MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ BÖLÜMÜ

EEM363 SAYISAL SİSTEM TASARIMI

Haluk BAYRAM

HAZIRLAYANLAR:

161201012 Gökçe Nur BEKEN
17121201033 Derya KARA

Ocak 2020
İSTANBUL

İÇİNDEKİLER

ŞEKİLLER

FPGA İLE ŞİFRELEME VE ŞİFRE ÇÖZÜMLEME

PROJE TANIMI

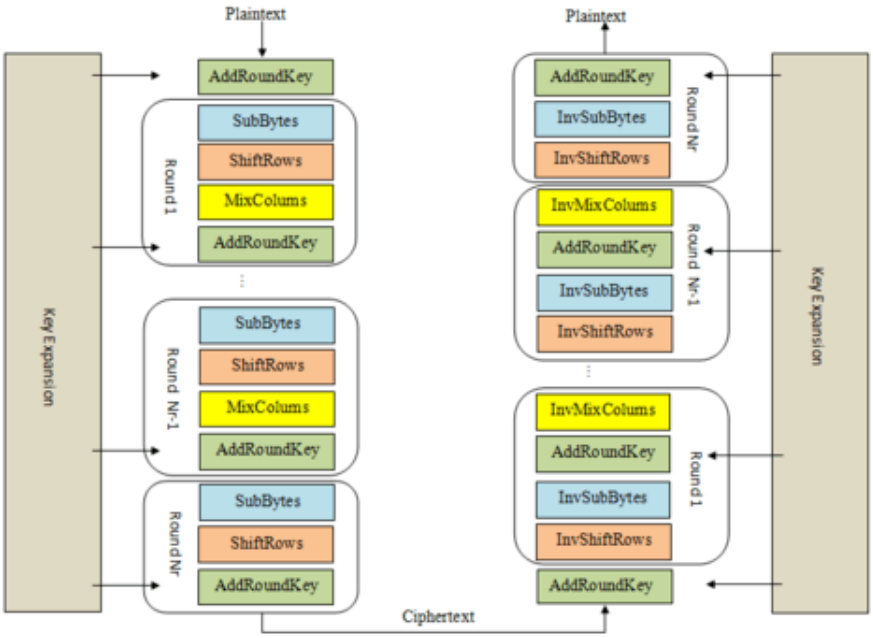
Bu projede düz metnin şifrelenmesi ve şifreli metnin şifresinin çözülmesi FPGA implemantasyonu ile sağlanması amaçlanmaktadır. Bunun için AES (Advanced Encryption Standart) algoritması kullanılmıştır. Kodlamasında VHDL (Very High Speed Integrated Circuit Hardware Description Language) donanım tanımlama dili kullanılmıştır. Kullanılan FPGA kartı, BASYS-3 Artix-7 FPGA Trainer Board'dır.

1. AES (Advanced Encryption Standard)

AES (Advanced Encryption Standard – Gelişmiş Şifreleme Standardı), Kriptografi şifreleme bilimi demektir. Elektronik verinin şifrelenmesi için sunulan bir standarttır. Amerika tarafından kabul edilen AES, uluslararası alanda da defacto şifreleme (kripto) standardı olarak kullanılmaktadır.

1.1. AES Algoritmasının Genel Yapısı

AES simetrik bir şifreleme algoritmasıdır yani hem şifreleme hem şifre çözme işlemleri için aynı anahtar kullanılır. Projede AES-128 modeli kullanılacaktır. Bu model, 128 bitlik bir girdi alır ve 128 bitlik çıktı verir. Bu işlemin işlem hızı, bilgisayarın ve FPGA kartının kapasitesine göre değişmektedir.



***Şekil 1**.AES Algoritması Genel Akış (Encryption: Şifreleme, Decryption: Şifre Çözme)*

AES-128 modelinde şifrelemenin (Encryption) ve şifre çözmenin (Decryption) gerçekleşebilmesi için 10 tur yapılacaktır. Şifrelemede bu 10 turun 9 turu; Bayt Değiştirme (SubBytes), Satır Kaydırma (ShiftRows), Sütun Karıştırma (MixColumns) ve Tur Anahtarı (AddRoundKey) ile toplama gibi adımların tekrar etmesi şeklinde gerçekleşir. Son turunda ise bayt değiştirme, satır kaydırma ve tur anahtarı ile toplama işlemi gerçekleştirilir. Şifre çözümlemede ise tur anahtarı dışındaki bütün işlemler tersi (Inverse) alınarak gerçekleştirilir.

1.2. Encryption (Şifreleme)

Girişten gelen metin 128 bitlik parçalara bölünür. Her parça durum matrisine yerleştirilir. Durum matrisi oluşturulduktan sonra, artık üzerinde tüm işlemler yapılabilir duruma gelmiş demektir. Aynı şekilde önceden alınan 128 bitlik anahtarda bu durum matrisi halinde işlem görür. Giriş metninin yazıldığı durum matrisi ilk olarak anahtar ile toplanır.

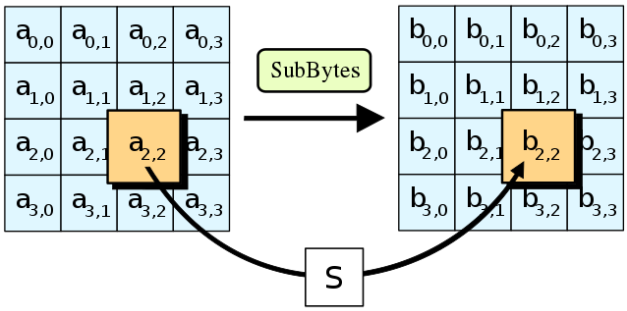
S0	S4	S8	S12
S1	S5	S9	S13
S2	S6	S10	S14
S3	S7	S11	S15

Şekil 2. *Durum Matrisi*

Şekil 2 ile gösterilen durum matrisinde her bir hücre 8 bit yer kaplamaktadır, 16 hücre bulunduğu için toplam 128 bitlik bir veriye karşılık düşer.

1.2.1. Bayt Değiştirme (Substitute Bytes)

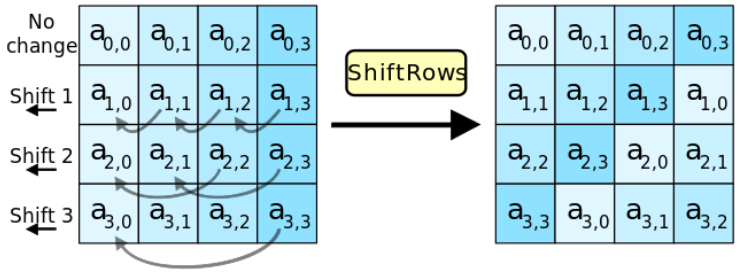
Bayt değiştirme algoritmanın tek doğrusal olmayan işlemidir. Durum matrisinin her elemanı, değerleri önceden hesaplanarak oluşturulmuş S-kutusundaki (S-Box) değerlerle değiştirilir.



Şekil 3. *S-Kutusundaki Değer ile Bayt Değişimi*

1.2.2. Satır Kaydırma (Shift Rows)

Satır kaydırma işleminde satırlar sırasıyla çevrimsel şekilde kaydırılırlar. Yani ilk satır değiştirilmez, ikinci satır da sola 1 ötelenir, üçüncü satır sola 2 ötelenir ve son satır sola 3 ötelenir. Taşan bölmeler kaydırmanın başına eklenir.



Şekil 4. *Satır Kaydırma İşlemi*

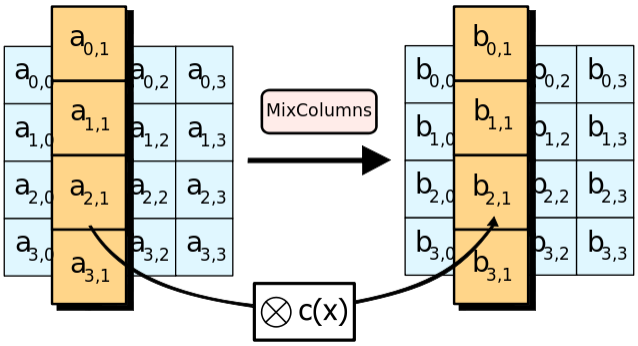
1.2.3. Sütun Karıştırma (Mix Columns)

Bu işlemde eski sütunun elemanları kullanılarak yeni sütun elde edilmektedir. Bu yapılırken yeni sütunun elemanları eski sütunun her elemanı hesaba katılarak tek tek hesaplanır. Yapılan hesap çarpma ve toplama işleminden oluşur. Çarpma işleminde belirli bir sabit sayı (a(x)) kullanılır.

a(x)={03}x3+{01}x2+{01}x+{02} ⟶ s'(x)=a(x)*s(x)

Bu işlem sırasında, her sütun sabit bir matris kullanılarak dönüştürülür (sütunla sola çarpılan matris, durumdaki yeni sütun değerini verir):

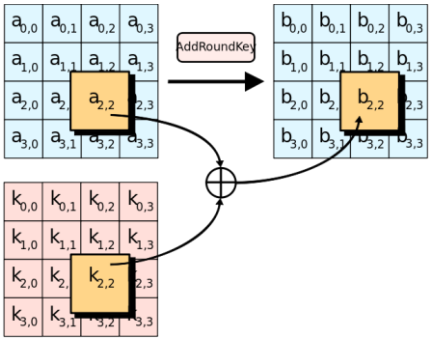
$$\begin{bmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_{0,j} \\ a_{1,j} \\ a_{2,j} \\ a_{3,j} \end{bmatrix} \qquad 0 \leq j \leq 3$$



Şekil 5. *Sütun Karıştırma*

1.2.4. Tur Anahtarıyla Toplama (Add Round Key)

Her turda daha önce saydığımız işlemlerle birlikte bir de tur anahtarı oluşturma işlemi yapılmaktadır ve her turda sonuçta oluşan durum ile o tur için hazırlanmış olan yeni anahtar toplama işlemine tabi tutulur. Bu işlem sonlu alanlarda yapılan toplama işlemidir ve bit mertebesinde özel veya işlemine karşılık düşer. 128 bitlik durum matrisi ile 128 bitlik ara anahtar değeri bit bit özel veya elemanı ile toplanır.



Şekil 6. *Tur Anahtarı ile Toplama İşlemi*

1.2.5. Anahtar Üretme (Key Generation)

AES algoritması anahtarı alır ve bir dizi işlemden geçirerek işlem sayısı kadar anahtar oluşturur. Bu sayı 128 bitlik uzunluk için 10'dur. 10 farklı anahtar oluşturulur ve oluşan son anahtar şifreyi çözmede kullanılan ilk anahtar haline gelir. Çözerken de aynı işlemler benzer olarak tersten yürütülerek kullanılır. Anahtar üretmede her yeni oluşturulan yeni anahtar kendinden önceki anahtarlar kullanılarak elde edilir.

1.3. Şifreli Metni Çözme (Decryption)

AES algoritmasında oluşturulan şifreli metin kolaylıkla, ters işlemlerle tekrar çözülerek giriş metni elde edilebilmektedir. Tur anahtarı dışında bütün işlemler tersinir olarak yapılır.

1.3.1. Ters Satır Kaydırma İşlemi (Inverse Shift Rows)

Şifreleme yaparken kullanılan satır kaydırma işleminin sola değil de sağa kaydırarak yapılmasıdır.

1.3.2. Ters S-Kutusundan Geçirme İşlemi (Inverse Substitute Bytes)

Bu kez de değişim uygulanan S-kutusundaki bir değişiklik yapılmaktadır. Eski değerlere geri dönebilmek için S-kutusunu oluşturan değerler farklı bir metodla hesaplanır. Sonuç olarak oluşan kutu; normal S-kutusunda elde ettiğimiz değeri tekrar girişine uyguladığımızda bize ilk değeri geri verecek şekilde düzenlenmiş halindedir.

1.3.3. Ters Sütun Karıştırma İşlemi (Inverse Mix Columns)

Bu işlemde sütunlar şifreleme sırasındaki sütun karıştırma işleminden farklı bir sabit polinomla çarpılır.

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{2,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{2,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix}$$

1.3.4. Tur Anahtarı ile Toplama (Add Round Key)

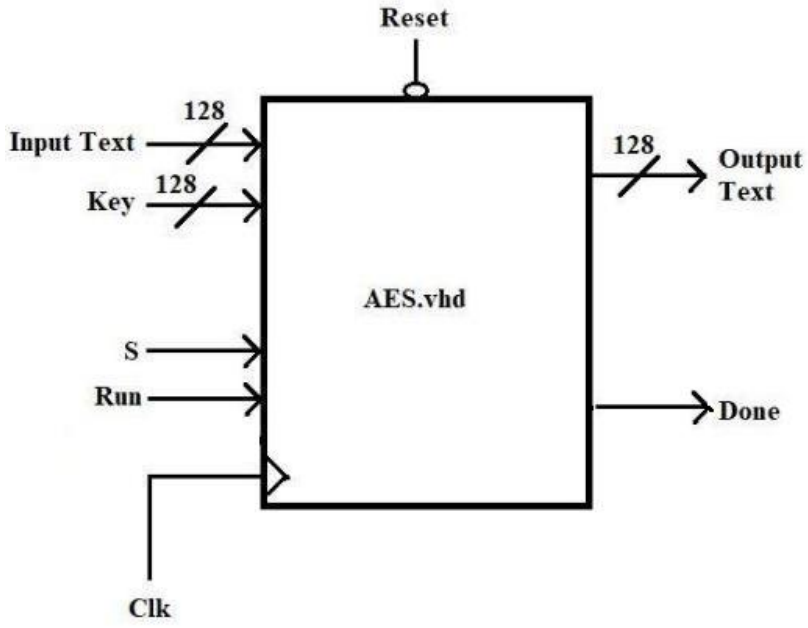
Bit bit özel veya işleminden ibaret olması sebebi ile tekrar özel veya işlemine tabi tutulması eski haline dönmesine sebep olmaktadır. Sonuç olarak şifreleme işlemindeki tur anahtarı ile toplama işleminin aynısı yürütülür.

1.3.5. Ters Anahtar Üretimi (Inverse Key Generator)

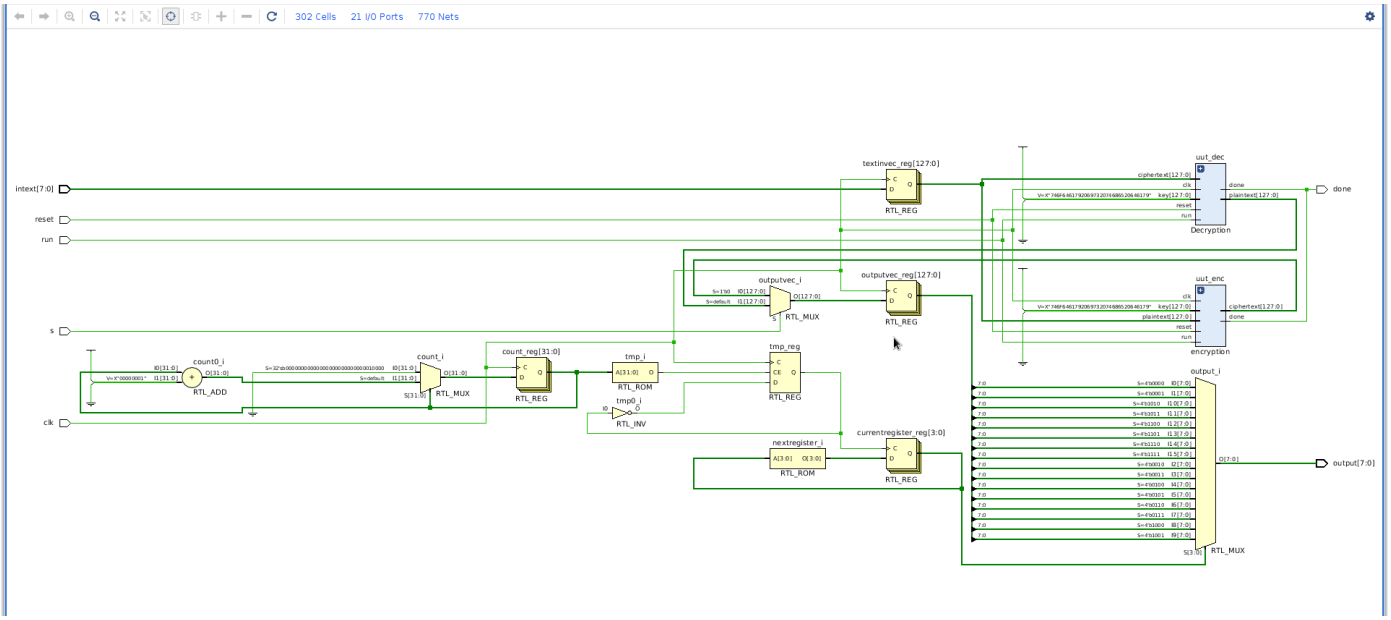
Şifreleme işleminde oluşan son anahtar, çözme işleminin ilk anahtarı olarak kullanılır. Anahtar üretiminin tersi işlemler ile ters anahtar üretimi gerçekleştirilir.

SİSTEM TASARIMI

Üzerinde işlem yapılacak girdi 128 bitlik (4 kelime uzunluğunda) metindir. Bunu şifrelemek ya da şifresini çözmek için 128 bitlik bir anahtar da sisteme girdi olarak verilir. Bu anahtar her turda değişecektir. Bir diğer girdi olan S, kullanıcının girdi olarak verilen metin üzerinde hangi işlemin (şifreleme ya da şifre çözme) yapılacağını seçmesini sağlar. Run, sistemin çalışmasını kontrol eden girdidir. Done işlemin sonlandığını ya da devam ettiğini gösteren çıktıdır. Sistemin çalışmasını tetikleyen bir clock sinyali ve istenildiğinde sistemi sıfırlayan reset sinyali de bulunmaktadır. Ayrıca sistem clock'un yükselen kenarında çalışmaktadır. Sistemin çıktısı 128 bitlik işlenmiş metindir.



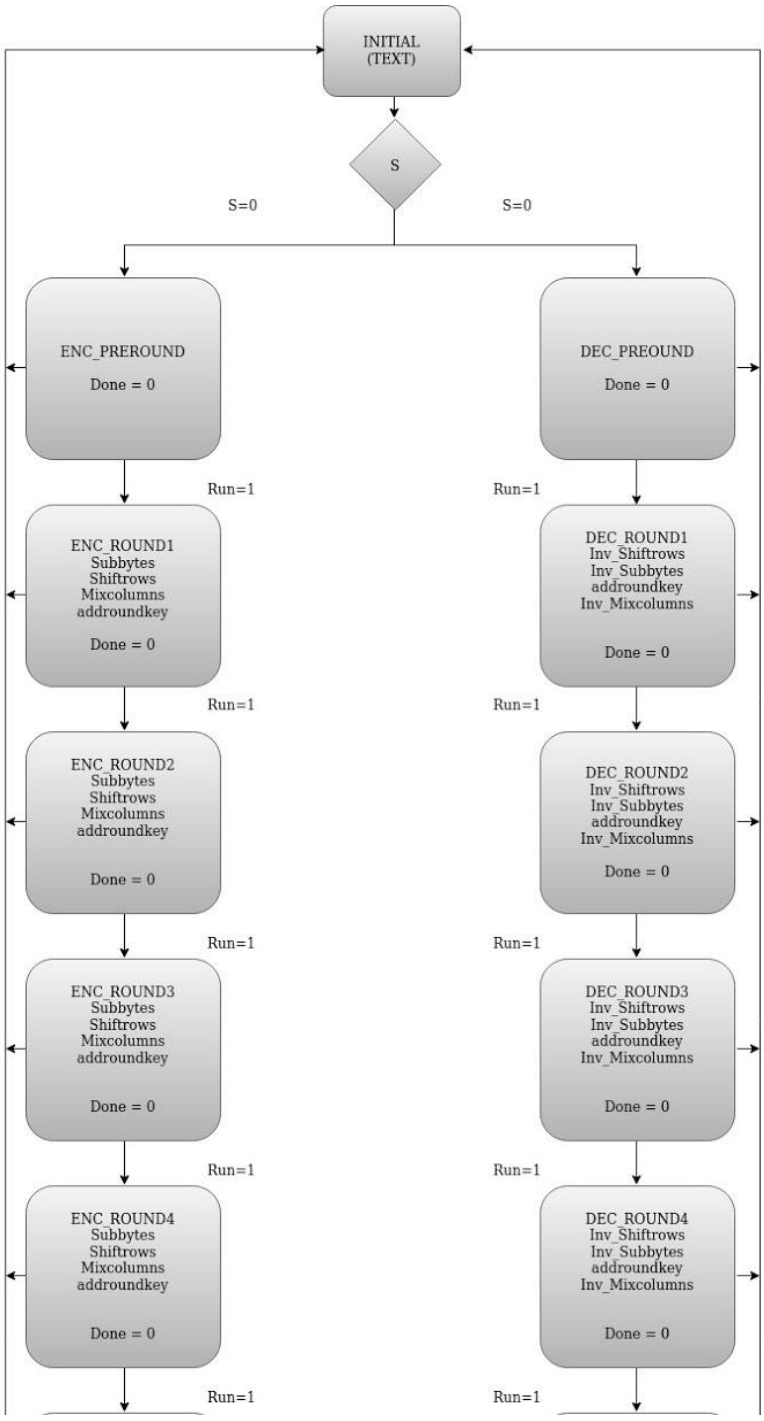
Şekil 7. *AES.vhd Sisteminin Genel Görünümü*

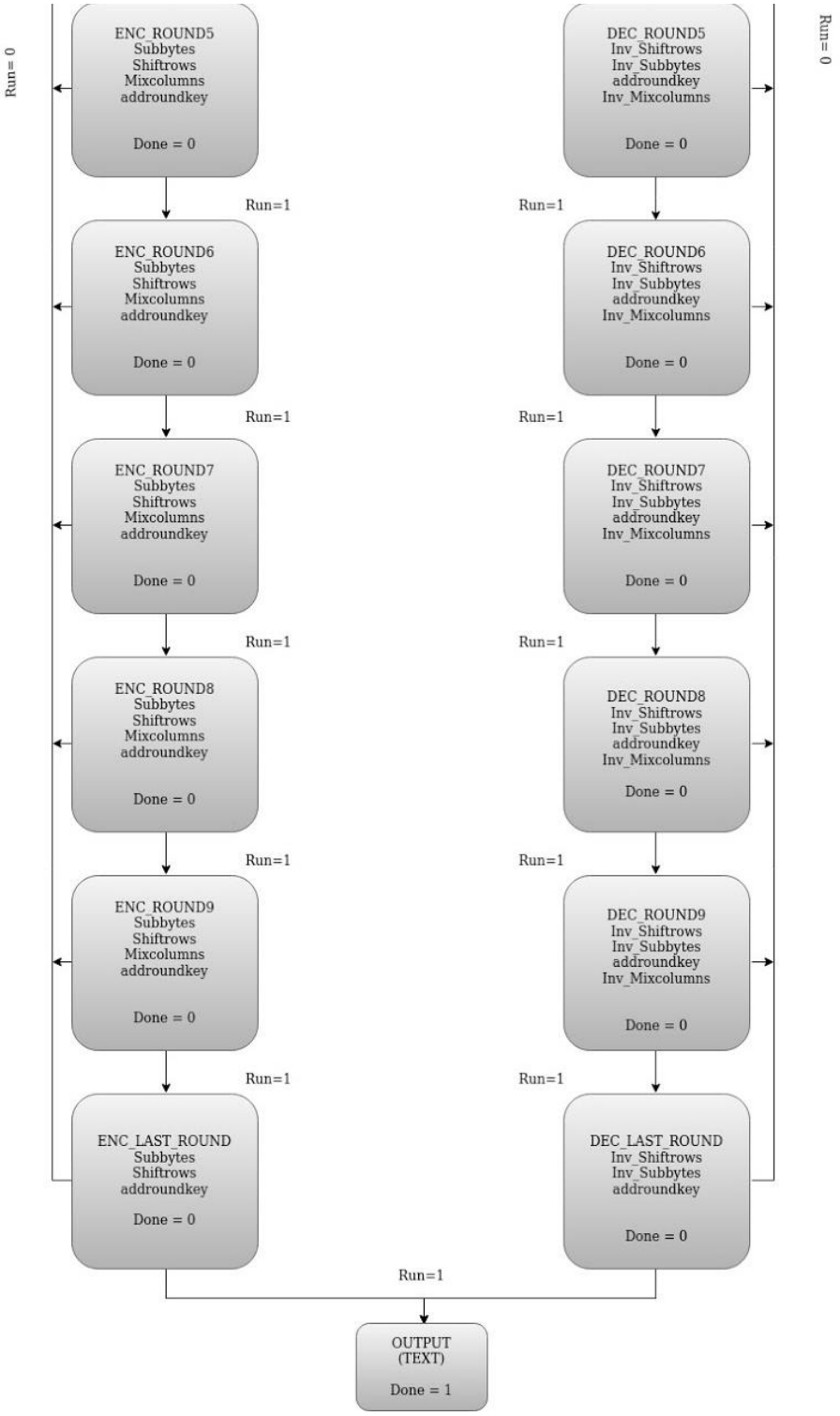


Şekil 8. *AES.vhd Sisteminin RTL Şematiği*

Şifreleme modülleri olan tur anahtarı ekleme, bayt değiştirme, sütun karıştırma ve satır kaydırma ve anahtar oluşturma işlemleri özel algoritmalar olduğu ve üzerinde pek bir değiştirme yapılamayacağı için kod hazır olarak kullanılmıştır. Onun dışında FPGA kartına implementasyonu, şifreleme ana sistemi ve ana sistemin tasarımı, iç bileşenlerin tasarımı ve şifre çözme algoritmasının tasarımı ve kodlaması tarafımızca yapılmıştır. Sistemin girdi ve çıktı boyutları çok büyük olduğu için başta karta implemente edilememiştir. Çünkü kullanılan kartta 16 switch ve 16 led bulunmaktaydı. Bu yüzden anahtar kodun içinde sabit bir şekilde verilmiştir. Girdi ve çıktı 8 bit olarak belirlenmiştir. Girdide girilen her bir bit, sistemin asıl girdisindeki 16 bite karşılık gelmektedir. Tam olarak bütün durumlar gösterilmese de çoğu durum bu şekilde gözlenebilir. Çıktıda böyle bir şey yapılamayacağı için gecikmeli bir şekilde 8 bit, 8 bit şeklinde gösterilmesi tercih edilmiştir. Burada “shift register” mantığı kullanılmıştır.

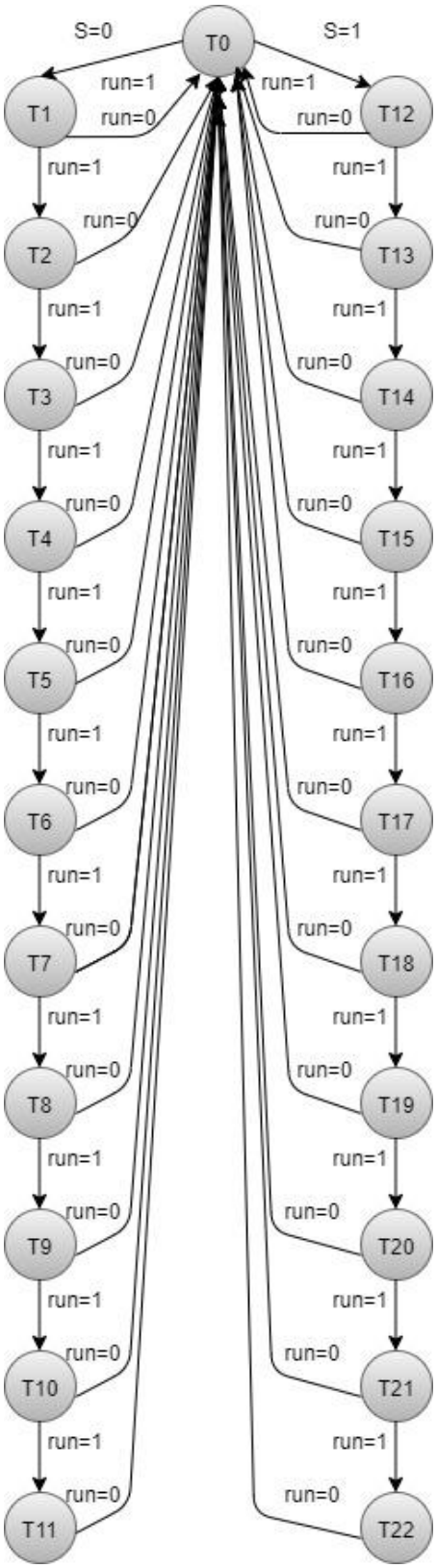
1. Akış Diyagramı





Şekil 9. *Akış Diyagramı*

2. State Diyagramı

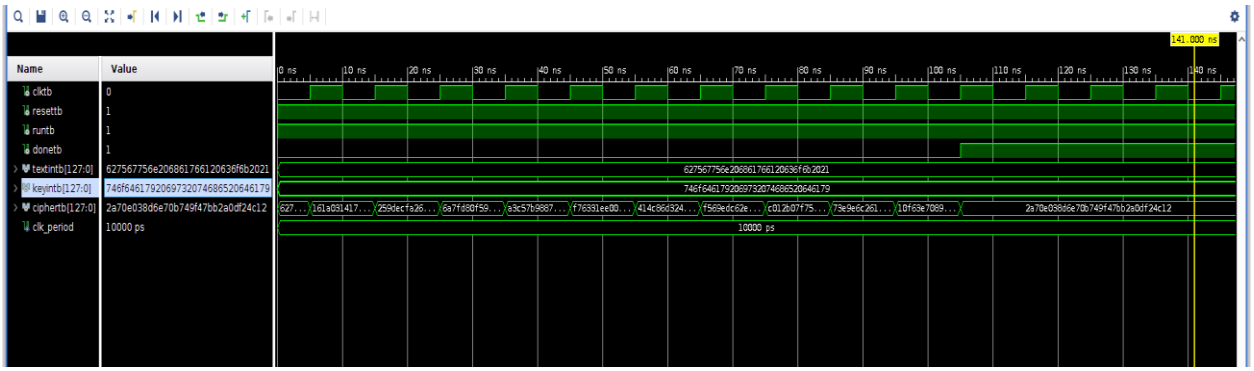


Şekil 10. *State Diyagramı*

SONUÇ

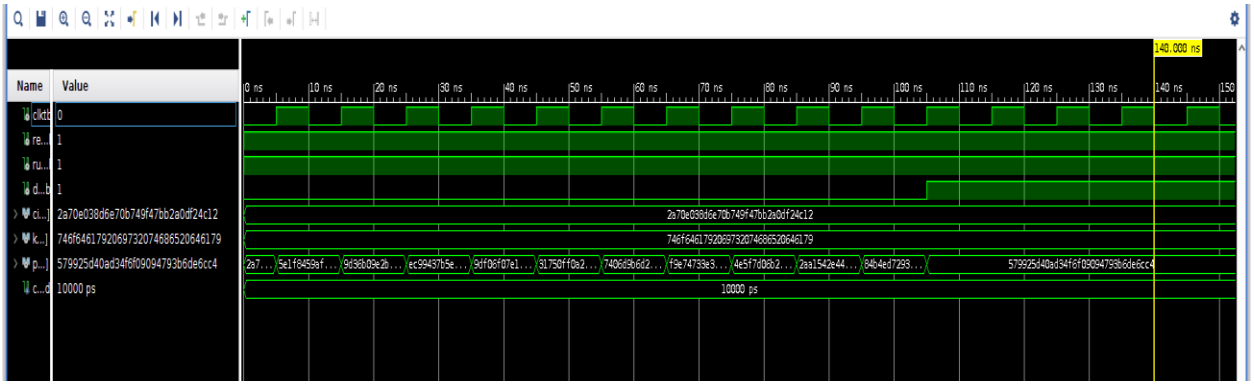
İstenilen sistem oluşturulmuştur. Fakat şifreleme kodu tam doğru bir şekilde çalışırken şifre çözümleme kodu ise çalışmakta ancak istenilen sonuç elde edilememektedir.

Şifreleme kodunun simülasyonu aşağıdaki gibidir.



Şekil 11. *Encryption (Şifreleme)*

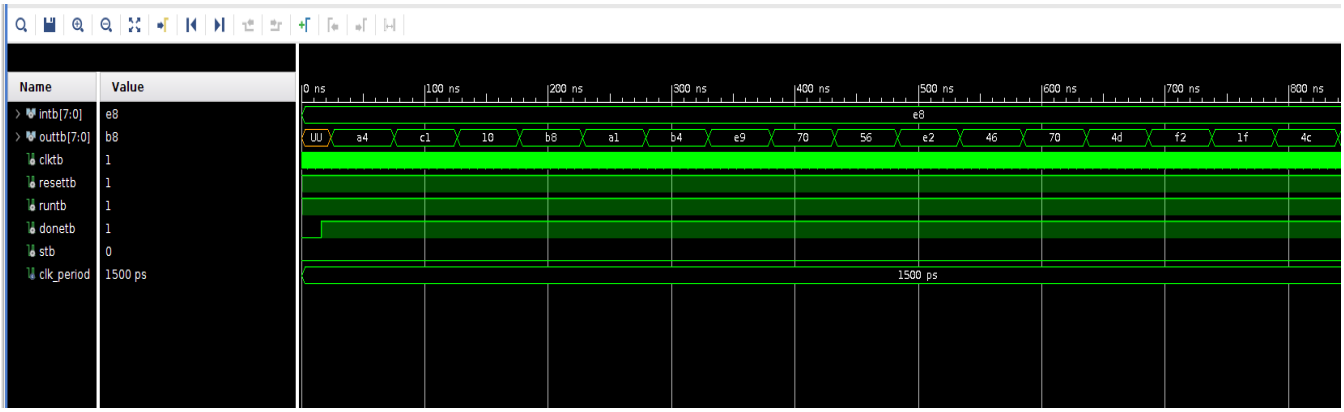
Şifreleme kodunun çıktısı olan şifrelenmiş metin, şifre çözme koduna girdi olarak verildiğinde bir çıktı elde edilmektedir. Fakat bu çözümlenmiş metin değildir. Simülasyonu aşağıda verilmiştir.



Şekil 12. *Decryption (Şifre Çözme)*

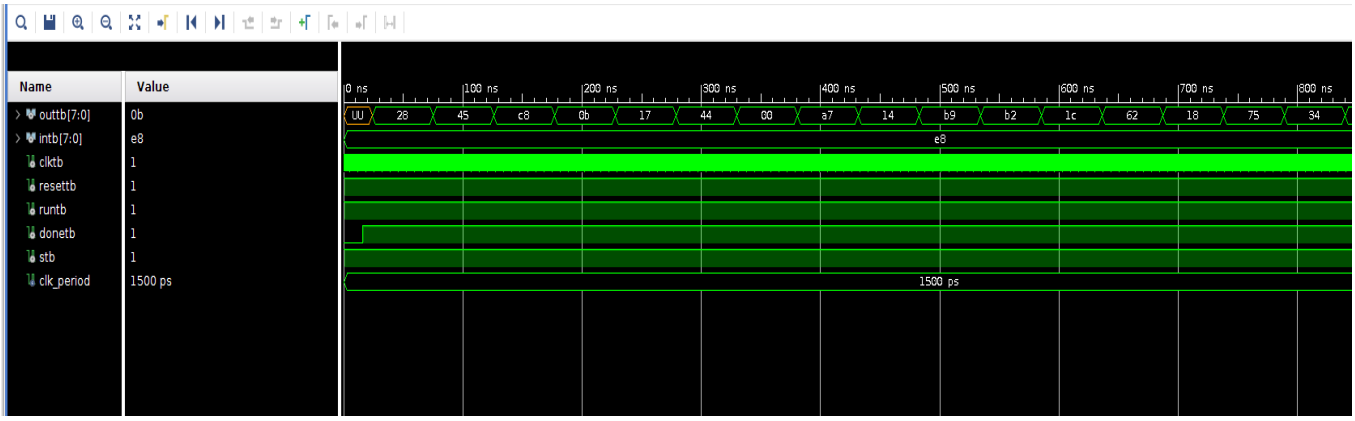
Öte yandan şifreleme ve şifre çözme kodunun birleştirildiği AES.vhd kodunda istenildiğinde şifreleme ve şifre çözme kodu çalışmaktadır.

S = 0 olduğunda şifreleme,



Şekil 13. *AES.vhd Kodunda Şifreleme*

S = 1 olduğunda şifre çözme kodu çalıştırılmaktadır.



Şekil 14. *AES.vhd Kodunda Şifre Çözme*

Şifreleme kısmı karta atıldığında istenilen sonuç elde edilmiştir. Fakat şifreleme ve şifre çözme birleştirildiğinde karta atılabildiği halde sonuç göstermemiştir. Reset = 0 durumunda çıktı vermemektedir. Reset = 1 ve run = 1 olduğunda done = 1 olmakta yani şifreleme veya şifre çözme tamamlanmakta, reset = 1 run = 0 olduğunda done = 0 olmaktadır yani seçilen işlem tamamlanmaktadır. Bu ise, çıktı metninin gösterilmesi dışında her şeyin doğru çalıştığını göstermektedir.

REFERANSLAR

https://github.com/pnvamshi/Hardware-Implementation-of-AES-VHDL

https://sites.math.washington.edu/~morrow/336_12/papers/juan.pdf

https://www.researchgate.net/publication/317615794_Advanced_Encryption_Standard_AES_Algorithm_to_Encrypt_and_Decrypt_Data

http://bilgisayarkavramlari.sadievrenseker.com/2009/06/03/aes-ve-rijndeal-sifreleme/