biçimde gerçekleştirilmesini sağlarlar. Hem mikrodenetleyicilerin sağladığı tasarım kolaylığını kullanmak hem de FPGA'lerin sağladığı esneklikten yararlanmak için uygulamamızda microblaze yazılımsal işlemci çekirdeğini kullanarak bir gömülü şifreleme sistemi oluşturduk. Daha sonra sistem başarımını artırmak için microblaze işlemcisine yardımcı işlemci ekledik. Uyguladığımız şifreleme sistemi Eliptik Eğri Şifreleme Sistemidir.

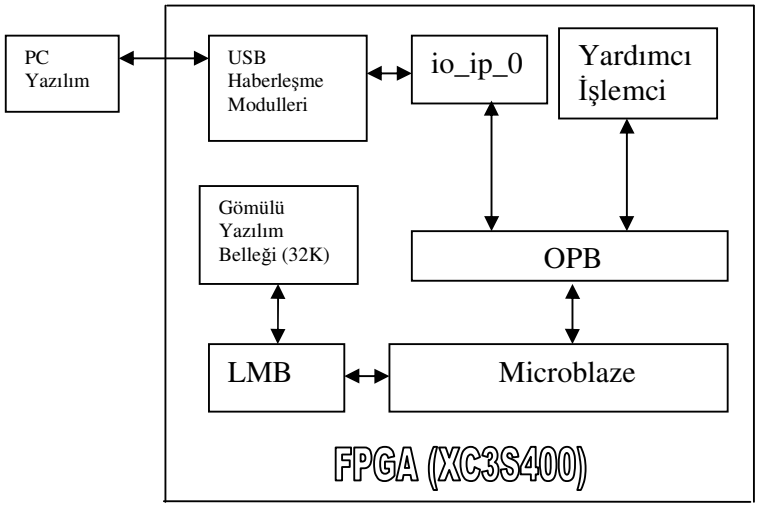
Uygulamaya geçirdiğimiz sistem bir yongada sistemdir. Yongada sistem fiziksel olarak Xilinx XC3S400 FPGA'i kullanılarak oluşturulmuştur. Yongada sistem, USB arabirimini kullanarak bilgisayar ile haberleşmektedir. Tüm şifreleme işlemleri yongada sistem tarafından gerçekleşmektedir. Bilgisayar, kullanıcı arayüzü yazılımı sayesinde USB arabiriminden yongada sisteme komut göndermekte ve sonuçları göstermektedir.

5.1 Yongada Sistem

Yongada sistem, Şekil 5.1'deki gösterildiği gibi microblaze işlemcisi, OPB ve LMB veri yolları, gömülü yazılım belleği, USB haberleşme modülleri ve io_ip_0 ve yardımcı işlemciden oluşur.

Io_ip_0, VHDL ile yazılmıştır ve microblaze ile USB haberleşme modüllerinin iletişimini sağlar.

Yardımcı işlemci sisteme sonradan, sistem başarımını artırmak için eklenmiştir. Yardımcı işlemci de VHDL kullanılarak yazılmıştır. Eliptik eğri nokta çarpımı yardımcı işlemci tarafından yapılmaktadır. Microblaze, yardımcı işlemciye eliptik eğri üzerindeki noktayı ve bu noktayı çarpacak olan sayıyı gönderir. Microblaze yardımcı işlemciye çarp komutunu gönderdiğinde yardımcı işlemci nokta çarpımını yapar ve microblaze yardımcı işlemciden verinin hazır olduğunu okuduktan sonra çarpım sonucunu da okur.



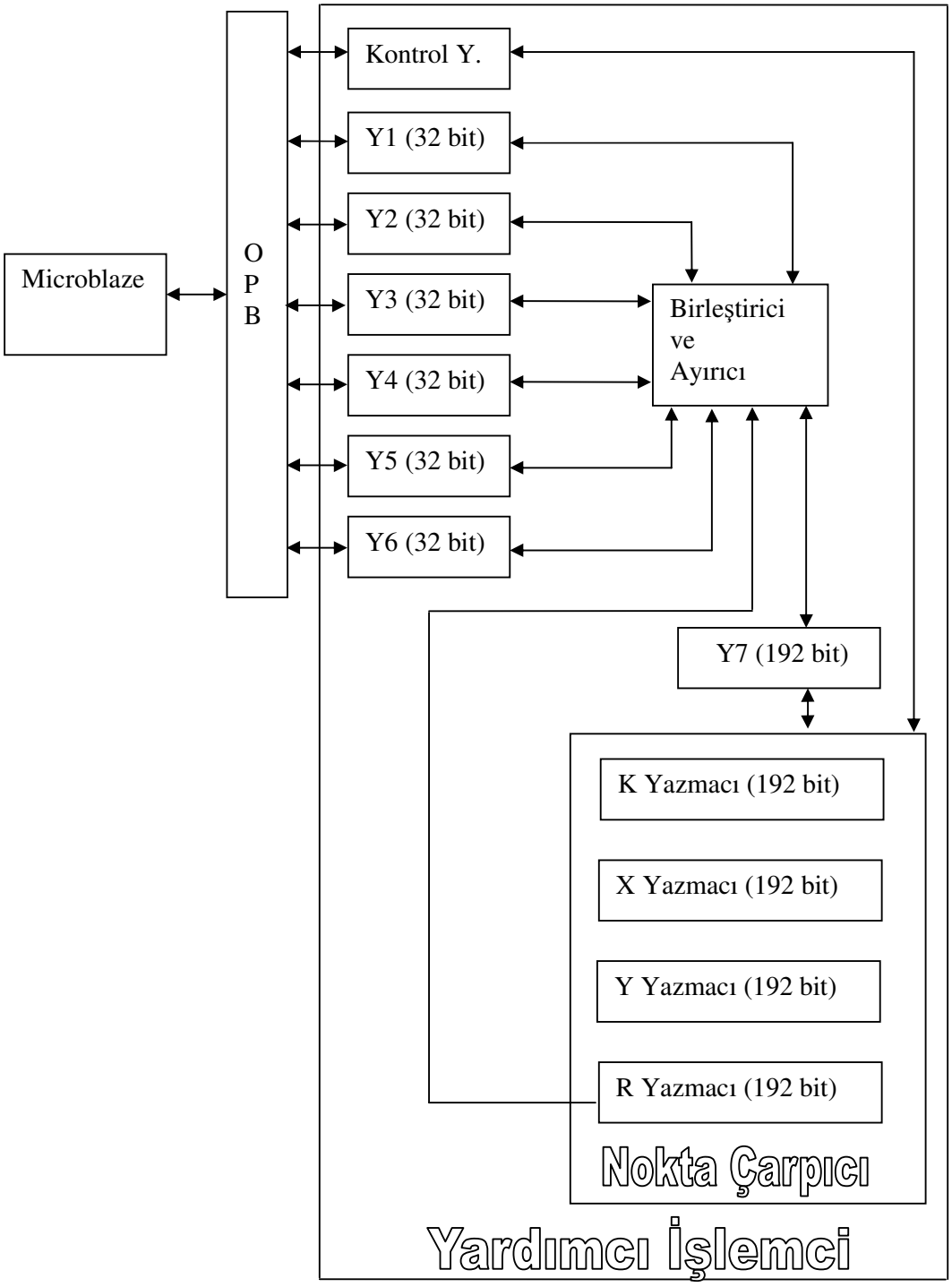
Şekil 5. 1 Yongada Sistemin Öbek Şeması

Sistem fiziksel olarak Şekil 5.2'de gösterilen Opal Kelly XEM 3001 modülü kullanılarak uygulamaya geçirilmiştir. Bu modül üzerinde Xilinx XC3S400 FPGA'i vardır ve yongada sistem bu FPGA içindedir. Modülde ayrıca USB iletişimini sağlayan Cypress'in CY7C68013 yongası bulunmaktadır.

Şekil 5. 2 FPGA Modülü

Yongada sistemde bulunan io_ip_0, VHDL ile yazılmış bir microblaze çevre birimidir. Bu birim on adeti giriş, on adeti de çıkış olmak üzere toplam yirmi adet 32 bitlik yazmaça sahiptir. Microblaze bu yazmaçlara keyfi olarak erişebilir. Io_ip_0'ın bu yazmaçları sayesinde microblaze'in USB iletişim modülleri ile bağlantısı sağlanır.

USB iletişim modülleri Opal Kelly tarafından sağlanan, önceden oluşturulmuş modüllerdir ve tasarıma dahil edilerek FPGA'in Cypress CY7C68013 yongası sayesinde USB iletişiminin gerçekleşmesini sağlar.



Şekil 5. 3 Yardımcı İşlemci Öbek Şeması

5.2 Yardımcı İşlemci

Gömülü sistem başarımını artırmak amacıyla tasarladığımız şifreleme sistemine yardımcı işlemci ekledik. Yardımcı işlemci, şekil 5.3'te de görüldüğü gibi microblaze ile OPB'yi kullanarak haberleşmektedir.

Şekil 5.3'te gösterilen Y1'den Y6'ya kadar sıralanan yazmaçlar microblaze ile yardımcı işlemcinin haberleşmesini sağlamaktadır. Bu yazmaçlar 32 bitlik yazmaçlardır. Bu 32 bitlik yazmaçlardaki veri, birleştirici/ayırıcı birimindeki birleştirici sayesinde 192 bitlik veri haline dönüştürülür ve Y7 yazmacına yazılır. Y7 yazmacındaki veri kontrol yazmacının durumuna göre nokta çarpıcıdaki K, X veya Y yazmaçlarından birine aktarılır. Daha sonra microblaze kontrol yazmacını kullanarak nokta çarpımının başlamasını sağlar. Yardımcı işlemci ise çarpım işlemini bitirdiğinde kontrol yazmacındaki bir biti mantıksal bir yapar ve çarpım sonucunu R yazmacına yazar. Microblaze de bu bitin mantıksal bir olduğunu gördüğünde R yazmacındaki çarpım sonucunu okur.

Yardımcı işlemci sistem başarımını paralel işlem yaparak artırmaktadır. Örneğin elimizde ikilik düzende A ve B sayıları olsun.

Biz ise

$$C = AxB \text{ yi} \qquad (5.1)$$

bulmak istiyoruz.

Bu işlemi gömülü yazılımla yapmak istersek aşağıdaki algoritmayı kullanabiliriz.

$B = \sum_{i=0}^{m-1} b_i 2^i \quad (b_i \in 0,1)$

C = 0;

for i from 0 to m-1 do

begin

    if b_i = 1 then

    begin

```
            C :=  C + (A  <<  i);
        end;
end
```

Fakat aynı işlemi donanımla yapmak istersek, bir döngü içinde uygun sayıda sola kaydırma işlemini yapmak yerine aynı anda sola kaydırma işlemlerini gerçekleştirip sonuçları toplayabiliriz.

5.3 Gömülü Sistem Yazılımı

Gömülü sistem yazılımı, tüm şifreleme işlemlerini, yardımcı işlemciyi de kullanarak bilgisayar yazılımından gelen komutlara göre gerçekleştirir. İki ana parçadan oluşur. Komut algılayıcı ve şifreleme işlemlerinin gerçekleştirildiği ECC parçası.

5.3.1 Komut Algılayıcı

Komut algılayıcı, io_ip_0 aygıt sürücülerini kullanarak USB iletişim modülleri ile haberleşir ve bilgisayar yazılımının verdiği komutlara göre şifreleme işlemlerinin gerçekleştirilmesini veya şifreleme işlemlerinin sonuçlarının bilgisayara gönderilmesini sağlar.

Komut algılayıcıda, POLY_DISP, P0_X_DISP, P0_Y_DISP, BASEX_DISP, BASEY_DISP, ECC_RUN, ERT_RUN, M_DISP, E_DISP, D_DISP, M_RECEIVE komutları yer almaktadır.

POLY_DISP komutunu bilgisayar yazılımı yongada sisteme gönderdiğinde, yongada sistem bilgisayar yazılımına şifreleme sisteminde kullanılan mod alıcı asal sayıyı gönderir ve bilgisayar yazılımı gönderilen bu asal sayıyı ekranda gösterir.

P0_X_DISP komutunu bilgisayar yazılımı yongada sisteme gönderdiğinde, yongada sistem bilgisayar yazılımına şifreleme sisteminde kullanılan polinomun a katsayısını gönderir ve bilgisayar yazılımı gönderilen katsayıyı ekranda gösterir.

P0_Y_DISP komutunu bilgisayar yazılımı yongada sisteme gönderdiğinde, yongada sistem bilgisayar yazılımına şifreleme sisteminde kullanılan polinomun b katsayısını gönderir ve bilgisayar yazılımı gönderilen katsayıyı ekranda gösterir.

BASEX_DISP komutunu bilgisayar yazılımı yongada sisteme gönderdiğinde, yongada sistem bilgisayar yazılımına şifreleme sisteminde kullanılan taban noktasının apsisini gönderir ve bilgisayar yazılımı gönderilen veriyi ekranda gösterir.

BASEY_DISP komutunu bilgisayar yazılımı yongada sisteme gönderdiğinde, yongada sistem bilgisayar yazılımına şifreleme sisteminde kullanılan taban noktasının ordinatını gönderir ve bilgisayar yazılımı gönderilen veriyi ekranda gösterir.

ECC_RUN komutunu bilgisayar yazılımı yongada sisteme gönderdiğinde, yongada sistem eliptik eğri şifreleme yordamını koşturur.

ERT_RUN komutunu bilgisayar yazılımı yongada sisteme gönderdiğinde, yongada sistem ERTAUL şifreleme yordamını koşturur.

M_DISP komutunu bilgisayar yazılımı yongada sisteme gönderdiğinde, yongada sistem bilgisayar yazılımına şifrelenecek mesajı gönderir ve bilgisayar yazılımı gönderilen mesajı ekranda gösterir.

E_DISP komutunu bilgisayar yazılımı yongada sisteme gönderdiğinde, yongada sistem bilgisayar yazılımına şifrelenmiş mesajı gönderir ve bilgisayar yazılımı gönderilen mesajı ekranda gösterir.

D_DISP komutunu bilgisayar yazılımı yongada sisteme gönderdiğinde, yongada sistem bilgisayar yazılımına şifresi çözülmüş mesajı gönderir ve bilgisayar yazılımı gönderilen mesajı ekranda gösterir.

M_RECEIVE komutunu bilgisayar yazılımı yongada sisteme gönderdiğinde, yongada sistem bilgisayar yazılımının gönderdiği şifrelenecek mesajı okur ve uygun yazmaca yazar.

5.3.2 ECC

Bu bölümde şifreleme işlemleri, yardımcı işlemci de kullanılarak gerçekleştirilmiştir. İki ayrı şifreleme sistemi işlemleri bu parça sayesinde gerçekleştirir. Bunlar, Diffie-Hellman anahtar değişimini kullanan eliptik eğri şifreleme sistemi ve ERTAUL şifreleme sistemidir.

Bu sistemlerde kullanılan eliptik eğri NIST[1] tarafından tavsiye edilmiş bir eliptik eğri olup, sistem 192 bitlik bir şifreleme sistemidir. Kullanılan eğri ve parametreler aşağıdaki gibidir.

$$y^2 = x^3 + ax + b \pmod p \tag{5.2}$$

$$a = -3 \tag{5.3}$$

Çizelge 5. 1 Eliptik Eğri Parametreleri

P	6277101735386680763835789423207666416083908700390324961279
R	6277101735386680763835789423176059013767194773182842284081
B	64210519 e59c80e7 0fa7e9ab 72243049 feb8deec c146b9b1
G_x	188da80e b03090f6 7cbf20eb 43a18800 f4ff0afd 82ff1012
G_y	07192b95 ffc8da78 631011ed 6b24cdd5 73f977a1 1e794811

Gömülü yazılımın bu bölümünde, point_multiplier, ECC, sha_1, ertaul gibi fonksiyonlar bulunmaktadır ve bu fonksiyonlar sayesinde Bölüm 1 'de bahsedilen şifreleme sistemleri gerçekleştirilmektedir.

Yardımcı işlemci eklenmeden önce elliptic_double ve elliptic_add fonksiyonları eliptik eğri üzerindeki iki noktanın toplanıp gene eliptik eğri üzerindeki üçüncü noktanın elde edilmesi işlemini gerçekleştiriyorlardı. Elliptic_double, toplanacak iki noktanın eşit olduğu durumda, elliptic_add ise toplanacak iki noktanın birbirinden farklı olduğu durumda kullanılıyordu. Elliptic_subtract fonksiyonu ise eliptik eğri üzerinde çıkarma işlemini gerçekleştiriyordu. Yardımcı işlemci eklendikten sonra bu fonksiyonlar gömülü yazılımdan kaldırıldı ancak bu fonksiyonlarda kullanılan algoritmalar paralelleştirilerek yardımcı işlemciye aktarıldı.

Elliptic_double fonksiyonu aşağıdaki gibi işlemekteydi. Yardımcı işlemci sisteme eklenmeden önce aşağıdaki aşamalardaki basamaklar ardışık olarak sırayla microblaze tarafından işleniyordu ama yardımcı işlemci sayesinde bu aşamalardaki basamaklar aynı anda paralel olarak işlendi. Bu sayede sistem başarımı artırıldı.

1.Aşama:

$$x^2 = x^2 \tag{5.4}$$

$$y^2 = y^2 \tag{5.5}$$

$$xy = x*y \tag{5.6}$$

2. Aşama:

$$xy^2 = (xy)_{1.aşama} * y \tag{5.7}$$

$$2x^2 = (x^2)_{1.aşama} << 1 \tag{5.8}$$

$$2y = y << 1 \tag{5.9}$$

3. Aşama:

$$3x^2 = (x^2)_{1.aşama} + (2x^2)_{2.aşama} \tag{5.10}$$

$$4xy^2 = (xy^2)_{2.aşama} << 2 \tag{5.11}$$

$$8xy^2 = (xy^2)_{2.aşama} << 3 \tag{5.12}$$

$$4y^2 = ((2y)_{2.aşama})^2 << 2 \tag{5.13}$$

4. Aşama:

$$m = (3x^2)_{3.aşama} + a \tag{5.14}$$

$$4y^4 = (y^2)_{1.aşama} + (4y^2)_{3.aşama} \tag{5.15}$$

5. Aşama:

$$m^2 = (m_{4.aşama})^2 \tag{5.16}$$

$$8y^4 = (4y^4)_{4.aşama} << 1 \tag{5.17}$$

6. Aşama:

$$u = (m^2)_{5.aşama} - (8xy^2)_{3.aşama} \tag{5.18}$$

7. Aşama:

$$v = u_{6.aşama} - (4xy^2)_{3.aşama} \tag{5.19}$$

$$x_3 = (2y)_{2.aşama} \cdot (u)_{6.aşama} \tag{5.20}$$

8. Aşama:

$$w = m_{4.aşama} * v_{7.aşama} \quad (5.21)$$

9. Aşama:

$$y^3 = -(w_{8.aşama}) - (8y^4)_{5.aşama} \quad (5.22)$$

Elliptic_add fonksiyonu aşağıdaki gibi işlemekteydi. Yardımcı işlemci sisteme eklenmeden önce aşağıdaki aşamalardaki basamaklar ardışık olarak sırayla microblaze tarafından işleniyordu ama yardımcı işlemci sayesinde bu aşamalardaki basamaklar aynı anda paralel olarak işlendi. Bu sayede sistem başarımı artırıldı.

1. Aşama:

$$u = y_1 - y_2 \quad (5.23)$$

$$t = y_1 + y_2 \quad (5.24)$$

$$v = x_1 - x_2 \quad (5.25)$$

$$w = x_1 + x_2 \quad (5.26)$$

2. Aşama:

$$u^2 = (u_{1.aşama})^2 \quad (5.27)$$

$$v^2 = (v_{1.aşama})^2 \quad (5.28)$$

3. Aşama:

$$v^3 = v_{1.aşama} * (v^2)_{2.aşama} \quad (5.29)$$

$$wv^2 = w_{1.aşama} * (v^2)_{2.aşama} \quad (5.30)$$

4. Aşama:

$$a = (u^2)_{2.aşama} - (wv^2)_{3.aşama} \quad (5.31)$$

$$tv^3 = t_{1.aşama} * (v^3)_{3.aşama} \quad (5.32)$$

5. Aşama:

$$2a = (a)_{4.aşama} << 1 \quad (5.33)$$

6. Aşama:

$$c = (wv^2)_{3.aşama} - (2a)_{5.aşama} \quad (5.34)$$

$$x_3 = v_{1.aşama} - (2a)_{5.aşama} \quad (5.35)$$

7. Aşama:

$$uc = u_{1.aşama} * c_{6.aşama} \quad (5.36)$$

8. Aşama:

$$y_3 = (uc)_{7.aşama} - (tv^3)_{4.aşama} \quad (5.38)$$

Point_multiplier fonksiyonu, eliptik eğri üzerinde seçilen bir noktayla bir skalerin çarpılmasıdır. Kullanılan algoritma iki katını al ve ekle (double and add) algoritmasıdır. Algoritma aşağıdaki gibidir ve bu yardımcı işlemci tarafından gerçekleştirilmektedir. Microblaze yardımcı işlemciye uygun parametreleri gönderdikten sonra çarp komutunu gönderir ve çarpım sonucunu yardımcı işlemciden okur.

Q = k*P'yi hesaplamak istiyoruz. K = $\sum_{i=0}^{m-1} b_i 2^i$ ($b_i \in 0,1$) olsun ve P'de eliptik eğri üzerinde bir nokta olsun.

```
Q := P;
for i from m-2 downto 0 do
begin
    Q = elliptic_double(Q);
    if b_i = 1 then
    begin
```

```
            Q := elliptic_add(P, Q);

        end;

end
```

ECC fonksiyonu, Bölüm 1'de bahsedilen Diffie-Hellman anahtar değişimi ile eliptik eğri şifrelemesinin ve şifre çözümünün gömülü sistemde uygulamasını gösteren fonksiyondur.

Sha_1 fonksiyonu ERTAUL şifreleme sisteminde kullanılan kıyım algoritmasının uygulamaya geçirildiği fonksiyondur.

ERT fonksiyonu ise ERTAUL şifreleme sisteminin uygulamaya geçirildiği fonksiyondur.

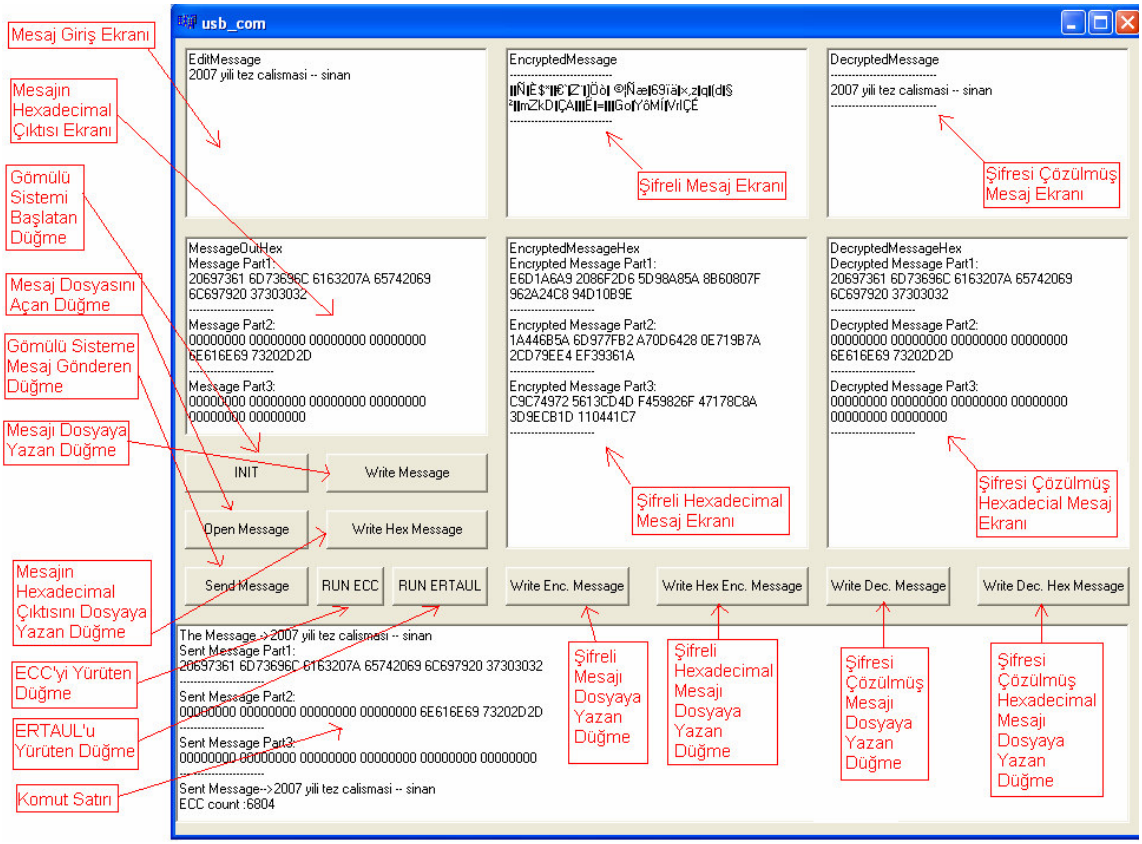
5.4 Bilgisayar Yazılımı

Bilgisayar yazılımı kullanıcının USB üzerinden gömülü sisteme komut gönderebilmesini ve gömülü sitemin aldığı sonuçların kullanıcı tarafından görülebilmesini sağlar.

Yazılım USB haberleşmesini Opal Kelly[24]'nin sağladığı DLL ile gerçekleştirir.

Yazılımın fpga_routines, data_com ve komutların gönderildiği ve sonuçların gösterildiği ana parçası vardır. Fpga_routines parçasında FPGA'i programlayan ve gömülü sistemi başlatan yordamlar bulunur. Data_com parçasında ise USB iletişimini sağlayan yordamlar bulunur.

Bilgisayar yazılımının kullanıcı arayüzü Şekil 5.4'te görüldüğü gibidir. Bu arayüzde, kullanıcının şifrelenmesini istediği mesajı girdiği bir ekran, girilen mesajın onaltılı sayı sistemine göre karşılığının yazıldığı bir ekran, şifrelenmiş mesajı ve bu şifrelenmiş mesajın onaltılı sayı sistemine göre karşılığının yazıldığı birer ekran, şifresi çözülmüş mesajı ve bu mesajın onaltılı sayı sistemine göre karşılığının yazıldığı birer ekran, komut satırı ekranı ve komut düğmeleri vardır. Bu komut düğmeleri, gömülü sistemi başlatan düğme, gömülü sisteme kullanıcı tarafından girilen şifrelenecek mesajı gönderen düğme, ECC'yi yürüten düğme ve ERTAUL'u yürüten düğmedir.



Şekil 5. 4 Bilgisayar Yazılımının Kullanıcı Arayüzü

5.4.1 Kullanıcı Arayüzü

Kullanıcı arayüzünde kullanıcının şifrelenmesini istediği mesajı girdiği, Şekil 5.4'te de gösterilen bir ekran bulunmaktadır. Buradan kullanıcı istediği metni girebilmektedir. Bu ekranın altındaki ekranda ise, girilen metnin onaltılı sayı sistemine göre karşılığının da yazıldığı bir ekran bulunmaktadır. Benzer biçimde, bu ekranların yanında, şifrelenmiş mesajı ve şifresi çözülmüş mesajı gösteren ekranlar ve bunların onaltılı sayı sistemine göre karşılığının yazıldığı ekranlar bulunmaktadır. Mesajların onaltılı sayı sistemlerine göre karşılıklarının yazıldığı ekranları daha rahat karşılaştırma yapabilmek amacıyla kullanıcı arayüzüne eklenmiştir.

Şekil 5.4'te gösterilen INIT düğmesi, gömülü sistemi başlatan düğmedir. Bu düğmeye basıldığında yazılım FPGA kodunu USB üzerinden yükler. Dolayısıyla gömülü sistemi başlatmış ve gömülü sistemin kullanıcının göndereceği komutları işlemeye ve kullanıcının isteyeceği verileri göndermeye hazır hale getirmiş olur.

Şekil 5.4'teki Send Message düğmesi, kullanıcının şifrelenmesini istediği mesajı gömülü sisteme gönderir. Bu düğmeye basıldığında gömülü sisteme Bölüm 5.2.1'de bahsedilmiş olan M_RECEIVE komutu ve veri USB üzerinden gönderilir. M_RECEIVE komutunu alan gömülü sistemde bulunan komut algılayıcı, şifrelenecek veriyi okur ve uygun yazmaca yazar.

Şekil 5.4'teki Open Message düğmesi, bir dosyadan şifrelenecek mesajın açılabilmesini ve bu mesajın, mesaj giriş ekranında gösterilmesini sağlar. Kullanıcı açılan mesajı isterse değiştirebilir. Bu değişmiş mesajı yeni bir dosyaya yazmak için Write Message düğmesi kullanılabilir. Kullanıcı, ECC veya ERTAUL sonucunda oluşan şifreli ve şifresi çözülmüş mesajı ve bunların onaltılı sayı sistemine göre karşılıklarını da Write Enc. Message, Write Hex Enc. Message, Write Dec. Message, Write Dec. Hex Message düğmelerinden uygun olanına basarak bir dosyaya yazabilir.

RUN ECC ve RUN ERTAUL düğmeleri ise, gömülü sisteme, şifrelenecek mesajı eliptik eğri şifreleme algoritmasına veya ERTAUL algoritmasına göre önce şifreleyip sonra şifresini çözüp sonuçları bilgisayar yazılımına göndermesini sağlayan komutları gönderir. Sonuçlar daha önce bahsedilmiş olan şifreli mesajı gösteren ekranlara ve şifresi çözülmüş mesajı gösteren ekranlara gönderilir. Bu düğmelerden RUN ECC'ye basıldığında öncelikle gömülü sisteme Bölüm 5.2.1'de bahsedilen ECC_RUN komutu, RUN ERTAUL'a basıldığında ise ERT_RUN komutu gönderilir. Komutu algılayan komut algılayıcı, şifrelenecek mesajı uygun algoritmayı kullanarak önce şifreler ve bu şifrelenmiş mesajı uygun yazmaçlara yazar. Daha sonra şifrelenmiş mesajın şifresini çözer ve şifresi çözülmüş mesajı da uygun yazmaçlara yazar. Bilgisayar yazılımı ECC_RUN veya ERT_RUN komutlarının işlenmesinin tamamlanmasını bekler. Komutlar tamamlandıktan sonra sonuçlar Bölüm 5.2.1'de bahsedilmiş olan E_DISP ve D_DISP komutları gönderilerek, gömülü sistemden çekilir. Bu komutlardan E_DISP daha önce bahsedildiği gibi şifrelenmiş mesajın bilgisayar yazılımına gönderilmesini, D_DISP ise şifresi çözülmüş mesajın bilgisayar yazılımına gönderilmesini sağlar. Daha sonra bilgisayar yazılımı alınan verileri şifreli mesaj ekranlarında ve şifresi çözülmüş mesaj ekranlarında gösterir.

Son olarak kullanıcı arayüzünde Şekil 5.4'te de gösterildiği gibi komut satırı bulunmaktadır. Buradan kullanıcı gömülü sisteme komut gönderebilmektedir.

Kullanıcı komut adını yazıp enter tuşuna bastığında yazılım uygun komutu gömülü sisteme göndermektedir.

Komut satırında bulunan komutlardan en önemlileri CMDS, INIT, TEST1, TEST2, TESTECC ve TESTERT'dir. CMDS komutu, komut satırında bulunan komutları listeler. INIT komutu gömülü sistemi başlatır. Bu komut INIT düğmesi ile özdeştir. TEST1 komutu Opal Kelly üzerinde bulunan ledlere sabit veri yazar. Bu veri onaltılı sayı sisteminde AA sayısıdır. Dolayısıyla 8 adet LED'in sırasıyla birisi yanmakta yanındaki sönmektedir(açık-kapalı-açık-kapalı-açık-kapalı-açık-kapalı). TEST2 komutu modül üzerinde bulunan düğmelerden veri okumayı sağlamaktadır. TEST1 ve TEST2 komutu gömülü sistem oluşturulurken USB üzerinden iki yönlü haberleşmeyi test edebilmek amacıyla oluşturulmuştur. Komut satırında bulunan TESTECC eliptik eğri şifreleme sistemi test etmek amacıyla, TESTERT ise ERTAUL şifreleme sistemini test etme amacıyla oluşturulmuştur. Bunlar RUN ECC ve RUN ERTAUL düğmeleri ile özdeştir ancak gömülü sistemden gelen verileri komut satırına yazarlar.

5.4.2 Bilgisayar Yazılımı ve USB İletişimi

Bilgisayar yazılımı USB iletişimini Opal Kelly'in sağladığı DLL ile gerçekleştirmektedir. Bu DLL'de önemli iki adet kotarıcı(handler) bulunmaktadır. Bunlardan okUSBFRONTPANEL_HANDLE tüm modülün kotarıcısı, okPLL22150_HANDLE ise modül üzerinde bulunan PLL'in kotarıcısıdır. Bu kotarıcılar sayesinde programcı tarafından belirlenen parametreler, DLL'in sağladığı fonksiyonlara taşınır. Örneğin, modül üzerinde bulunan PLL'i istediğimiz saat frekansını FPGA için oluşturmaya kilitlemek istiyoruz. Bu durumda öncelikle pll kotarıcısını oluşturmamız sonra da istediğimiz saat frekansı için gerekli parametreleri pll'e göndermemiz gerekir. PLL sağladığı saat frekansı f = ((48MHz / Q)xP) / D formülü ile hesaplamaktadır. Bizim ise 100MHz'lik saat sinyali oluşturmak istediğimizi düşünelim bu durumda aşağıdaki sözde programı (pseudocode) yazabiliriz.

okPLL22150_HANDLE pll;

P = 400;

Q = 48;

```
D = 4;

pll = okPLL22150_Construct();

okPLL22150_SetVCOParameters(pll, P, Q);

okPLL22150_SetDiv1(pll, D);

 okPLL22150_SetOutputSource(pll, 0);

 okPLL22150_SetOutputEnable(pll, 0, true);
```

Yukarıdaki sözde programda okPLL22150_Construct fonksiyonu kotarıcıyı oluşturur. okPLL22150_SetVCOParameters fonksiyonu P ve Q parametrelerini, okPLL22150-_SetDiv1 fonksiyonu ise D parametresini PLL'e gönderir. okPLL22150_SetOutput-Source fonksiyonu ise ilk çıkışı bu P,Q ve D parametreleri ile yapılandırır ve okPLL22150_SetOutputEnable fonksiyonu ise ilk çıkışı çalışır duruma getirir.

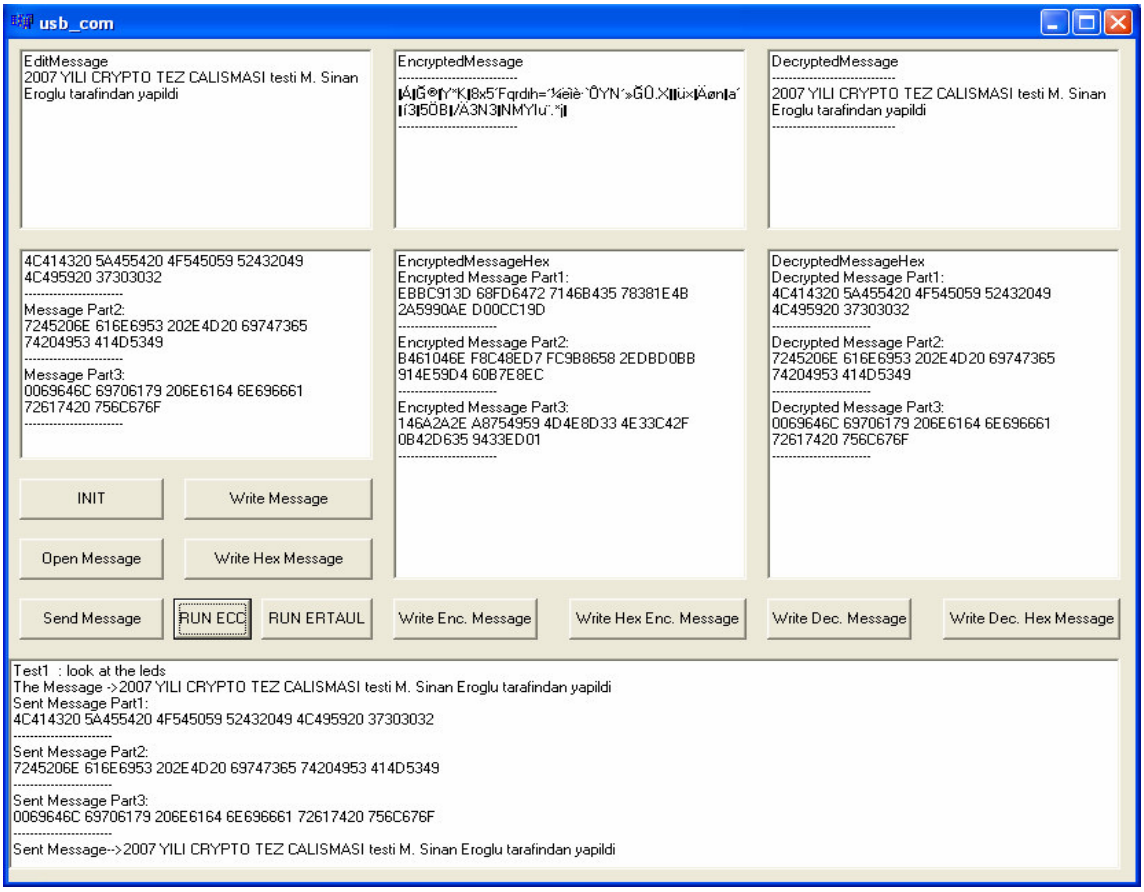
Bu DLL'de bulunan diğer önemli fonksiyonlar ise tüm modülün kotarıcısını oluşturan okUsbXEM3001v2_Construct fonksiyonu, modülü açan okUsbFrontPanel_Open fonksiyonu, FPGA'i programlayan okUsbFrontPanel_ConfigureFPGA fonksiyonu ve DLL kütüphanesini yükleyen okFrontPanelDLL_LoadLib fonksiyonudur.

Gömülü sistemi başlatmak, modülde bulunan FPGA'i programlamak ve USB iletişimini başlatmak için öncelikle okFrontPanelDLL_LoadLib fonksiyonu kullanılarak DLL yüklenmektedir. Daha sonra modülün kotarıcısı okUsbXEM3001v2_Construct fonksiyonu ile oluşturulup okUsbFrontPanel_Open fonksiyonu ile modül açılmaktadır. Daha sonra önceden bahsettiğimiz gibi PLL yapılandırılıp FPGA, okUsbFrontPanel-_ConfigureFPGAfonksiyonu ile programlanmaktadır.

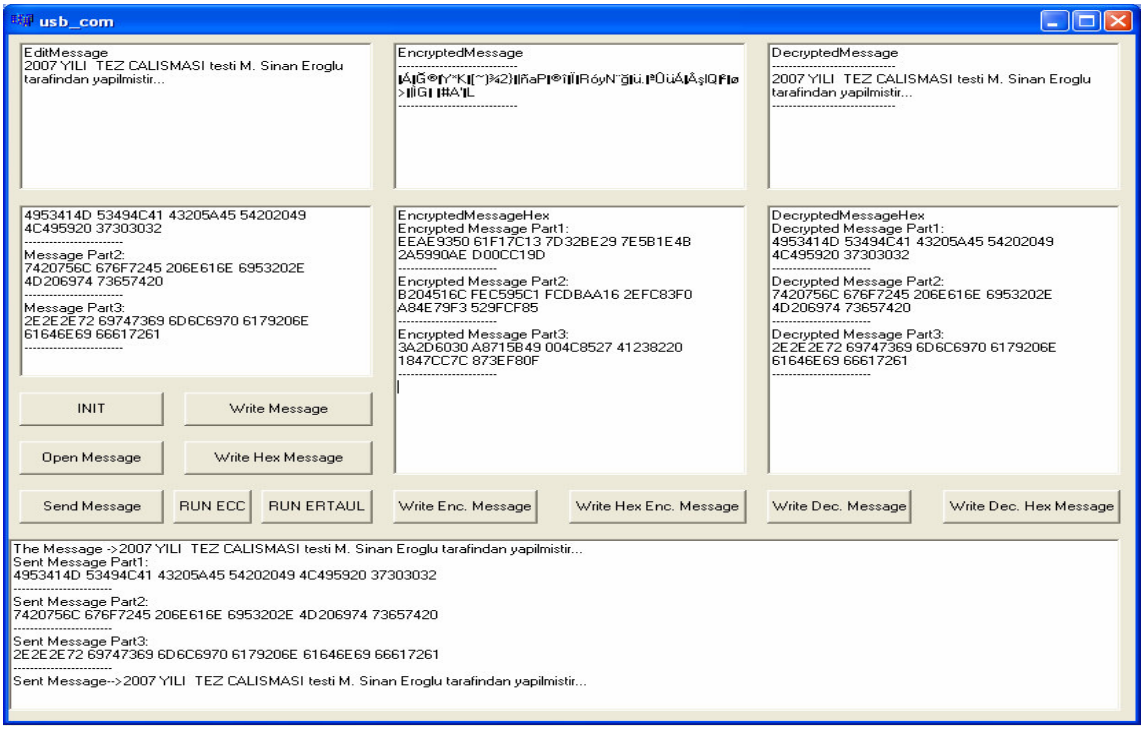
USB üzerinden veri aktarımı yapmak için ise okUsbFrontPanel_SetWireInValue ve okUsbFrontPanel_GetWireOutValue fonksiyonları kullanılmaktadır. Bu fonksiyonlardan okUsbFrontPanel_SetWireInValue fonksiyonu sayesinde gömülü sisteme veri gönderilmekte, okUsbFrontPanel_GetWireOutValue fonksiyonu sayesinde ise gömülü sistemden veri alınmaktadır.

5.5 Sistem Testleri

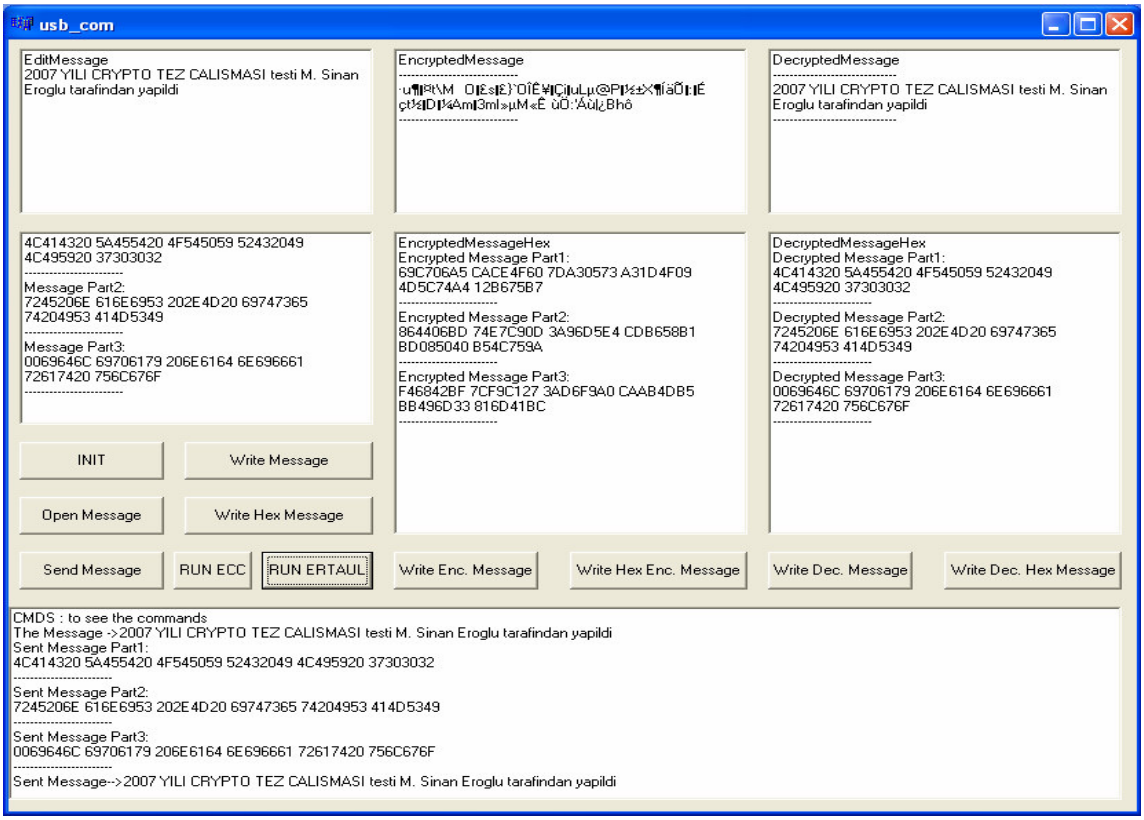
Şekil 5.5 ve 5.6'de örnek eliptik eğri şifreleme sistemi testleri, Şekil 5.7 ve 5.8'de ise örnek ERTAUL şifreleme sistemi testleri gösterilmektedir. Bu testlerin sonuçlarını rahat karşılaştırmak için, bu test sonuçları kullanılarak çizelge 5.2 oluşturulmuştur.



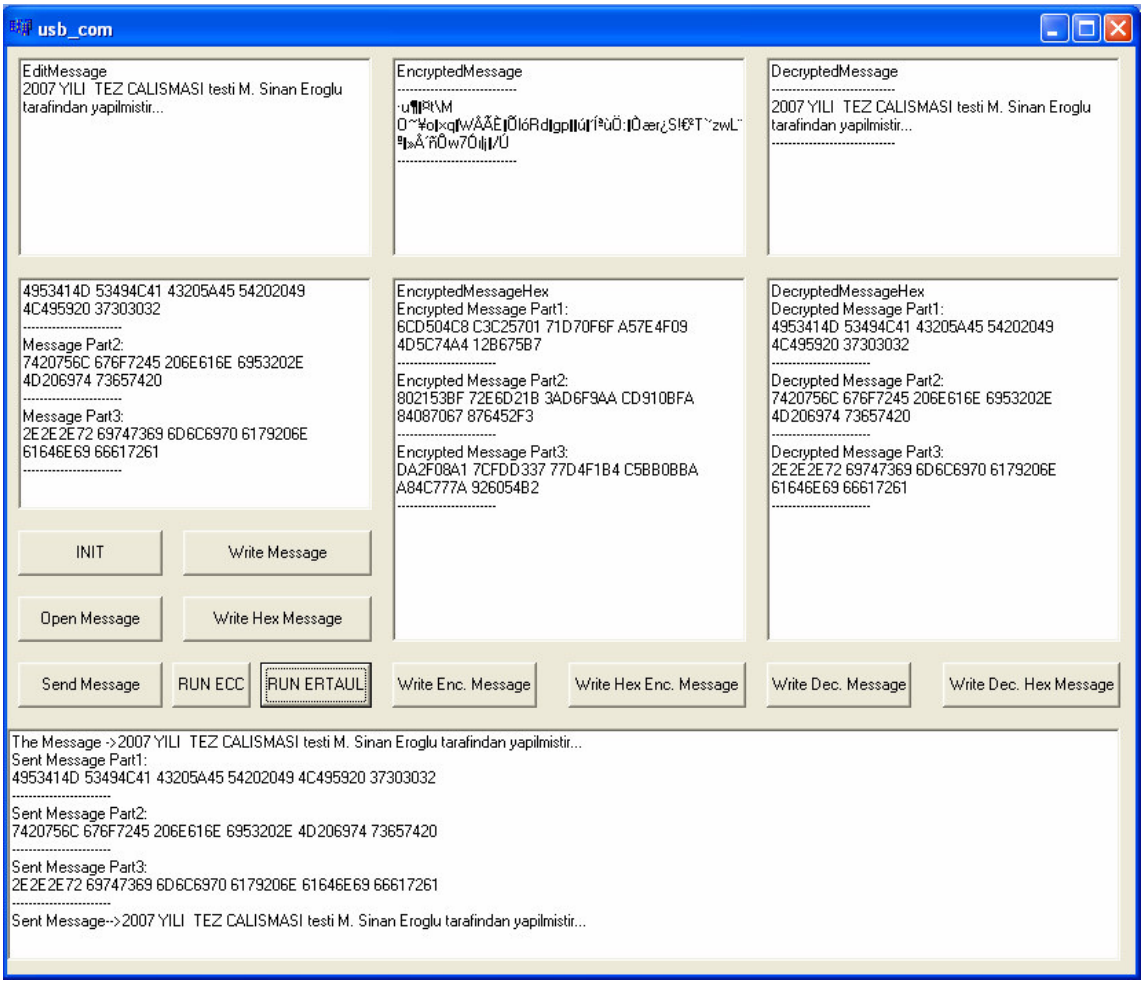
Şekil 5. 5 Eliptik Eğri Şifreleme Sistemi için ilk test



Şekil 5. 6 Eliptik Eğri Şifreleme Sistemi için ikinci test



Şekil 5. 7 ERTAUL Şifreleme Sisteminin ilk testi



Şekil 5.8 ERTAUL Şifreleme Sisteminin ikinci testi

Çizelge 5. 2 Sistem Testlerinin Sonuçları

	ECC	ERTAUL
1.Mesaj	2007 YILI CRYPTO TEZ CALISMASI testi M. Sinan Eroglu tarafindan yapildi	2007 YILI CRYPTO TEZ CALISMASI testi M. Sinan Eroglu tarafindan yapildi
1.Mesajın onaltılı sayı sistemine göre karşılığı	4C414320 5A455420 4F545059 52432049 4C495920 37303032 7245206E 616E6953 202E4D20 69747365 74204953 414D5349 2069646C 69706179 206E6164 6E696661 72617420 756C676F	4C414320 5A455420 4F545059 52432049 4C495920 37303032 7245206E 616E6953 202E4D20 69747365 74204953 414D5349 2069646C 69706179 206E6164 6E696661 72617420 756C676F
1.Şifrelenmiş mesaj	□ÁĞ®□Y*K-8x5´Fqrdıh=‘¼ëìè·`ÔYN‘» ĞÛ.X†›ü×□Äøn□a´□í3”5ÖB	·u¶[□¤t\MO□£s□£} `OÎÊ¥□ÇišuLµ@P□½±X¶ÍäÕ–: Éçt½□D†¼Am□3mI»µM«Ê

	/Ä3N3□NMYIu¨.*j□	ùÖ:'Áù¡¿Bhô
1.Şifrelenmiş mesajın onaltılı sayı sistemine göre karşılığı	EBBC913D 68FD6472 7146B435 78381E4B 2A5990AED00CC19D B461046E F8C48ED7 FC9B8658 2EDBD0BB 914E59D4 60B7E8EC146A2A2E A8754959 4D4E8D33 4E33C42F 0B42D635 9433ED01	69C706A5 CACE4F607DA30573 A31D4F09 4D5C74A4 12B675B7 864406BD 74E7C90D 3A96D5E4 CDB658B1 BD085040B54C759A F46842BF 7CF9C127 3AD6F9A0 CAAB4DB5 BB496D33 816D41BC
1.Şifresi çözülmüş mesaj	2007 YILI CRYPTO TEZ CALISMASI testi M. Sinan Eroglu tarafindan yapildi	2007 YILI CRYPTO TEZ CALISMASI testi M. Sinan Eroglu tarafindan yapildi
1.Şifresi çözülmüş mesajın onaltılı sayı sistemine göre karşılığı	4C414320 5A455420 4F545059 52432049 4C495920 37303032 7245206E 616E6953 202E4D20 69747365 74204953 414D5349 2069646C 69706179 206E6164 6E696661 72617420 756C676F	4C414320 5A455420 4F545059 52432049 4C495920 37303032 7245206E 616E6953 202E4D20 69747365 74204953 414D5349 2069646C 69706179 206E6164 6E696661 72617420 756C676F
2.Mesaj	2007 YILI TEZ CALISMASI testi M. Sinan Eroglu tarafindan yapilmistir...	2007 YILI TEZ CALISMASI testi M. Sinan Eroglu tarafindan yapilmistir...
2.Mesajın onaltılı sayı sistemine göre karşılığı	4953414D 53494C41 43205A45 54202049 4C495920 37303032 7420756C 676F7245 206E616E 6953202E 4D206974 73657420 2E2E2E72 69747369 6D6C6970 6179206E 61646E69 66617261	4953414D 53494C41 43205A45 54202049 4C495920 37303032 7420756C 676F7245 206E616E 6953202E 4D206974 73657420 2E2E2E72 69747369 6D6C6970 6179206E 61646E69 66617261
2.Şifrelenmiş mesaj	□ÁĞ®□Y*K-[~)¾2} □lñaP“®î… ÏŸRóyN¨ğƒü.□ªÛüÁ•ÅşlQ □²□ø>‡lİG□ ,#A'…L	·u¶□¤t\MO~¥o□×q□WÂÃÈ□ ÕlóRd‡gp□„ú ‘ÍªùÖ:□Òær¿S!€²T`’zwL¨º »Å´ñÔw7Ói¡□/Ú
2.Şifrelenmiş mesajın onaltılı sayı sistemine göre karşılığı	EEAE9350 61F17C13 7D32BE29 7E5B1E4B 2A5990AE D00CC19 DB204516C FEC595C1FCDBAA 16 2EFC83F0 A84E79F3 529FCF 853A2D6030 A8715B49004C852 741238220 1847CC7C 873EF80F	6CD504C8 C3C25701 71D70F6F A57E4F09 4D5C74A4 12B675B7 802153BF 72E6D21B3AD6F9AA CD910BFA 84087067 876452F3 DA2F08A17CFDD33777D4F1B4 C5BB0BBAA84C777A926054B2
2.Şifresi çözülmüş mesaj	2007 YILI TEZ CALISMASI testi M. Sinan Eroglu tarafindan yapilmistir...	2007 YILI TEZ CALISMASI testi M. Sinan Eroglu tarafindan yapilmistir...
2.Şifresi çözülmüş mesajın onaltılı sayı sistemine göre karşılığı	4953414D 53494C41 43205A45 54202049 4C495920 37303032 7420756C 676F7245 206E616E 6953202E 4D206974 73657420 2E2E2E72 69747369 6D6C6970 6179206E 61646E69 66617261	4953414D 53494C41 43205A45 54202049 4C495920 37303032 7420756C 676F7245 206E616E 6953202E 4D206974 73657420 2E2E2E72 69747369 6D6C6970 6179206E 61646E69 66617261

Çizelge 5.2 incelendiğinde, her iki mesaj için de mesajın ve şifresi çözülmüş mesajın aynı olduğu görülmektedir. Mesaj aynı kaldığında, ECC ve ERTAUL şifreleme sistemleri tarafından şifrelenmiş mesajların farklı olduğu ve mesajda küçük değişiklikler olsa bile aynı sistemde şifrelenmiş mesajların da farklı olduğu görülmektedir.

5.6 Uygulama Sonuçları

Uygulamamızda kullandığımız FPGA'in kullandığı kaynaklar yani sentez aracı sonuçları çizelge 5.3 'tedir.

Çizelge 5. 3 Kullanılan FPGA Kaynakları

Dilim Sayısı	3404 / 3584	%95
Dilim Flip Flopları Sayısı	5680 / 7168	%79
4 girişli LUT sayısı	6603 / 7168	%92
Mantıksal Kapı Sayısı	3165	
Kaydıran Yazmaç Sayısı	1311	
RAM	256Kbit	
IO Sayısı	102 / 141	%72
BRAM Sayısı	16 / 16	%100
MULT18X18 sayısı	16 / 16	%100
GCLK Sayısı	4 / 8	%50
DCM Sayısı	1 / 4	%25
Gömülü Yazılım Boyutu	14KB / 32KB	%43

Uygulamaya geçirdiğimiz sistemin daha geleneksel bir benzeri [8]'de görülmektedir. Buradaki sistemde 80MHz'de çalıştırılan bir ARM işlemcisi kullanılmıştır. Bu çalışmada, 192 bitlik sistemde eliptik eğri nokta çarpımının 69.2 milisaniyede sonuçlandırıldığı söylenmektedir. Bizim sistemimizde ise, yardımcı işlemci kullanmadığımız zaman eliptik eğri nokta çarpımı 71.20 milisaniyede gerçekleştirilmektedir. Yardımcı işlemci kullandığımızda ise 40.80 milisaniyede gerçekleşmektedir. [8]'deki çalışmada SHA'nın ise 2 milisaniyede gerçekleştirildiği söylenmektedir. Bizim sistemimizde ise ERTAUL algoritmasında kullandığımız SHA-1

algoritması 2.44 milisaniyede sonuçlanmaktadır. Bu sonuçlar çizelge 5.4'te özetlenmektedir.

Çizelge 5. 4 ARM kullanılan uygulama [8] ve Microblaze kullanılan uygulama karşılaştırması

	ARM Kullanılan Uygulama[8]	Microblaze kullanılan uygulama (Yardımcı İşlemcisiz)	Microblaze kullanılan uygulama (Yardımcı İşlemcili)
Eliptik Eğri Nokta Çarpımı	69.2 milisaniye	71.20milisaniye	40.80milisaniye
SHA	2 milisaniye	2.44 milisaniye	2.44 milisaniye

Microblaze'de ölçüm yapabilmek için, Opal Kelly modülündeki çıkış pinlerinden biri ölçüm pini olarak atanmıştır. SHA-1 veya eliptik eğri nokta çarpımı yürütülmeden önce ölçüm pini mantıksal bir yapılmış, algoritma yürütülmüş sora ölçüm pini mantıksal sıfıra çekilmiştir. Ölçüm pini üzerinde oluşan pozitif darbe Şekil 5.9, 5.10 ve 5.11'deki gibi osiloskopta gözlenmiş ve darbe genişliği ölçülmüştür. Dolayısıyla çizelge 5.4'deki Microblaze için verilmiş değerler çok az da olsa gerçek değerden fazladır.

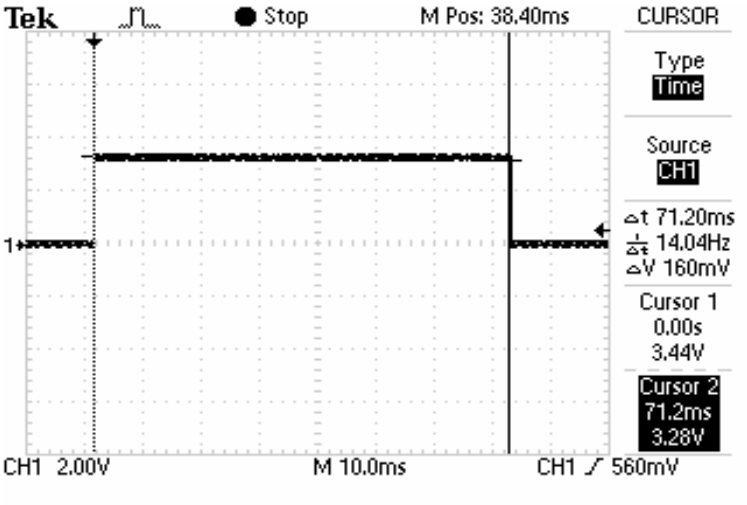
ERTAUL ve ECC arasında, USB iletişimi ve verilerin bilgisayar programı tarafından gösterilmesi de dahil olmak üzere tüm sistemin başarımının karşılaştırılmasını yapabilmek için, bilgisayar yazılımında uygulamaya geçirilmiş bir sayaç kullanılmıştır. Sayaç ERTAUL veya ECC yürütülmeden önce sıfırlanmaktadır. Daha sonra ERTAUL'u veya ECC'yi yürüten komut gömülü sisteme gönderilmekte ve gömülü sistem içinde, gönderilen bu komutun bitip bitmediğini gösteren bayrak işin henüz tamamlanmadığını gösterdiği sürece sayaç bir artırılmaktadır. İş bittikten sonra sayaç değerleri komut satırında gösterilmektedir. Bu ölçüm metoduna göre alınmış sonuçlar çizelge 5.5'de gösterilmiştir.

Çizelge 5. 5 ECC ve ERTAUL karşılaştırması

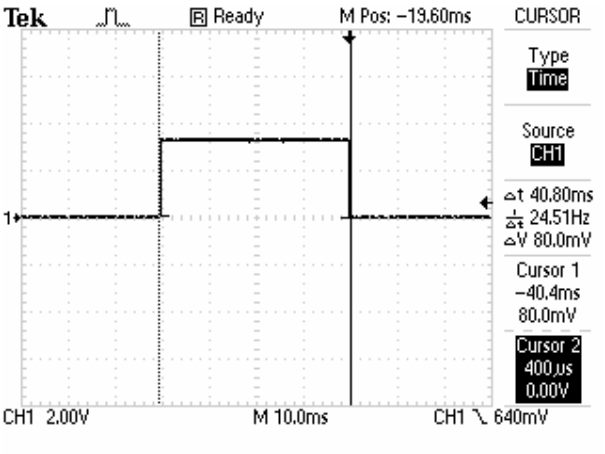
	Mesaj 1	Mesaj 2
ECC için sayaç değeri	6778	6807
ERTAUL için sayaç değeri	6751	6765

Çizelge 5.5'den de anlaşıldığı gibi ERTAUL algoritması Diffie-Hellman anahtar değişimli eliptik eğri algoritmasına göre az da olsa daha hızlı çalışmaktadır.

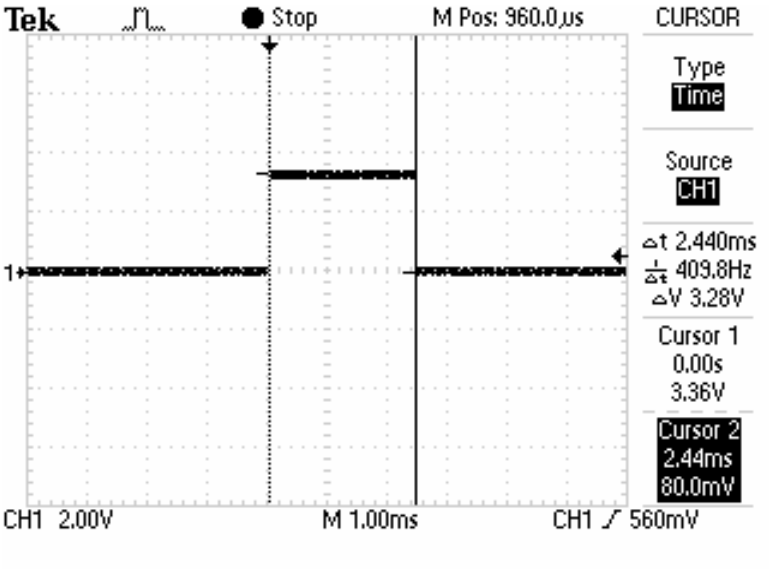
Mesaj 1 için hız farkı (6778 - 6751)/6778 ≈ 0.004 yani ‰4, mesaj 2 için ise hız farkı (6807 - 6765)/ 6807 ≈ 0.006 yani ‰6 biçiminde hesaplanabilir. Hız farkı çok az da olsa ERTAUL algoritması daha hızlı çalışmaktadır.



Şekil 5.9 Eliptik Eğri Nokta Çarpımı Zaman Ölçümü (Yardımcı İşlemcisiz)



Şekil 5.10 Eliptik Eğri Nokta Çarpımı Zaman Ölçümü (Yardımcı İşlemcili)



Şekil 5.11 SHA Zaman Ölçümü

5.7 Gelecek Çalışması

Uyguladığımız eliptik eğri şifreleme sistemi NIST tarafından tavsiye edilen 192 bitlik bir şifreleme sistemidir. Gelecekte NIST tarafından tavsiye edilen 224, 256, 384 ve 521 bitlik şifreleme sistemleri de bu çalışmada bahsedilen metotlar kullanılarak yapılabilir.

Bu sistemde bilgisayar ve gömülü sistem arasındaki iletişim USB kullanılarak gerçekleştirilmiştir. Bundan sonraki çalışmalarda USB yerine ethernet kullanılabilir.

Gelecekteki çalışmalarda tasarlanan bilgisayar yazılımının grafiksel arayüzü daha kolay kullanılabilir hale getirilebilir.

6.SONUÇ

FPGA'ler ve yazılımsal işlemci çekirdekleri tasarıma kazandırdıkları esneklik sayesinde şifreleme sistemleri uygulamaları için uygun ortamı sağlamaktadırlar. Yeniden yapılandırılabilir kaynaklar açısından zengin olan FPGA'ler bu özellikleri sayesinde başarımı yüksek sistemlerin gerçek hayata taşınmasına olanak sağlamaktadırlar. FPGA'lerin iyi yapılandırılmış dağınık mimarileri sayesinde, sentez araçları etkin uygulamalar için eniyileme yapabilmektedirler.

Eliptik eğri şifreleme sistemlerine yeniden yapılandırılabilir donanım kavramı tam oturmaktadır. Modern şifreleme protokolleri uygulamaya geçirilirken yeniden yapılandırılabilir donanım mimarilerinin sağladığı değişken ortam istenilmektedir. Burada bahsettiğimiz çalışmada eliptik eğri şifreleme sistemlerinin yeniden yapılandırılabilir donanımlar ile uygulamaya geçirilebilineceğini göstermiş olduk. Yeniden yapılandırılabilir donanım kullanmanın getirebileceği tek sınırlama, geliştirme aşamasında karşılaşılabilecek uzun süren yerleştirme ve yönlendirme(place and route) aşaması olabilir. Mevcut araçlar hızlı biçimde gelişmekte ve üreticiler sürekli yeni cihazları piyasaya sürmektedir. Gelişmeler, yakın gelecekte büyük ve çok karmaşık yongada sistem tasarımlarının daha kolay biçimde yapılabilmesini sağlayacaktır.

Bu çalışmada, Bölüm 5.4'de de bahsettiğimiz gibi başarımı iyi olan bir şifreleme sistemini yongada sistem ve yeniden yapılandırılabilir donanım kavramları ile uygulamaya geçirmiş olduk. Gömülü bir şifreleme uygulaması yapılmak istendiğinde FPGA'ler ve yazılımsal işlemci çekirdekleri ve bu yazılımsal işlemci çekirdeğine yardımcı işlemci kullanılabilir. Bu seçim tasarımcıya mikroişlemci kullanarak kazanacağı avantajların hepsini sağlamaktadır. Tasarımcı bu seçim ile ek olarak FPGA'lerin getireceği esneklikten ve paralel işlem gücünden de yararlanabilecektir.

KAYNAKLAR

1. http://csrc.nist.gov/CryptoToolkit/dss/ecdsa/NISTReCur.pdf

2. L. Ertaul, W. Lu, "ECC based Threshold Cryptography for Secure Data Forwarding and Secure Key Exchange in Mobile Ad Hoc Networks (MANET) I", Proc. of Networking 2005 International Conference, May 2005, University of Waterloo, Ontario, CA

3. 'D. Hankerson, A. Menezes, S. Vanstone "Guide to Elliptic Curve Cryptography", Springer Verlag 2004

4. I. Blake, G. Seroussi, N. Smart "Advences in Elliptic Curve Cryptography", Cambridge University Press '2005

5. http://www.itl.nist.gov/fipspubs/fip180-1.htm

6. W. Diffie and M. Hellman, "New directions in cryptography," IEEE Trans. Inform. Theory, vol. IT-22, pp. 644-654, 1976.

7. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," lEEE Trans. Inform Theory, vol. IT-31, pp. 469-472, 1985.

8. M Aydos, T Yanık, ÇK Koç, "An High-speed ECC-based Wireless Authentication Protocol on an ARM Microprocessor" IEEE, 2000

9. Scott A. Vanstone and Robert J. Zuccherato, "Elliptic Curve Cyptosystems Using Curves of Smooth Order Over the Ring Z_n,, ," lEEE Trans. Inform Theory, vol. 43, No.4, July 1997

10. Sarwono Sutikno, Andy Surya, Ronny Efendi, "An Implementation of ElGamal Elliptic Curves Cryptosystems" ," lEEE Trans. Inform Theory, pp. 483-486, 1998.

11. Sarwono Sutikno, Andy Surya, Ronny Efendi, "Design and Implementation of Arithmetic Processor F 2^{166} for Elliptic Curves Cryptosystems" ," lEEE Trans. Inform Theory, pp. 647-650, 1998.

12. Constantin Popescu, “An Identification Scheme Based on the Elliptic Curve Discrete Logarithm Problem” Trans. Inform Theory, pp. 624-625, 2000.

13. Sarwono Sutikno, Andy Surya, “An Architecture of F 2^N Multiplier for Elliptic Curves Cryptosystems” ,” IEEE International Symposium on Circuits and Systems, May 28, 2000

14. Sarwono Sutikno, Andy Surya, Ronny Efendi, “Design and Implementation of Arithmetic Processor F 2^{166} for Elliptic Curves Cryptosystems” ,” lEEE Trans. Inform Theory, pp. 647-650, 1998.

15. MJ Potgieter and BJ van Dyk, “Two Hardware Implementations of the Group Operations Necessary for Implementing an Elliptic Curve Crypto System Over a Characteristic Two Finite Field” IEEE Africon, 2002

16. http://standards.ieee.org/reading/ieee/std_public/description/dasc/1076.3-1997_desc.htm

17. G. B. Agnew, R. C. Mullin, and S . A. Vanstone, “An implemention of Elliptic Curve Cryptosystems Over F 2^{155}" IEEE Journal on Selected Areas in Communications, Vol 11, No.5, June 1993

18. Bülent Sankur, “Bilişim Sözlüğü”, 2005

19. http://tools.ietf.org/html/rfc1321

20. http://tools.ietf.org/html/rfc1319

21. The CoreConnect™ Bus Architecture (White Paper), International Business Machines Corporation (IBM), 1999. http://www.01.ibm.com/chips/techlib/techlib.nsf/techdocs/852569B20050FF77852569910050C0FB

22. Microblaze™ Hardware Reference Guide,Xilinx Inc, 2006

23. Microblaze™ Software Reference Guide,Xilinx Inc, 2006

24. www.opalkelly.com

Ek – 1 : İngilizce Türkçe Terimler Sözlüğü

address : Adres

address bus : Adres yolu

addressing : Adresleme

available : Mevcut

buffer : Yastık, tampon

bus : Yol

chip : Yonga

command : Komut

compiler : Derleyici

counter : Sayaç

cryptography - crypto: Şifreleme

data bus : Veri yolu

decryption : Şifre çözme

development : Geliştirme

device : Cihaz

digit : Basamak

encryption : Şifreleme

gata : Kapı

hash : Kıyım

hexadecimal : Onaltılı

input : Giriş

input / output : Giriş / çıkış

instruction : Komut

interface : Arabirim

kit : araç

local : Yerel

microcontroller : Mikrodenetleyici

microprocessor : Mikroişlemci

output : çıkış

performance : Başarım

place : Yerleştirme

point multiplier : Nokta çarpıcı

processor : İşlemci

private key : Gizli anahtar

public key : Açık anahtar

register : Yazmaç

route : Yönlendirme

security : Güvenlik

soft core : Yazılımsal çekirdek

source : Kaynak

specification : Şartname

ÖZGEÇMİŞ

Adı Soyadı : Mehmet Sinan EROĞLU

Doğum Yeri : Sandıklı/AFYON

Doğum Yılı : 1982

Medeni Hali : Bekar

Eğitim ve Akademik Durumu:

Lise	1997 – 2000	Afyon Süleyman Demirel Fen Lisesi
Lisans	2000 – 2004	H.Ü Mühendislik Fakültesi Elektrik – Elektronik Mühendisliği Bölümü

Yabancı Dil: İngilizce

İş Tecrübesi:

2006 - 2007	Nanomanyetik Bilimsel Cihazlar Ltd. Şti.
2004 – 2005	Z-Sistem Bilgi Teknolojileri Ltd. Şti.