İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DES BLOK ŞİFRELEME ALGORİTMASININ FPGA ÜZERİNDE DÜŞÜK ENERJİLİ TASARIMI

YÜKSEK LİSANS TEZİ
Tarık KAPLAN

Anabilim Dalı : Elektronik ve Haberleşme Mühendisliği

Programı : Elektronik Mühendisliği

HAZİRAN 2009

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

DES BLOK ŞİFRELEME ALGORİTMASININ FPGA ÜZERİNDE DÜŞÜK ENERJİLİ TASARIMI

YÜKSEK LİSANS TEZİ

Tarık KAPLAN
(504061224)

Tezin Enstitüye Verildiği Tarih : 04 Mayıs 2009
Tezin Savunulduğu Tarih : 05 Haziran 2009

Tez Danışmanı : Yrd. Doç. Dr. Sıddıka Berna ÖRS YALÇIN (İTÜ)
Diğer Jüri Üyeleri : Prof. Dr. Ece Olcay GÜNEŞ (İTÜ)
Prof. Dr. Ali Emre HARMANCI (İTÜ)

HAZİRAN 2009

ÖNSÖZ

Yüksek lisans tezimin ortaya çıkmasında önemli katkıları bulunan, bana her zaman yol gösteren ve beni teşvik eden danışman hocam Yrd. Doç. Dr. S. Berna ÖRS YALÇIN' a çok teşekkür ederim.

Ayrıca yardımlarını benden esirgemeyen arkadaşım Ali Can ATICI' ya ve beni hayatım boyunca her yönden destekleyen ve hiçbir zaman yalnız bırakmayan aileme şükranlarımı sunarım.

Son olarak, tez çalışmalarım sırasında maddi destekte bulunan TÜBİTAK'a teşekkür ederim.

Haziran 2009

Tarık KAPLAN

Elektronik Mühendisi

İÇİNDEKİLER

Sayfa

KISALTMALAR

AES : Advanced Encryption Standard
ASIC : Application Specific Integrated Circuit
CBC : Cipher Block Chaining
CFB : Cipher Feedback
CLB : Configurable Logic Blocks
DES : Data Encryption Standard
ECB : Electronic Code Book
EEPROM : Electrically Erasable Programmable Read Only Memory
EPROM : Erasable Programmable Read Only Memory
FIPS : Federal Information Processing Standards
FPGA : Field Programmable Gate Array
HDL : Hardware Description Language
IP : Initial Permutation
LUT : Look-Up Table
NIST : National Institute of Standards and Technology
OFB : Output Feedback
PLD : Programmable Logic Devices
RAM : Random Access Memory
ROM : Read-Only Memory
SRAM : Static Random Access Memory
VHDL : Very High Speed Integrated Circuit HDL

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

DES BLOK ŞİFRELEME ALGORİTMASININ FPGA ÜZERİNDE DÜŞÜK ENERJİLİ TASARIMI

ÖZET

Günümüz teknoloji dünyasında veri aktarımı önemli konulardan biridir. Özellikle veri aktarımının güvenli bir şekilde yapılması, üzerinde en çok durulan konuların başında gelir. Veri aktarımının güvenli bir şekilde yapılmasında şifreleme algoritmalarından yararlanılmaktadır. İlk zamanlarda şifreleme algoritmalarının tasarımında yüksek hız ve yüksek işlem hacmi hedeflenmekteyken, son yıllarda enerji ve alan kısıtlamasına sahip ortamlarda kullanım arttıkça enerji tasarruflu tasarımlar da büyük önem kazanmıştır. Düşük enerji tüketen devre tasarlayabilmek için var olan şifreleme algoritmaları üzerinde çalışmalar devam etmektedir.

Bu çalışmada en çok bilinen blok şifreleme algoritmalarından biri olan Veri Kodlama Standardı (Data Encryption Standard (DES)) algoritması sahada programlanabilir kapı dizileri (Field Programmable Gate Array (FPGA)) üzerinde gerçeklenmiş ve enerji tasarrufu yöntemlerinden yararlanılarak tasarlanan farklı DES yapıları sonucunda literatürdeki en düşük enerji tüketimine sahip devre elde edilmiştir.

DES blok şifreleme algoritmasının FPGA üzerinde düşük enerjili tasarımı yüksek lisans tezinde öncelikle çalışmanın amacından ve FPGA' lardan bahsedilmiştir. Daha sonra DES blok şifreleme algoritması ve gerçeklenmesi ile enerji tasarrufu yöntemleri anlatılmıştır. Son olarak, DES blok şifreleme algoritmasının düşük enerjili tasarımı geniş bir biçimde anlatılmış ve elde edilen sonuçlar yorumlanmıştır. Bu çalışmada, güç tasarrufu yöntemlerinden ve işlem hacmini arttırma yöntemlerinden yararlanılarak, altı farklı DES yapısı gerçeklenmiştir. Enerji tasarrufu yöntemlerinden yararlanılarak işhattı yapısında, yazmaçsız yani kombinezonsal yapıda, klasik yapıda, sekiz turda iki S-kutulu yapıda, sekiz turda tek S-kutulu yapıda ve iç yazmaçlı işhattı yapısında DES devreleri gerçeklenmiştir. Gerçeklenen bu farklı yapılar alan, zaman, işlem hacmi, güç ve enerji tüketimleri açılarından kapsamlı olarak incelenmiş ve karşılaştırılmıştır. Ayrıca literatürdeki diğer DES gerçeklemeleri de incelenerek, en düşük enerji tüketen DES devresi elde edilmiştir. Bu çalışmadaki tüm tasarımlar sırasında FPGA teknolojisi kullanılmıştır.

Literatürde, DES gerçeklemelerinde genellikle yüksek hız hedef olarak alınmıştır. Düşük güç veya düşük enerji hedefiyle tasarım yok denecek kadar azdır. Bu çalışma ile hem literatürdeki DES gerçeklemeleri işlem hacmi, güç ve enerji açılarından karşılaştırılmış hem de enerji tüketimi en düşük devre gerçeklemesi sunulmuştur. Ayrıca düşük güç tüketen devre tasarımı için gerekli yapısal değişiklikler ve yüksek hızlı devre için gerekli yapısal değişiklikler üzerinde durulmuştur. Bununla birlikte, DES algoritmasını gerçeklemek isteyen tasarımcılar için kapsamlı bir kaynak hazırlanmaya çalışılmıştır.

ENERGY EFFICIENT FPGA IMPLEMENTATION OF DES ALGORITHM

SUMMARY

In today's technology world, data transfer is one of the important issues. Especially the security of transferred data is one of the most popular issues. Cryptographic algorithms help to provide a secure data transfer. In early times, the aims of the cryptographic circuit designs were high speed and high throughput. In recent years, energy efficient designs are becoming more important due to the increase in usage of cryptographic algorithms in energy and area-limited platforms. Several researches are on going about the design of energy efficient implementations of cryptographic algorithms.

In this thesis, Data Encryption Standard (DES), this is one of the most popular block ciphers, is implemented on a Field Programmable Gate Array (FPGA). By using low energy design methods, six different architectures of DES algorithm are implemented and the DES design with the lowest energy consumption in the literature is obtained.

In energy efficient FPGA implementation of DES algorithm thesis, firstly the aim of the thesis and FPGA structures are explained. Then, the DES block cipher algorithm and its implementation are introduced. In addition, low energy methods are defined. And then, energy efficient implementation of DES block cipher algorithm is explained deeply and the results of implementations are compared. In this thesis, by using low power and high throughput methods, six different DES architectures are implemented. By using low energy methods, DES designs in pipeline architecture, combinational architecture, classical architecture, eight rounds with two S-boxes and one S-box architectures and inner-outer pipeline architecture are implemented. These implemeneted different architectures are compared from area, speed, throughput, power and energy consumption perspectives. FPGA technology is used for these implementations. In addition, other DES implementations in the literature are searched and the DES implementation with the lowest energy consumption is obtained.

In the literature, the aim of the DES implementations is usually the high speed. Low power or low energy aimed designs are very few. In this thesis, DES implementations in the literature are compared from throughput, power and energy consumption perspectives and also the implementation with the lowest energy consumption is added to the literature. In addition, low power design techniques and high throughput design techniques are explained. In brief, a very detailed resource is tried to be prepared for the designers who want to implement the DES algorithm.

1. GİRİŞ

Günümüz teknoloji dünyasında veri aktarımı önemli konulardan biridir. Özellikle veri aktarımının güvenli bir şekilde yapılması, üzerinde en çok durulan konuların başında gelir. Gün geçtikçe çok büyük miktarlardaki verilerin bilgisayarlar tarafından işlenmesi, saklanması ve elektronik haberleşme kanalları üzerinden bir yerden diğer bir yere iletilmesi gündelik hayatın sıradan işlerinden biri haline gelmektedir. Ancak verilerin iletimi sırasında kullanılan haberleşme kanallarının herkesin kullanımına ya da erişimine açık olması, sözkonusu verilerin yetkili olmayan (üçüncü) şahıslar tarafından değiştirilmesi, yok edilmesi ve içeriğine ulaşılması problemini gündeme getirmektedir. Bu noktada, mesajların herkesin erişimine açık elektronik haberleşme kanallarından iletilebilmesi için bir takım dönüşümler sonucunda değişikliğe uğratılarak üçüncü şahıslar için anlaşılamaz bir hale getirilmesi gerekmektedir. Bu amaçla yapılan tüm işlemlere birden Kriptografi ya da Şifreleme adı verilir [1]. Diğer bir tanımla Kriptografi, en az iki kişinin güvenli olmayan bir kanal üzerinden üçüncü bir şahsa bilgi sızdırmadan haberleşmesini sağlamak amacıyla matematiksel tekniklerin geliştirilmesidir [2].

Kriptografi, bilgiyi yani veriyi güvenli bir şekilde sadece istenilen kişiye ulaştırmak amacıyla uzun süredir kullanılmaktadır. İlk başlarda askeri amaçlı kullanıldıysa da gelişen teknoloji ile birlikte ortaya çıkan güvenlik açığını kapatmak ve bilginin güvenilir bir şekilde taşınmasını sağlamak amacıyla sivil yaşamda da yerini almıştır. Veri transferi sırasında güvenilirliği sağlamak için sıkça kriptografik algoritmalar kullanılır. Kullanılan bu algoritmaları donanımsal veya yazılımsal olarak gerçeklemek mümkündür. Yazılımsal gerçeklemeler daha az maliyet getirmekle birlikte yavaş ve güvensizdirler. Donanımsal gerçeklemelerin ise istenilen yüksek hızlarda çalışma imkanları vardır, ayrıca yazılımsal gerçeklemelere oranla daha güvenilir gerçeklemelerdir [3].

Pek çok kripto algoritması yüksek hız ve yüksek işlem hacmine sahip olacak şekilde Sahada Programlanabilir Kapı Dizileri (FPGA) ve Uygulamaya Özel Tümdevre (ASIC) teknolojileri kullanılarak gerçeklenmektedirler. Tasarımlarda belirlenen hedeflere göre algoritmaların gerçeklenmesi farklı şekillerde olabilmektedir. Kimi tasarımlar hızı ön plana koyarken, kimi tasarımlar alandan tasarrufu hedeflemektedir. Bunların dışında, güvenliği, düşük güç tüketimini veya düşük enerji harcamayı hedefleyen devre tasarımları olabilmektedir. Doğal olarak her tasarımcının hedefi, gerçeklediği algoritmayı en iyi şekilde yapmaktır. Ancak bahsedilen hedefleri çoğu zaman aynı anda yakalamak mümkün değildir. Bu sebeple Kripto algoritmalarının tasarımı belirlenen hedefe yönelik çalışılarak gerçeklenmektedir.

Düşük enerji hedeflenen bir Kripto algoritması tasarımında, dikkat edilmesi gereken konular güç ve zamandır. Devrelerin harcadığı enerji, tükettiği güç ve şifreleme için kullanılan süreyle doğru orantılıdır. Bir elektronik devrede temel olarak harcanan iki tür güç kavramından bahsedilebilir. Bunlardan ilki statik güç olarak adlandırılır, devreye herhangi bir giriş uygulanmadan, devrenin sükunet halinde harcadığı güçtür. Devre üzerinde harcanan diğer güç ise dinamik güçtür. Dinamik güç devreye verilen girişlerin işlenmesi sırasında harcanan güçtür. Bir devreye ait statik güç harcaması zamanla değişmezken, dinamik güç harcaması uygulanan girişlere bağlı olduğundan zamanla değişmektedir. Enerjiyi etkileyen diğer faktör olan şifreleme zamanı da devrenin çalışma frekansına ve algoritmanın gerçeklenme şekline bağlı olarak değişim göstermektedir. Düşük enerji tüketen devre elde edebilmek için hem düşük güç tüketimi hem de yüksek işlem hacmi gerekmektedir.

1.1 Tezin Kapsamı

Veri Kodlama Standardı (Data Encryption Standart (DES)), 1977 yılında Amerikan Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology (NIST)) tarafından standart olarak kabul edilmiş ve Federal Bilgi İşleme Standardı (Federal Information Processing Standards (FIPS)) olarak yayınlanmıştır [4]. DES algoritması uzun yıllar standart olarak kullanılmış, ancak 1999 yılında yerini Gelişmiş Kodlama Standardı (Advanced Encryption Standard (AES))' e bırakmıştır. DES algoritması, bilgiyi şifrelemek ve çözmek için kullanılan simetrik bir blok şifreleyicidir.

DES blok şifreleme algoritması, en çok bilinen blok şifreleme algoritmalarından biridir. Günümüzde çok yüksek güvenlik gerektirmeyen alanlarda halen kullanılmaktadır. Kablolu modemler, uzaktan erişim sunucuları, şifreli veri depolama aygıtları örnek kullanım alanları olarak verilebilir [5]. Literatürde pek çok yazılımsal ve donanımsal DES gerçeklemesi mevcuttur. Gelişen teknoloji ile birlikte kaynak kısıtlı ortamlarda da bilgiyi şifreleme gereksinimi duyulmaktadır. Bu ortamlarda enerji tasarruflu DES gerçeklemelerine ihtiyaç duyulmaktadır.

Bu çalışmada, DES blok şifreleme algoritmasının düşük enerjili tasarımı hedeflenmiştir. Güç tasarrufu yöntemlerinden ve işlem hacmini arttırma yöntemlerinden yararlanılarak, farklı DES yapıları gerçeklenmiştir. Literatürdeki diğer DES gerçeklemeleri de incelenerek, en düşük enerji tüketen DES devresi gerçeklenmiştir. Tüm bu tasarımlar sırasında Sahada Programlanabilir Kapı Dizileri (FPGA - Field Programmable Gate Array) teknolojisi kullanılmıştır.

1.2 Tezin Konuya Katkısı

Literatürde, DES gerçeklemelerinde genellikle yüksek hız hedef olarak alınmıştır. Düşük güç veya düşük enerji hedefiyle tasarım yok denecek kadar azdır. Bu çalışma ile hem literatürdeki DES gerçeklemeleri alan, zaman, iş hacmi, güç ve enerji açısından karşılaştırılmış hem de enerji tüketimi en düşük devre gerçeklemesi sunulmuştur. Ayrıca düşük güç tüketen devre tasarımı için gerekli yapısal değişiklikler ve yüksek hızlı devre için gerekli yapısal değişiklikler üzerinde durulmuştur. Bununla birlikte, DES algoritmasını gerçeklemek isteyen tasarımcılar için kapsamlı bir kaynak hazırlanmaya çalışılmıştır.

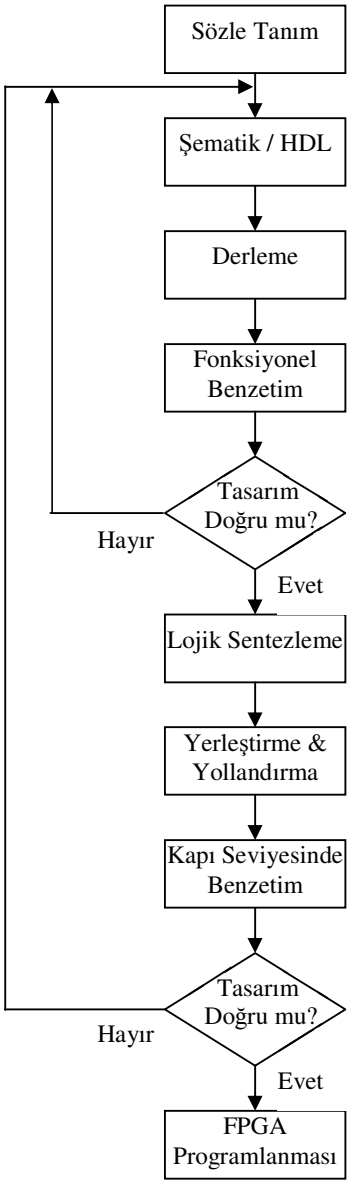
2. SAHADA PROGRAMLANABİLİR KAPI DİZİLERİ

Sahada Programlanabilir Kapı Dizileri (Field Programmable Gate Array, FPGA) yaygın olarak kullanılan programlanabilir devre elemanlarıdır. Programlanabilir devre elemanları, geniş uygulama alanları sağlayabilmek için genel amaçlı tümdevreler olarak tasarlanmışlardır. Programlanabilir eleman ve arabirimler VE (AND), VEYA (OR), ayrıcalıklı VEYA (E-XOR), DEĞİL (NOT) işlemlerini veya daha karmaşık olan dekoder, çoklayıcı gibi matematiksel işlemleri gerçekleştirmek amacıyla programlanabilir. FPGA yapısında bulunan arabirimler tasarımdaki bağlantılar göz önüne alınarak elektriksel olarak programlanabilir ve istenildiği kadar programlanabilme yapısına sahip olduğundan tasarımlarda büyük kolaylık sağlar. Pek çok FPGA yapısı programlanabilir eleman ve ara birimlere ek olarak hafıza birimleri de bulundurur. Bu hafıza birimleri ayrık flip-flop yapılarından veya hafıza bloklarından oluşabilir [3].

Programlanabilir yapılar, örneğin FPGA'lar ASIC tasarımlara oranla daha yavaş çalışmalarına rağmen tekrar programlanabilme özelliği ve tasarımların basit kontrol edilebilir olması nedeniyle tasarımın daha ucuza mal edilebilmesi açısından önemli avantajlara sahiptir.

FPGA' ların programlanması aşamasında ilk olarak tasarlanacak devrenin sözle tanımı verilir. Daha sonra şematik olarak veya yüksek seviyeli donanım tanımlama dilleri (Hardware Description Language, HDL) kullanılarak tasarım yapılır. Tasarım her ne şekilde olursa olsun derleme işleminden sonra devreye ait standart bağlantı listesi (netlist) oluşturulur. Yapılan tasarımın devreye ait istenen özellikleri yerine getirip getirmediği fonksiyonel benzetim (functional simulation) yapılarak test edilir. Benzetim sonucuna göre gerekirse, tasarımda değişiklikler yapılarak istenen sonuç elde edilene kadar bu şekilde iterasyona devam edilir. İstenilen sonuç elde edildikten sonra lojik sentezleme adımına geçilir. Bu aşamada üretilecek devre gerçeklenirken kullanılacak FPGA seçilir. Buna göre lojik sentezleyicinin, kullanılacak FPGA' yı desteklemesi gerekmektedir. Sentezleme işleminde ayrıca varsa, devreye ait lojik kısıtlamalar (Giriş / Çıkış bacakları, zamanlama, yerleştirme, saat frekansı, kritik

yollar gibi) kullanıcı kısıtlama dosyası (user constraints file, Xilinx) ile birlikte verilebilir. Lojik sentezleyiciler, istenilen fonksiyonların en iyi şekilde gerçeklenebilmesi için gerekli lojik indirgemeleri (logic optimization) de yaptıktan sonra elde edilen lojik fonksiyonların FPGA içerisindeki lojik bloklarla eşleştirilmesi işlemi yapılarak (technology mapping) kapı seviyesinde bir bağlantı listesi oluşturulur. Teknoloji eşleştirilmesi sırasında, kullanıcı kısıtlama dosyası da kullanılarak zamanlama gereksinimi karşılanmak amacıyla gerekirse daha fazla lojik eleman kullanılabilir. Sentezleme sonrasında, yerleştirme ve yollandırma (placement and routing) işlemleri yapılır. Bu adımda, devre fonksiyonları ile eşleştirilmiş lojik bloklar FPGA içerisinde uygun yerlere yerleştirilir ve bu bloklar arasındaki bağlantılar oluşturulur. Lojik yolların daha kısa olması amacıyla birbirleriyle ilişkili CLB (Configurable Logic Blocks)' ler yakın yerleştirilir. Yollandırmada ise bağlantılar uygun şekilde seçilir. Örneğin, tasarımın bir çok alanında bir işarete ihtiyaç varsa en küçük gecikmeyi sağlamak amacıyla uzun bir yol kullanılır. Bu aşamadan sonra kapı seviyesinde benzetimin gerçekleştirilmesi uygun olacaktır. Çünkü artık bütün CLB' lere (LUT veya Çoğullayıcılar ve flip-flop' lara) ve yollandırma bağlantılarına ait gecikmeler gerçeğe çok yakın olarak elde edilmiştir. Bu gecikmeler de eklenerek devrenin benzetimi yapıldığında, zamanlama ve hız açısından kritik yollar saptanabilir. Yerleştirme ve yollandırma sonrası benzetimlerde istenilen sonuçlar elde edildikten sonra FPGA' nın programlanması aşamasında kullanılacak bit dizisi, üreticinin sağladığı yazılımla elde edilir. FPGA' nın uygun donanım kullanılarak programlanmasıyla tasarım tamamlanır. Şekil 2.1'de yüksek seviyeli tasarım sürecine ilişkin akış diyagramı verilmiştir [2].



Şekil 2.1 : Tasarım sürecine ilişkin akış diyagramı.

2.1 FPGA' ların Güç Tüketim Özellikleri

FPGA' ların güç tüketim özelliklerinden bahsetmeden önce güç kavramını anlamak gerekir. Devrelerde harcanan toplam güç, statik ve dinamik güçlerin toplamıdır. Statik güç, devrenin sükunet halinde harcadığı güçtür. Dinamik güç ise devredeki elemenların çıkışlarının 0'dan 1'e veya 1'den 0'a lojik geçişleri sırasında harcadıkları güçtür. Statik güç, devre çıkışındaki lojik geçişler ve devrenin çalışma frekansından bağımsızdır. Halbuki dinamik güç, kullanılan devre yapısına göre farklılık gösterir. Bu yüzden çalışma boyunca kullanılan güç kavramı, aksi belirtilmediği sürece, dinamik gücü temsil etmektedir.

Günümüzde en çok kullanılan programlama teknolojileri SRAM (Static Random Access Memory), anti-sigorta ve FLASH programlamadır. Bunların dışında, sigorta, EPROM (Erasable Programmable Read Only Memory) ve EEPROM (Electrically Erasable Programmable Read Only Memory) teknolojileri de mevcuttur. Ticari amaçlı kullanılan bazı FPGA'lar Çizelge 2.1' de verilmiştir.

Çizelge 2.1 : Bazı ticari FPGA'lar.

Üretici	Mimari	Lojik Blok Tipi	Programlama Teknolojisi
Actel	Satır Bazlı	Çoğullayıcı Bazlı	Anti-Sigorta
Altera	Hiyerarşik PLD	PLD Blok	EPROM
QuickLogic	Simetrik Dizi	Çoğullayıcı Bazlı	Anti-Sigorta
Xilinx	Simetrik Dizi	Doğruluk Tablosu	Statik RAM

Anti-sigorta tabanlı FPGA' lar sadece bir kez programlanabilir. Bu cihazlar, programlandıktan sonra birer ASIC gibi ele alınabilir. Ancak enerjileri kesildiğinde, SRAM tabanlı FPGA' ların aksine kurgularını korurlar. SRAM tabanlı FPGA' lar ise tekrar tekrar programlanabilmektedir. Bu cihazlar, her ilklendirmede, kurgulama belleğine, kurgu bilgisini yeniden yükler. Enerjileri kesildiğinde ise kurgulama belleği silinir. Bu sayede, sistemin donanımsal tasarımını değiştirmeden, sadece kurgulama dosyası değiştirilerek, sistemin işlevi değiştirilebilir. Bu özellik, araştırma-geliştirme uygulamalarında büyük kolaylık sağlamakta ve sistem maliyeti düşürmektedir. Ürünün çalıştığı ortamda güncellenebilmesi SRAM FPGA' ların seçilmesindeki en önemli etkendir [6-8].

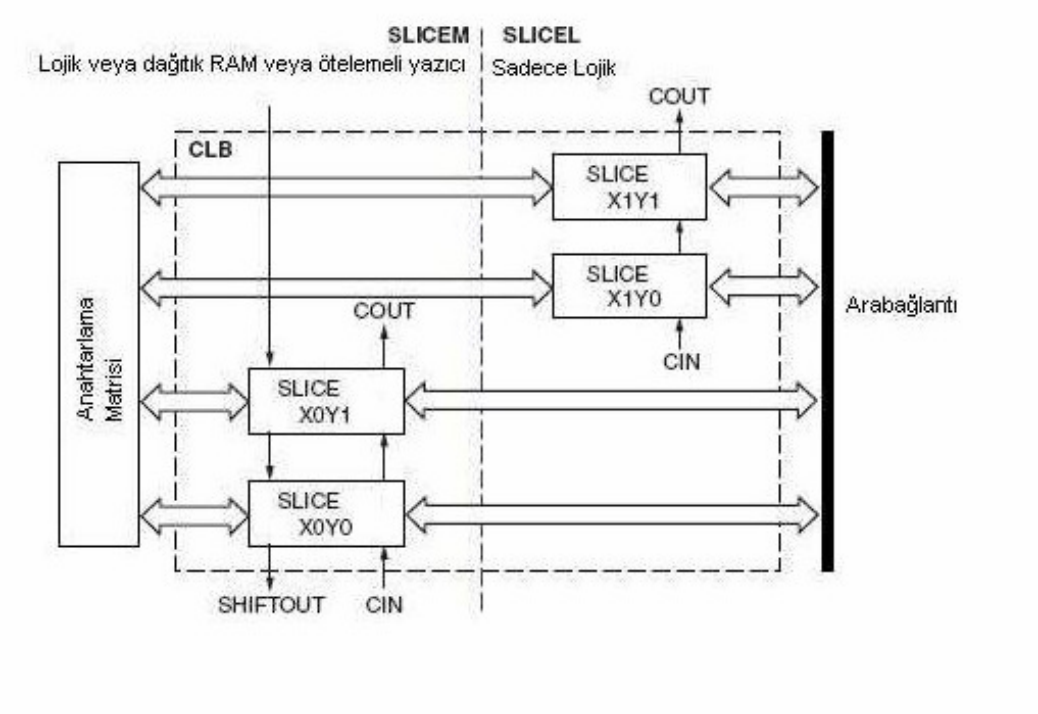
Kullanılan programlama teknolojisine göre devrelerin harcadığı güç değişmektedir. SRAM teknolojisinde her ilklendirmede tekrar kurgu yükleme bilgisine bağlı olarak güç harcaması anti-sigorta teknolojisine oranla daha fazladır. Ayrıca anti-sigorta FPGA' lar ara bağlantı noktalarındaki düşük çıkış kapasiteleri nedeniyle SRAM FPGA' lara oranla çok daha az miktarlarda dinamik güç harcar. SRAM tabanlı FPGA' larda dışardan bağlanan EPROM veya Flash bellek ile konfigürasyon sağlanır. Ayrıca konfigürasyon kurulumu için de güç tüketimi yapılmaktadır. Bu bilgiler ışığında SRAM tabanlı FPGA' ların anti-sigorta tabanlı FPGA'lara oranla çok daha fazla güç tükettikleri açıktır [3]. Düşük güç tüketim avantajına sahip olan anti-sigorta tabanlı FPGA' lar belirlenmiş bir tasarıma uygun olarak üretildikleri için başka bir tasarım için kullanılmaları mümkün değildir. Bu yüzden yapılan çalışma sırasında anti-sigorta tabanlı FPGA' lar düşük güç tüketecekleri bilindiği halde seçilmemişlerdir. Seçilen FPGA SRAM tabanlıdır. Çalışma süresince, SRAM tabanlı FPGA' ların güç tüketim dezavantajları göz önüne alınmıştır.

2.2 Gerçekleme Sırasında Kullanılan FPGA

DES Blok Şifreleme Algoritmasının FPGA üzerinde düşük enerjili tasarımı çalışmasında Xilinx firmasına ait Spartan-3 XC3S5000 FPGA' sı kullanılmıştır. XC3S5000 FPGA' sı çok büyük bir FPGA' dır. 633 adet giriş çıkış ve 8320 adet konfigüre edilebilir lojik bloğa kadar destek sağlayabilmektedir [9]. Çalışma sırasında farklı DES yapıları gerçeklenmiş ve karşılaştırılmıştır. Çalışma başında Xilinx firmasına ait VirtexE XCV1000E FPGA' sı kullanılmaya çalışılmış fakat tasarlanan devrede 194 adet giriş çıkış olduğundan ve XCV1000E FPGA' da en fazla

162 adet giriş çıkış desteklendiğinden bu FPGA kullanılamamıştır. Daha sonra VirtexE XCV2000E FPGA' sı seçilerek tasarım yapılabilmiştir. XCV2000E FPGA' da 404 adet giriş çıkış desteklendiğinden tasarıma uymuştur. Ancak XCV2000E FPGA kullanılarak tasarlanın devrenin XPOWER güç ölçüm aracı ile yapılan güç analizinde besleme gerilimi sükunet değeri sıfır çıkmıştır. Yapılan güç analizlerini etkilememesi amacıyla sükunet besleme geriliminin gerçek değerini verebilen Spartan-3 serisinin en gelişmiş versiyonu olan XC3S5000 FPGA' sı seçilerek bütün DES yapıları gerçeklenmiş ve karşılaştırmalar adil bir şekilde yapılmıştır. Gerçekleme aşamasında VHDL dili kullanılmış olup, FPGA' da sentezleme, yerleştirme ve yollandırma aşamaları için Xilinx ISE 9.2i programı kullanılmıştır. Ayrıca simülasyon aşamasında Modelsim 6.3c ve güç ölçümleri için de XPOWER aracı kullanılmıştır.

Diğer FPGA' larda olduğu gibi XC3S5000 de konfigüre edilebilir lojik blok yapısı (CLB), giriş çıkış bağlantıları ve Blok RAM' lardan oluşmaktadır. CLB yapıları RAM tabanlı LUT (Look-Up Table)' ler içerir, yazmaç (register) olarak veya lojik işlemler için kullanılır. Bu bloklar genel yollandırma hattına bağlanmaktadır. Her bir CLB için 4 ara bağlantı dilimi bulunur, bu ara bağlantı dilimleri SLICEM ve SLICEL olarak adlandırılır. Şekil 2.2' de bu yapı verilmektedir. Giriş/Çıkış blokları, giriş/çıkış pinleri ile iç yapıdaki elemanlar arasındaki veri akışını kontrol eder. Her bir giriş/çıkış bloğu çift yönlü iletimin yanında 3-durumlu çıkış kontrolünü de destekler. Blok RAM' lar ise veri depolama görevini üstlenirler. XC3S5000 FPGA' sı 4 adet Blok RAM sütununa sahiptir [9].



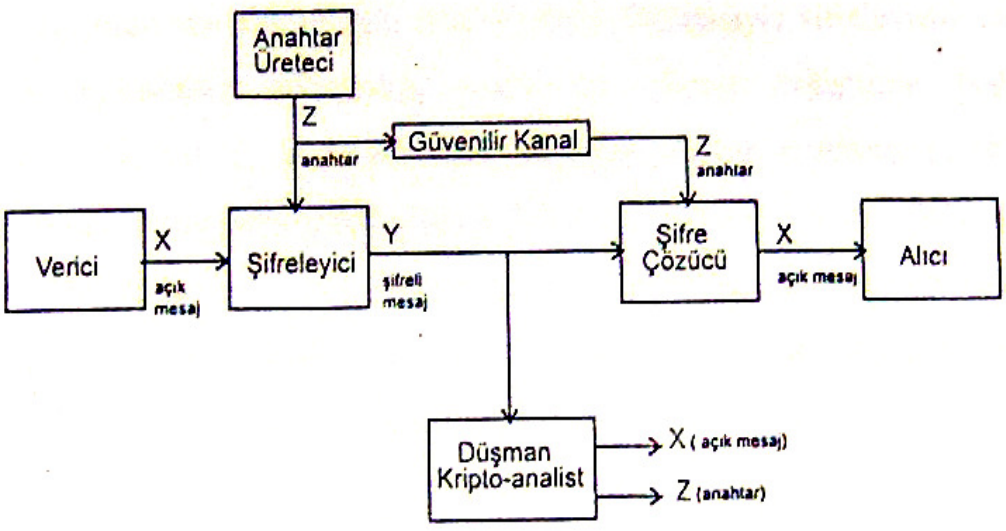
Şekil 2.2 : Spartan 3 CLB Ara Bağlantıları.

3. DES BLOK ŞİFRELEME ALGORİTMASI ve GERÇEKLENMESİ

3.1 Blok Şifreleme Sistemleri

Günümüzde şifreleme sistemleri, şifreleme anahtarının gizli tutulduğu "gizli-anahtarlı" ve şifreleme anahtarının açık olduğu "açık-anahtarlı" sistemler olmak üzere ikiye ayrılırlar. Bu sistemler sırasıyla "simetrik" ve "anti-simetrik" şifreleme sistemleri olarak da bilinirler. Açık-anahtarlı ya da anti-simetrik şifreleme sistemlerine örnek RSA algoritmasının kullanıldığı sistemlerdir. Bu tür sistemlerde şifreleme ve şifre çözme işlemleri, gizli-anahtarlı sistemlerin aksine, farklı anahtarlar kullanılarak gerçekleştirilir. Bu nedenle açık-anahtarlı sistemler simetrik olmayan bir yapıya sahiptir [10].

Simetrik yapılı gizli-anahtarlı şifreleme sistemlerinde şifreleme ve şifre çözme anahtarları aynıdır. Şekil 3.1' de gizli-anahtarlı sistemin genel yapısı görülmektedir:



Şekil 3.1 : Gizli-anahtarlı şifreleme sisteminin genel görünüşü.

Sistemde *E* şifreleme, *D* de şifre çözme dönüşümlerini göstermektedirler. Şifreleme ve şifre çözme işlemleri aynı anahtarı kullanılarak gerçekleştirilir:

$Y = E_Z(X)$ (Şifreleme) **(3.1)**

$X = D_Z(Y)$ (Şifre çözme) **(3.2)**

Düşman kripto-analistin gizli anahtara erişimi güvenilir bir kanalla engellendiği taktirde sistem oldukça güvenilir olabilmektedir.

Gizli anahtar kullanan simetrik şifreleme sistemleri iki gruba ayrılır: Dizi Şifreleme (Stream Cipher) ve Blok Şifreleme (Block Cipher) sistemleri.

"Dizi şifreleme" sistemleri her bir mesaj birimini, zamanla değişen bir fonksiyon kullanarak şifreler. Dizi şifreleyiciler yüksek hızlı iletişim için en iyi alternatiflerden biridir. Yüksek hatalı iletişim ortamlarında hata riskini azalttığından tercih sebebidirler [3]. Dizi şifreleme sisteminin iç yapısı zamana bağlı olarak durum değiştiren bir makine gibi düşünülebilir. Dolayısıyla şifrelemede kullanılan fonksiyonun zamana bağımlılığı makinenin durum değiştirme bağıntısıyla tanımlanır. Bu tür bir sistemde mesaj biriminin büyük olmasına gerek yoktur. Aksine mesaj biriminin olabildiğince küçük tutulması istenir. Dizi şifreleme sistemlerinde kullanılan mesaj birimleri genelde Latin alfabesinden bir karakter ya da tek dijitlik sayılardır. Bir dizi şifreleme sisteminde mesaj dijitleri tek tek şifrelendiğinden mesajlar genelde bir dizi şeklinde düşünülür. Mesaj dizisinin her bir dijiti şifrelendikten sonra sistem belirli bir kural uyarınca durum değiştirir. Sistemin durum değiştirmesi şifreleme anahtarının da değişmesine neden olur. Bu nedenle dizi şifreleme sistemlerinde anahtar da mesajla aynı boyutta bir dizi şeklindedir. Bu anahtar dizisinin yapısı şifreleme sisteminin güvenilirliği konusunda oldukça büyük bir önem taşımaktadır.

İkinci simetrik şifreleme sistemi olan "Blok Şifreleme" sistemi, en temel anlamıyla basit yerine koyma sistemleridir. Blok Şifreleyiciler, sabit uzunluktaki bloklar halinde aldıkları açık veriyi, yine aynı uzunluktaki bloklar halinde şifrelenmiş veriye çeviren simetrik anahtarlı şifreleme algoritmalarıdır. Bu dönüştürme işlemi kullanıcı tarafından belirlenen gizli bir anahtar kullanılarak yapılır. Şifre çözme işlemi de, yine sabit uzunluktaki bloklar halinde alınan kapalı verinin bu kez ters dönüşümden geçirilerek, bloklar halinde açık veriye dönüştürülmesiyle gerçekleştirilir [8].

Blok şifreleyiciler günümüzde şifreleme işlemlerinde yaygın olarak kullanılmaktadır. En çok bilinenlere örnek olarak; 64 bitlik blokları kullanan DES ve 128 bitlik

blokları kullanan AES algoritmaları gösterilebilir. Şifrelenecek olan verinin uzunluğu blok uzunluğundan daha büyük olması durumunda veri blok uzunluklarına parçalanır ve şifreleme işlemi gerçekleştirilir. Veriyi blok uzunluklarına parçaladıktan sonra şifreleme işlemi Elektronik Kod Kitabı (ECB), Şifre Bloklarını Zincirleme (CBC), Şifreyi Geri Besleme (CFB) ve Çıkışı Geri Besleme (OFB) gibi yöntemlerle gerçeklenebilir. Seçilen yöntem şifrelenmiş veri üzerinde çok büyük bir etkiye sahiptir. Aynı veri bloğunun, aynı anahtar bloğu ile şifrelenmesi sonucu hep aynı çıkış bloğu oluşur, bu nedenle şifreleme işleminde güvensizlik oluşabilir. Kullanılan farklı yöntemler sayesinde şifreleme işlemi daha güvenilir hale getirilir [3].

Bu çalışmada, DES Blok Şifreleme algoritması en çok bilinen yöntem olan Elektronik Kod Kitabı (ECB) yöntemiyle gerçeklenmiştir. ECB yönteminde giriş verisi olarak gelen her düz metin bloğu diğer bloklardan bağımsız olarak şifrelenir veya çözülür.

3.2 DES Blok Şifreleme Algoritması

Günümüzde teknolojinin sürekli geliştiği ve çok hızlı bir şekilde gelişmeye devam edeceği bilinen bir gerçektir. Gelişen teknoloji ile birlikte veri iletimi ve iletilen verinin güvenliği önemli bir unsur haline gelmiştir. Şifreleme algoritmaları, iletilen verinin güvenliği için kullanılan yöntemlerdir. Bu algoritmaların teknolojinin gerektirdiği şekilde olacağı, teknolojiye ayak uyduramayanların kullanımının terk edileceği, yeni algoritmaların bulunacağı bilinen bir gerçektir. Teknoloji ile gelişen bu algoritmalar standartlaşma ihtiyacı hissetmektedir. Bu yüzden birçok ülke kendi standartlaşma enstitüsünü kurmuştur. ABD Ulusal Standartlar ve Teknolojiler Ensititüsü (NIST) de bunlardan birisidir [11]. Bu bölümün devamında NIST' ın standart olarak kabul ettiği DES algoritması ayrıntılı olarak incelenecektir.

Veri Kodlama Standardı (Data Encryption Standart (DES)), 1977 yılında Amerikan Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology (NIST)) tarafından standart olarak kabul edilmiş ve Federal Bilgi İşleme Standardı (Federal Information Processing Standards (FIPS)) olarak yayınlanmıştır [4]. İlk başlarda 10-15 yıl standart olarak kullanılacağı düşünülen DES algoritması, uzun yıllar standart olarak kullanılmış, ancak 1999 yılında yerini Gelişmiş Kodlama Standardı (Advanced Encryption Standard (AES))' e bırakmıştır [12]. DES algoritması, bilgiyi şifrelemek ve çözmek için kullanılan simetrik bir blok şifreleyicidir.

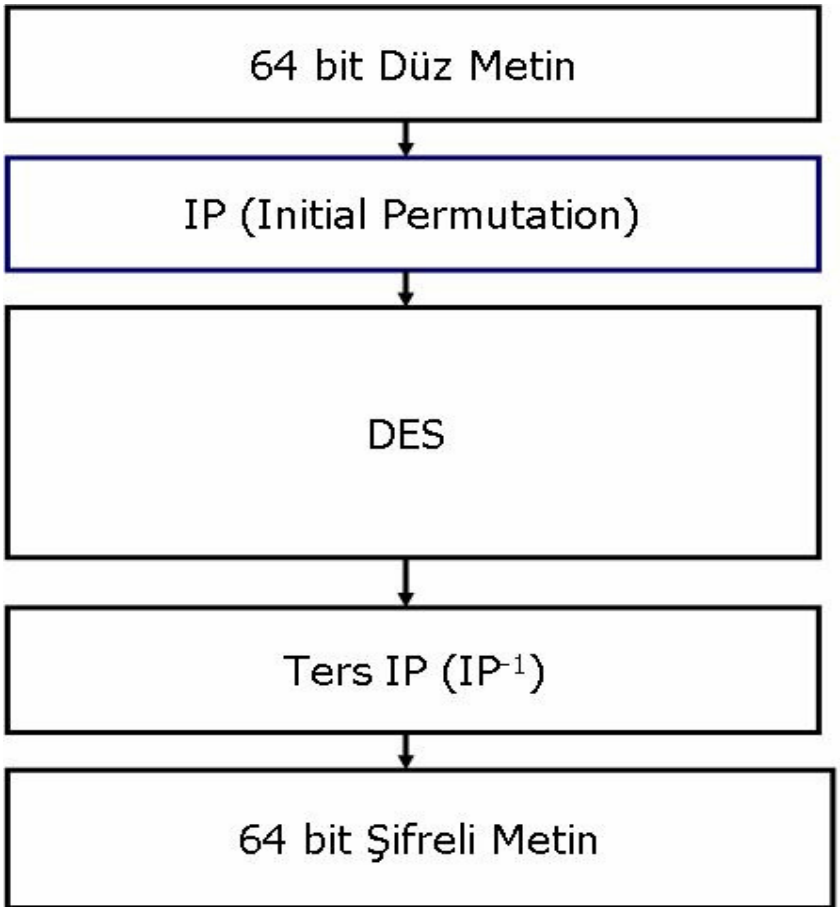
DES Blok Şifreleme Algoritması, 64 bitlik anahtar ile 64 bitlik düz metni 64 bitlik şifreli metne dönüştüren senkron bir blok şifreleme sistemidir. DES algoritması, daha önceden bulunmuş olan Feistel şifreleme yönteminin özel bir versiyonudur. Feistel tipi şifrelemede, her adımda veri iki eşit uzunluktaki blok halinde şifrelenir. Bu iki eşit blok genelde sol yarı ve sağ yarı blok olarak tanımlanır. DES algoritmasında, şekil 3.2'de görüldüğü gibi, şifrelenecek metin öncelikle IP (Initial Permutation – ilk permütasyon)' dan geçer. Daha sonra DES kutusundan geçer ve ters IP sonucunda şifreli metin elde edilir.

DES algoritması için kullanılan 64 bitlik anahtarın 56 biti algoritma içinde işlemden geçer. Kalan 8 bit benzerlik veya hata bulmak için kullanılabilir.

64 bitlik düz metin öncelikle IP' den geçer. Permütasyon işleminde, bitlerin yerleri değiştirilerek şifrelemenin ilk adımı gerçekleştirilir. İlk Permütasyondaki bit değişim sırası şöyledir [4]:

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7



Şekil 3.2 : DES Algoritması Genel Yapısı.

IP işleminden sonra girişin 58. biti ilk bit, 50. biti ise 2. bit olur ve bu şekilde devam eder. Permütasyondan geçen giriş bloğu daha sonra karmaşık anahtar bağımlı DES kutusuna girer. DES kutusu çıkışı ise ters ilk permütasyona (IP^{-1}) girer. Ters ilk permütasyon sırasındaki bit değişim sırası şöyledir [4]:

IP^{-1}

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Burda da girişin 40. biti ilk bit, 8. biti ikinci bit olur ve böyle devam eder. Ters IP çıkışında ise şifreli metin elde edilmiş olur.

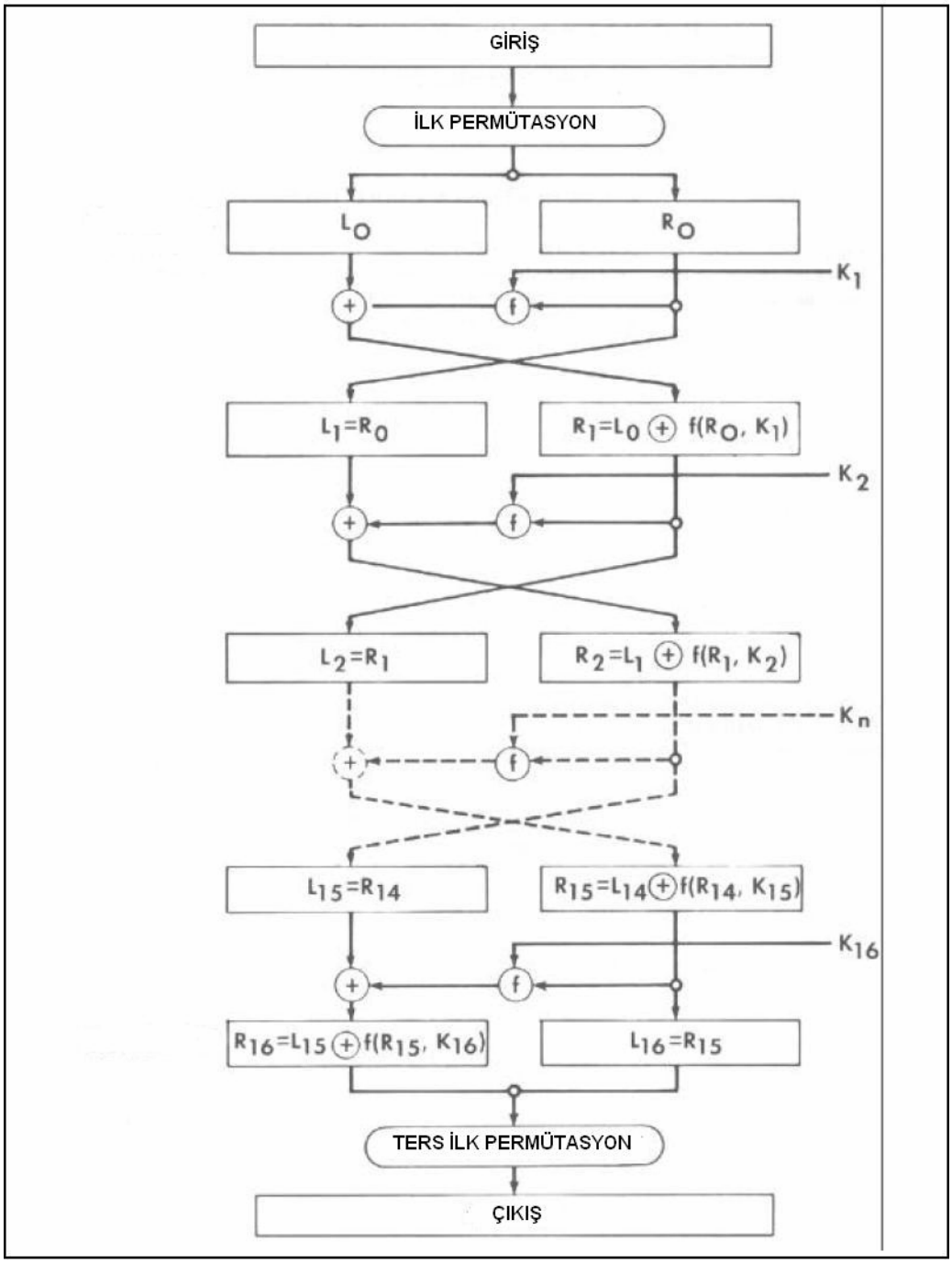
DES algoritması Feistel turu da denilen 16 turdan oluşur. Şekil 3.3' te DES kutusunun içeriği ve turların dağılımı görülmektedir.

İlk permütasyon çıkışında oluşan sağ ve sol yarı bloklar birinci tur işlemlerine girer. Sonraki turun sol yarı bloğu, var olan turun sağ yarı bloğudur. Sağ yarı blok ise anahtar ile birlikte DES *f* fonksiyonuna girer. DES *f* fonksiyonu çıkışı, sol yarım küre ile e-xor' lanarak sonraki turun sağ yarı bloğunu oluşturur. Denklem 3.3 ve denklem 3.4' te tur içindeki işlemler sembolize edilmiştir. L' ve R' sonraki turu ifade eder. Turlar bu şekilde devam eder ve 16. tur sonunda oluşan sol ve sağ yarı bloklar ters IP fonksiyonuna girer. Diğer turlardan farklı olarak 16. turdaki çıkışlar ters IP fonksiyonuna şekil 3.3' te görüldüğü gibi yer değiştirerek girer.

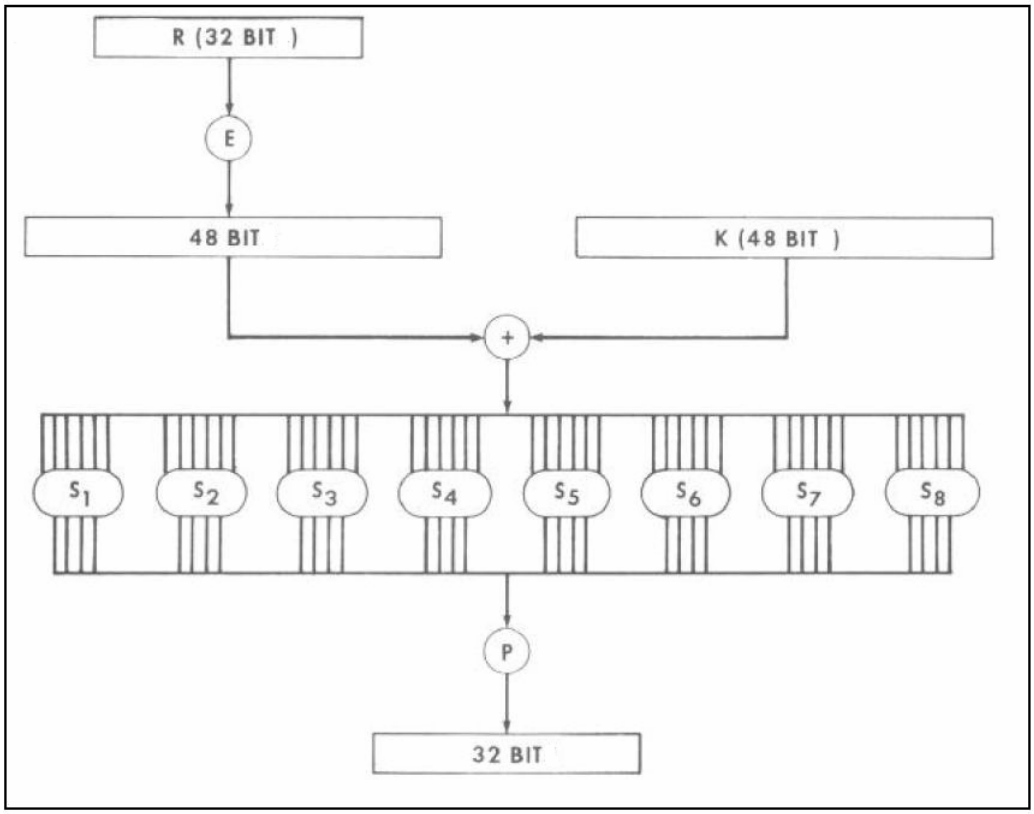
$$L' = R \tag{3.3}$$

$$R' = L \oplus f(R,K) \tag{3.4}$$

Her DES turunda bulunan DES *f* fonksiyonu şekil 3.4'te verilmiştir.



Şekil 3.3 : DES Kutusu ve Turların Dağılımı.



Şekil 3.4 : DES *f* fonksiyonu.

Şekilden de görüldüğü gibi, var olan turdaki 32 bitlik sağ yarı blok E (Expansion) fonksiyonu ile 48 bite genişletilir. DES *f* fonksiyonu içindeki E fonksiyonu şöyledir [4]:

E FONKSİYONU

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

E fonksiyonu çıkışındaki 48 bit, 48 bitlik anahtar ile e-xor' lanarak S-Kutusu girişlerini oluşturur. S-Kutularına 6 bit girer ve 4 bit çıkar. S-Kutularının hangi girişe

karşılılık hangi çıkışı vereceği [4]' te bulunabilir. 6 bitlik girişin ilk ve son bitleri s-kutusundaki çıkışın satır numarasını, girişin ortada kalan 4 biti ise çıkışın sütun numarasını verir. S-kutusunda karşılık gelen değer 4 bitlik çıkışı verir.

8 adet 4 bitlik s-kutusu çıkışları son olarak olarak permütasyondan geçer. DES *f* fonksiyonu içindeki permütasyon şu şekilde bit yer değişimi yapar [4]:

P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

Son permütasyon çıkışındaki 32 bit, DES *f* fonksiyonunun çıkışı olur.

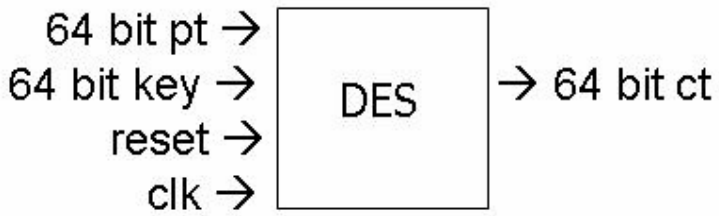
Böylelikle DES algoritmasındaki bütün fonksiyonlar tamamlanarak 64 bitlik anahtar ile 64 bitlik düz metin, 64 bitlik şifreli metne dönüşmüş olur.

3.3 Gerçekleme Adımları

DES Blok Şifreleme Algoritmasının FPGA üzerinde düşük enerjili tasarımı yüksek lisans tez çalışmasının ilk aşaması olarak VHDL donanım betimleme dilinin öğrenilmesi gelmektedir. Bilinen en güncel donanım betimleme dillerinden biri olan VHDL' nin öğrenilmesinde [13]' ten yararlanılmıştır.

Gerçekleme aşamasında VHDL dili kullanılmış olup, FPGA'da sentezleme, yerleştirme ve yollandırma aşamaları için Xilinx ISE 9.2i programı kullanılmıştır. Ayrıca simülasyon aşamasında Modelsim 6.3c ve güç ölçümleri için de XPOWER aracı kullanılmıştır.

DES algoritmasının VHDL ile yazılımı için öncelikle [4] dökümanı çok iyi anlaşılmış ve daha sonra algoritmanın yazılmasına başlanmıştır. Bölüm 3.2' de anlatılan yapıların VHDL kodu yazılmıştır. Kodun tamamının bir dosyada yazılması yerine içiçe fonksiyonlar yazılarak kodun anlaşılması ve yazılması kolaylaştırılmıştır. DES ana fonksiyonunun blok şeması şekil 3.5' te verilmiştir. DES kutusuna 64 bitlik düz metin ve anahtarla birlikte saat ve yeniden başlatma işaretleri girer, 64 bitlik şifreli metin çıkar.



Şekil 3.5 : DES ana fonksiyonu.

DES algoritmasında kullanılan temel alt bloklar şunlardır: keysched (anahtar üretim) bloğu, ip (ilk permütasyon) bloğu, roundfunc (tur fonksiyonu) bloğu ve fp (ters ilk permütasyon) bloğudur.

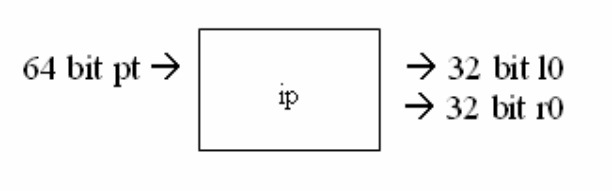
keysched alt bileşenine 64 bitlik anahtar girer ve her bir turda kullanılmak üzere 48 bitlik 16 adet anahtar çıkar. Şekil 3.6' da blok şema verilmiştir. keysched bileşeni iki alt bileşenden oluşur: PC1 ve PC2.

Şekil 3.6 : Anahtar üretim fonksiyonu.

PC1 alt bileşenine 64 bitlik anahtar girer ve 56 bit olarak çıkar. PC1 alt bileşeninde permütasyon ve sola kaydırma işlemleri yapılır. Atılan 8 bit hata veya benzerlik tanımlama için kullanılır.

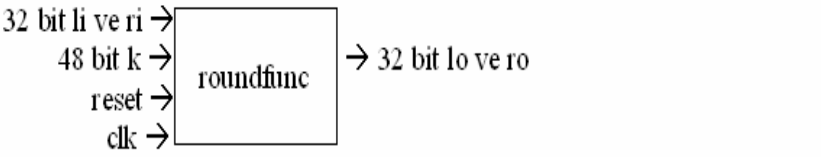
PC2 alt bileşenine ise 56 bit girer ve 48 bitlik anahtar çıkar. Bu alt bileşende de yine permütasyon yapılır.

ip alt bileşeninde permütasyon yapılır. 64 bitlik düz metin girer, 32 bitlik sol ve sağ yarı blokları çıkar. Şekil 3.7' de blok şeması verilmiştir.



Şekil 3.7 : İlk permütasyon fonksiyonu.

Üçüncü temel alt bileşen olan tur fonksiyonu, DES kutusunun en temel bileşenidir. Şekil 3.8' de blok şeması verilmiştir. Tur fonksiyonu bileşenine 32 bitlik sol yarı ve sağ yarı blokları ile birlikte 48 bitlik anahtar, saat ve yeniden başlatma işaretleri girer, 32 bitlik sol yarı ve sağ yarı blokları çıkar.



Şekil 3.8 : DES tur fonksiyonu.

DES tur fonksiyonu bileşeninin 13 adet alt bileşeni vardır.

xp (genişletme) alt bileşeninde 32 bitlik giriş, bazı bitler tekrarlanarak ve permütasyon yapılarak 48 bitlik çıkış elde edilir.

desxor1 alt bileşeninde xp'den çıkan 48 bit ile 48 bitlik anahtar e-xor'lanır. Sonuçta 8 adet 6'şar bitlik çıkış s-kutularına giriş olur.

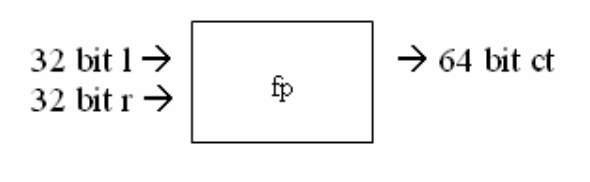
s1,s2,…,s8 s-kutularına 6 bit girip 4 bit çıkar. Bir çeşit LUT görevi yaparlar.

pp (son permütasyon) alt bileşeninde yine permütasyon yapılır. 8 adet dörder bitlik s-kutusu çıkışı girer ve 32 bit olarak permütasyon sonucu çıkar.

desxor2 alt bileşeninde ise yine pp çıkışı ile bir önceki turun sol yarısı e-xor' lanır.

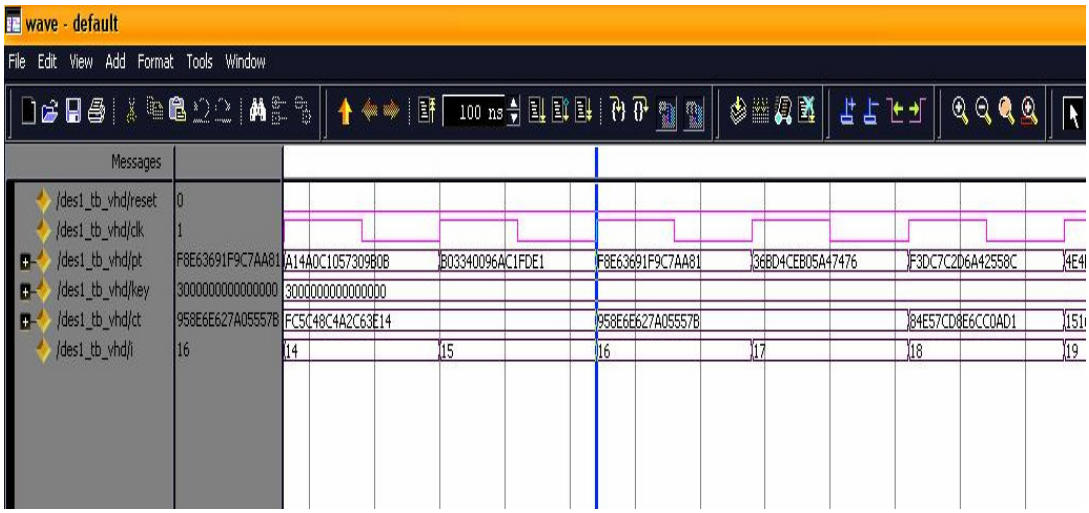
reg32 alt bileşeninde ise her bir turda kullanılan yazmaç (register) yapısı vardır.

Son alt bileşen olan fp (ters ilk permütasyon) bileşeninde ise adından da anlaşılacağı üzere ip' de yapılan permütasyonun tersi yapılır. 32 bitlik sol ve sağ yarı blokları girer, 64 bitlik şifreli metin çıkar. Şekil 3.9' da blok şeması verilmiştir.



Şekil 3.9 : Ters ilk permütasyon fonksiyonu.

DES algoritmasının kodu yazıldıktan sonra, şekil 2.1' deki tasarım sürecine uygun olarak fonksiyonel benzetim (simülasyon) yapılmıştır. ModelSim programı yardımıyla yapılan simülasyon sonucu kodun önceden bilinen üç adet giriş çıkış değeri için doğru çalıştığı gözlenmiştir. Şekil 3.10' da ModelSim programında elde edilen simülasyon sonucu verilmiştir.



Şekil 3.10 : DES simülasyon sonucu.

Yazılan kodun doğru çalıştığına emin olunduktan sonra, tasarım sürecinde bir sonraki adım olan FPGA üzerinde gerçeklemeye geçilmiştir.

3.4 FPGA Üzerinde Gerçekleme

DES Blok Şifreleme Algoritmasının FPGA üzerinde düşük enerjili tasarımı yüksek lisans tez çalışmasının önemli bir aşaması da yazılan VHDL kodlarının FPGA donanımı üzerinde gerçeklenmesidir. Gerçeklemeler sırasında devrelerin alan, zaman, iş hacmi (throughput), güç ve enerji bilgileri toplanmıştır. Bu çalışmada bölüm 2.2' de de bahsedildiği gibi Xilinx firmasına ait Spartan-3 XC3S5000 FPGA' sı kullanılmıştır. Çalışma sırasında farklı DES yapıları gerçeklenmiş ve karşılaştırılmıştır. Adil bir karşılaştırma olması amacıyla bölüm 5.2' de anlatılacak olan bütün gerçeklemelerde aynı adımlar takip edilmiştir.

Yazılan bir VHDL kodunun doğruluğu yapılan simülasyon sonucu anlaşıldıktan sonra FPGA üzerinde gerçekleme işlemi Xilinx ISE 9.2i programında kodun sentezlenmesiyle başlar. Sentezleme aşamasında dikkat edilmesi gereken noktalar vardır. Yazılan kodun çalışma amacına bağlı olarak, sentezleme seçeneklerinden optimizasyon hedefi (optimization goal) olarak hız veya alan seçilir. Ayrıca optimizasyon girişimi (optimization effort) normalden yükseğe çekilerek FPGA üzerinde en iyi gerçekleme sağlanmıştır. Sentezleme sonucunda devre ile ilgili zaman ve alan bilgileri elde edilir. Fakat bu değerler sentezleme sırasındaki tahmini değerlerdir, gerçek değerler için yerleştirme ve yollandırma (place & route) sonrasındaki alan ve zaman bilgilerine bakmak gerekir. Örneğin işhattı (pipeline) gerçekleme için sentezleme sonucu alan ve zaman bilgileri şöyledir:

Device utilization summary:

Selected Device : 3s5000fg900-5

Number of Slices:	2129	out of 33280	6%
Number of Slice Flip Flops:	1024	out of 66560	1%
Number of 4 input LUTs:	4123	out of 66560	6%
Number of IOs:	194		
Number of bonded IOBs:	186	out of 633	29%
Number of GCLKs:	1	out of 8	12%

Timing Summary:

Speed Grade: -5
Minimum period: 5.628ns (Maximum Frequency: 177.688MHz)
Minimum input arrival time before clock: 6.936ns
Maximum output required time after clock: 6.216ns
Maximum combinational path delay: No path found

Görüldüğü gibi devrenin çalışabileceği en yüksek saat frekansı yaklaşık 178 MHz (Mega-Hertz) çıkmıştır. Halbuki gerçek değerler yerleştirme ve yollandırma sonucu elde edilecektir.

Sentezleme işleminden sonra FPGA üzerinde dönüştürme (translate), eşleştirme (map), yerleştirme ve yollandırma (place & route) işlemleri gerçekleştirilmiştir. Ayrıca diğer bir önemli nokta da kullanıcı kısıtlamalarıdır (user constraints). Kullanıcı kısıtlamaları olarak saat frekansı verilmiş ve devrenin çalışabileceği en yüksek saat frekansı iterasyonla bulunmuştur. Bu işlemlerin sonucunda oluşan zamanlama ve eşleştirme raporlarından devrenin gerçek alan ve zaman bilgileri elde edilmiştir. Yine örnek olarak işhattı gerçeklemesi için alan ve zaman bilgileri şöyledir:

```
Design Summary
--------------
Logic Utilization:
  Number of Slice Flip Flops:        928 out of  66,560    1%
  Number of 4 input LUTs:          4,125 out of  66,560    6%
Logic Distribution:
  Number of occupied Slices:       2,320 out of  33,280    6%

Timing summary:
---------------

Design statistics:
   Minimum period:   9.796ns   (Maximum frequency: 102.082MHz)
```

Görüldüğü gibi alan ve zaman bilgileri sentezleme sonucuna göre farklılık göstermektedir.

Devrelerin harcadığı güç ve enerji bilgilerini elde edebilmek için yerleştirme ve yollandırma sonrası simülasyon (post-route simulation) yapılır. Bu simülasyon sırasında devreye MATLAB programı tarafından oluşturulmuş rastgele girişler uygulanır ve devre mümkün olan en yüksek saat frekansında çalıştırılarak simülasyon sonuçları elde edilir. Burada dikkat edilmesi gereken nokta, yerleştirme ve yollandırma sonrası simülasyonda kullanılacak saat frekansıdır. Yerleştirme ve yollandırma sonucu elde edilen rapordaki en yüksek saat frekansında devre içindeki kapı gecikmelerine bağlı olarak devre doğru çıkışları üretemeyebilir. Çıkışların doğruluğu kontrol edilerek, mümkün olan en yüksek saat frekansı iterasyonla bulunmalıdır. Örneğin işhattı gerçeklemesi için saat frekansı 102 MHz verilerek yapılan yerleştirme ve yollandırma sonrası simülasyon sonucu çıkışlar yanlış

olmaktadır. Bu sebeple saat frekansı iterasyonla düşürülerek 83 MHz verilmiş ve yapılan simülasyon doğru sonuçları vermiştir. Simülasyon sırasında güç analizi işleminde kullanılmak üzere .vcd uzantılı simülasyon dosyası da oluşturulmuştur. Güç analizi adımları için [3] no' lu kaynaktan yararlanılmıştır. Son olarak elde edilen simülasyon dosyası kullanılarak XPOWER güç ölçüm aracı yardımıyla devrenin harcadığı güç bilgisi elde edilmiştir. Örneğin işhattı gerçeklemesi için güç değeri 48,7 mW (mili-watt) çıkmıştır. Böylelikle yazılan VHDL kodunun FPGA üzerinde gerçeklenmesi aşaması tamamlanmıştır. Bununla birlikte farklı DES yapıları için elde edilen alan, zaman, iş hacmi (throughput), güç ve enerji bilgileri bölüm 5.2 ve 5.3' te detaylı olarak anlatılacaktır.

4. ENERJİ TASARRUFU YÖNTEMLERİ

Düşük enerji hedeflenen bir Kripto algoritması tasarımında, dikkat edilmesi gereken konular güç ve zamandır. Devrelerin harcadığı enerji, tükettiği güç ve şifreleme için kullanılan süreyle doğru orantılıdır. Bir elektronik devrede güç, devreye verilen girişlerin işlenmesi sırasında harcanır. Harcanan gücün miktarı devrenin yapısına bağlıdır. Ayrıca harcanan güç, uygulanan girişlere bağlı olduğundan zamanla değişmektedir. Enerjiyi etkileyen diğer faktör olan şifreleme zamanı da devrenin çalışma frekansına ve algoritmanın gerçeklenme şekline bağlı olarak değişim göstermektedir. Devrenin işlem hacmi, çalışma frekansına ve şifreleme için gerekli olan saat işareti sayısına bağlıdır. Düşük enerji tüketen devre elde edebilmek için hem düşük güç tüketimi hem de yüksek işlem hacmi gerekmektedir. Düşük enerji için ya harcanan gücü çok arttırmadan işlem hacmini arttırmayı ya da işlem hacmini çok düşürmeden harcanan gücü azaltmayı hedeflemek gerekir. Bu sebeple enerji tasarrufu yöntemleri işlem hacmini artırma ve güç tasarrufu yöntemleri olmak üzere iki ana alt başlıkta incelenebilir. Enerji tasarrufu yöntemleri, bu çalışmada kullanılabilecek yöntemler ön planda tutularak araştırılmıştır.

4.1 İşlem Hacmini Arttırma Yöntemleri

Kripto algoritmalarının tasarımında kullanılan işlem hacmini arttırma yöntemlerinden ilk akla gelen, tasarımın işhattı yapısında gerçeklenmesidir [14 – 17]. İşhattı gerçeklemesi, turların arasına yazmaç eklenmesi ile gerçekleştirilir. Örneğin DES devresinin hızlanması için 16 adet tur artarda dizilir ve döngü yapısından açık yapıya geçilir. Şifrelenecek veri 16 adet aşamadan geçer ve aralarda yazmaç olduğu için kritik yol da kısalır [14 – 15]. Kritik yolların kısalması devre seviyesinin düşmesini ve işlem hacminin artmasını sağlar [16]. Ayrıca işhattı yapısı algoritmayı eşit bloklara böler ve her bir saat darbesinde işaretlerin işlem görmesi için gereken yeterli süreyi sağlayarak devrenin saat frekansını arttırır [17]. Bölüm 5.2.1' de DES algoritmasının işhattı yapısında gerçeklenmesi anlatılacak ve işhattı yapısı ile ilgili daha detaylı bilgi verilecektir.

İşlem hacmini arttırmak için başvurulan diğer bir yöntem ise işhattı yapısının özelleşmiş bir hali olan iç yazmaçlı işhattı yapısıdır. İç yazmaçlı işhattı yapılarında, tasarımdaki turların arasındaki yazmaçlara ek olarak turların içine de yazmaç eklenir [16]. Yine DES algoritmasının tasarımında tur içindeki kritik yolları azaltmak amacıyla, tur içine iki, üç veya dört adet yazmaç eklenebilir. Eklenen yazmaç sayısına göre devrenin işhattı seviyesi belirlenir. İki adet iç yazmacı olan devre 32 seviyeli, üç adet iç yazmacı olan devre 48 seviyeli ve dört adet iç yazmacı olan devre 64 seviyeli işhattı yapısına sahip olur. Şekil 3.4' teki DES *f* fonksiyonunda E fonksiyonu çıkışına, S – kutularının girişlerine ve çıkışlarına, permütasyon fonksiyonunun çıkışına iç yazmaçlar eklenebilir. Böylelikle devrenin saat frekansı ve dolayısıyla işlem hacmi arttırılmış olur. Bölüm 5.2.6' da 64 seviyeli işhattı yapısına sahip DES devresi anlatılacak ve iç yazmaçlı işhattı yapısından daha detaylı bahsedilecektir.

Devrelerde işlem hacmini arttırmak için önerilen bir diğer yöntem ise devreyi asenkron yapıda tasarlamak ve genel saat işaretinden kurtulmaktır [18 – 19]. Fakat geniş çapta asenkron tasarım ASIC teknolojisi için geçerli olsa da henüz asenkron tasarım tekniklerini destekleyen ticari FPGA' lar bulunmamaktadır [16]. Bu yüzden FPGA üzerinde tasarım hedeflenen bu çalışmada asenkron tasarım tekniği kullanılmamıştır.

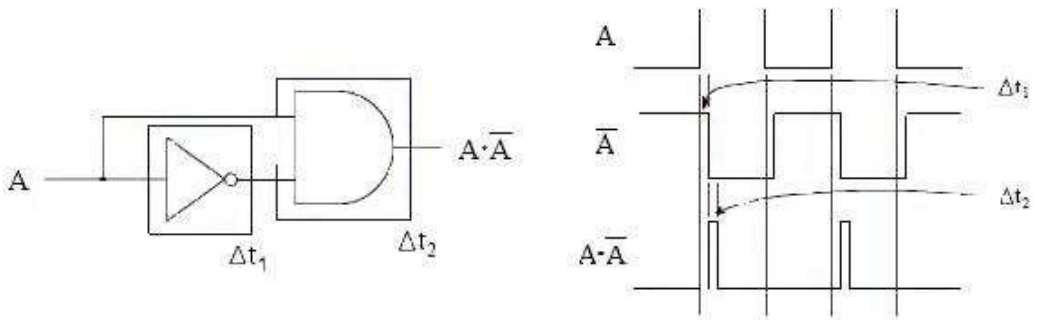
Düşük enerji tüketen devre çalışmasında işlem hacmini arttırmak kadar güç tüketimini düşürmek de önemli bir noktadır. Bu sebeple güç tasarrufu yöntemlerinden alt bölümde bahsedilmiştir.

4.2 Güç Tasarrufu Yöntemleri

Literatürde şifreleme algoritmalarının tasarımında kullanılan güç tasarrufu yöntemleri oldukça fazladır. Yine güç tasarrufu yöntemleri, bu çalışmada kullanılabilecek yöntemler ön planda tutularak incelenmiştir. Güç tasarrufu yöntemleri ile ilgili daha detaylı bilgiye [3,20–27] kaynaklarından erişilebilmektedir.

Güç tasarrufu yöntemlerinde genellikle amaç, devrelerin kapladığı alanı azaltmaktır. Böylelikle daha az devre elemanı kullanılarak devrenin harcadığı güç miktarı azaltılmaktadır. Örneğin DES algoritmasının tasarımında tek bir tur ve yazmaç kullanılarak bir döngü şeklinde 16 tur gerçeklenebilir. Böylelikle hem alandan kazanılmış hem de harcanan güç miktarı azaltılmış olur. Bölüm 5.2.3' te bu yöntem kullanılarak oluşturulan klasik yapı DES gerçeklenmesinde yöntemle ilgili detaylı bilgi verilecektir.

Güç tasarrufu için diğer bir yöntem ise kılçık azaltma yöntemidir. Giriş işareti değiştiği zaman çıkış işaretinin yeni değerini alabilmesi için zaman gerekir. Bu zaman aralığında çıkışta istenmeyen işaretler yani kılçık (glitch) oluşur, bu işaretler devrenin fazladan güç harcamasına neden olur. Şekil 4.1' de kılçık oluşturabilecek bir devre örneği verilmiştir [3]. Şekildeki çıkışta anlık olarak görünen lojik 1 çıkışları devredeki kılçık oluşumunu göstermektedir. Çıkışta olmaması gereken lojik 1 seviyelerinden dolayı, fazladan güç harcanır. Devrenin karmaşıklığına göre istenmeyen işaretlerden dolayı devrede gereksiz yere çok fazla dinamik güç harcaması yapılmasına neden olabilir, öyle ki tipik bir kombinezonsal devrede kılçıklardan kaynaklı güç harcaması %40' lara kadar çıkabilir [27]. Devrelerde oluşabilecek kılçıklardan kurtulmanın en kolay yolu işhattı yapısında gerçeklemedir [24–25]. İşhattı yapısı sayesinde bölüm 4.1' de de belirtildiği gibi devredeki kritik yollar kısalır ve devrenin çalışma frekansı artar. Ayrıca kılçık oluşumu engellendiğinden devrede harcanan güç miktarı da azalır. Bölüm 5.2.1' de DES algoritmasının işhattı yapısında gerçeklenmesi anlatılacak ve işhattı yapısı ile ilgili daha detaylı bilgi verilecektir.



Şekil 4.1 : Kılçık Örneği.

Güç tasarrufu için kullanılabilecek yöntemlerden biri de şifreleme algoritmasındaki tur sayısının azaltılması yani bir saat periyodunda birden fazla şifreleme yapılmasıdır [26]. Örneğin DES algoritmasındaki 16 turda gerçekleşen şifreleme, bir saat periyodunda iki şifreleme yapılarak 8 turda gerçeklenebilir. Böylelikle saat frekansı düşmesine rağmen şifreleme süresi kısaldığından iş hacminde fazla kayıp yaşanmaz. Ayrıca bir saat periyodu içinde gerçekleşen iki şifreleme için ortak yapılar kullanılarak alandan kazanç ve buna bağlı olarak da güç tasarrufu sağlanır. Bölüm 5.2.4 ve 5.2.5' te bu yöntemle gerçeklenen DES devreleri anlatılacak ve DES algoritmasının 8 turda gerçeklenmesi ile ilgili detaylı bilgi verilecektir.

Matematiksel iyileştirmeler de güç tasarrufu için kullanılan bir yöntemdir [3,25]. Matematiksel yöntemler sentez araçlarının tasarım üzerinde iyileştirme gerçekleştirmesi sırasında uygulanır. Bu yöntemde amaç tasarımı en basit şekilde gerçeklemek ve gereksiz eleman kullanımını engellemektir. Çalışma sırasında kullanılan Xilinx firmasına ait XST sentezleme aracında da bu iyileştirme özelliği bulunmaktadır. Bu yöntemde Bool Cebri işlemlerinden yararlanılarak ifadeler basitleştirilir. Örnek 4.1' de r = p+q sinyali için sentezleme sırasında matematiksel iyileştirmeler kullanıldığı için daha az kapı ihtiyacı duyulmakta ve böylece daha az güç harcaması yapılmaktadır [3,20].

Örnek 4.1

p = a.b+d

q = a+d.c

r = p+q = a+d

Güç tasarrufu için bu yöntemlerin dışında bu çalışmada kullanılmayan çok çeşitli yöntemler vardır. Burada bu çalışmada kullanılmayan bazı yöntemlerden bahsedilerek gelecek çalışmalara ışık tutulması hedeflenmiştir. Saat işaretini engelleme yönteminde devredeki yazmaçlara yeni veri gelmediğinde saat işareti yalıtılarak gereksiz güç harcaması engellenmiş olur. Yazmaçlar sadece yeni veri girişi olduğunda saat işaretine maruz kalır ve güç harcar. Böylelikle harcanan güç azalır [3]. DES algoritmasındaki yazmaçlar her saat işaretinde kullanıldığından bu yöntem DES algoritması için uygulanamamıştır.

Asenkron devre tasarımı da güç tasarrufu yöntemlerinden biridir. Asenkron tipi devre gerçeklemelerinde devrenin genel bir saat frekansı olmadığından senkron yapıdaki devrelere göre daha az güç tüketmesi beklenir [24]. Bölüm 4.1' de de belirtildiği gibi asenkron tasarım FPGA teknolojisine uygun değildir ve bu çalışmada kullanılmamıştır.

Giriş yalıtımı da güç tasarrufu için kullanılan yöntemlerdendir. Saat işareti engelleme ile aynı mantığa sahip olan bu yöntemde, çıkışı kullanılmayan kombinezonsal devreye giriş verisi uygulanmayarak gereksiz yere güç harcanması engellenmektedir [3]. Yine DES algoritmasına uygun olmadığından bu çalışmada kullanılmamıştır.

5. DÜŞÜK ENERJİLİ DES GERÇEKLEMESİ

DES blok şifreleme algoritmasının FPGA üzerinde düşük enerjili tasarımı yüksek lisans tez çalışmasının en temel bölümü olan düşük enerjili DES gerçeklemesi bölümünde ilk olarak literatürde bulunan önceki çalışmalardan bahsedilmiştir. Önceki çalışmalar bir çizelgede birleştirilmiş ve bu çalışmalar işlem hacmi, güç tüketim değeri ve bir biti şifrelemek için gerekli enerji miktarları açılarından karşılaştırılmıştır. Daha sonra en düşük enerji tüketen devreyi bulmak amacıyla gerçeklenen altı farklı DES yapısı detaylı olarak anlatılmıştır. Son olarak ise bütün DES gerçeklemeleri elde edilen alan, zaman, iş hacmi (throughput), güç ve enerji bilgileri doğrultusunda karşılaştırılmıştır ve en düşük enerji tüketen DES gerçeklemesi belirlenmiştir.

5.1 Önceki Çalışmalar

1977 yılında şifreleme standardı olarak kabul edilen ve yirmi yıl kadar standart olarak kullanılan DES blok şifreleme algoritmasının hem yazılımsal hem de donanımsal birçok gerçeklemesi literatürde mevcuttur [15–19,25,26,28–40]. Yapılan araştırma sonucu elde edilen DES gerçeklemeleri ile ilgili bilgiler çizelge 5.1' de verilmiştir. Çizelgede ilk sütunda devre ile ilgili bilgilerin elde edildiği kaynak numarası, ikinci sütunda gerçekleme sırasında kullanılan teknoloji verilmiştir. Üçüncü sütunda ise devrenin en yüksek çalışma frekansı (MHz – Mega-Hertz), dördüncü sütunda devrenin en yüksek işlem hacmi (Mb/s – Megabit/saniye), beşinci sütunda devrenin harcadığı güç miktarı (mW – mili-Watt) ve son sütunda devrenin bit başına harcadığı enerji yani bir bit şifreleme için gerekli enerji miktarı (nJ/bit – nanoJoule/bit) değerleri verilmiştir. Çizelgede bazı alanlar kaynaklarda ilgili değer verilmediğinden boş bırakılmıştır. Çizelge oluşturulurken işlem hacmi değeri için denklem 5.1' den ve bit başına enerji miktarı için denklem 5.2' den yararlanılmıştır. Denklem 5.1' deki adım sayısı, şifreleme işlemi için gerekli olan saat periyodu sayısını göstermektedir. Bu çalışma için en önemli değer olan bit başına harcanan enerji miktarı kaynakların sadece on tanesinden elde edilebilmiştir.

İşlem Hacmi = 64 / (Saat Periyodu * Adım Sayısı) **(5.1)**

Enerji / bit = Güç / İşlem Hacmi **(5.2)**

Çizelge 5.1 : Literatürdeki DES gerçeklemelerinin karşılaştırılması.

Kaynak	Teknoloji	Saat Frekansı (MHz)	İşlem Hacmi (Mb/s)	Güç (mW)	Enerji/bit (nJ/bit)
[28]	FPGA	10	26,7	-	-
[29]	FPGA	25,1	402,7	-	-
[30]	FPGA	47,7	3052	-	-
[31]	FPGA	168	10752	3200	0,297
[32]	FPGA	59,5	3808	-	-
[16]	FPGA	333	21300	-	-
[36]	FPGA	111,8	7160	-	-
[37]	FPGA	188,6	12000	-	-
[38]	FPGA	60	3870	-	-
[15]	FPGA	237	15100	-	-
[39]	FPGA	125	8000	148,6	0,018
[33]	Mikroişlemci	300	137	-	-
[34]	Mikroişlemci	-	-	-	2080
[25]	Mikroişlemci	8	0,079	55,53	703
[40]	Mikroişlemci	-	-	-	725
[17]	ASIC	105	9280	6500	0,700
[35]	ASIC	-	51,2	9,62	0,188
[19]	ASIC	13,56	102,4	15,97	0,156
[18]	ASIC	-	38780	420,8	0,011
[26]	ASIC	52,19	417,5	25,88	0,061

DES gerçeklemelerinde ilk zamanlarda yüksek hız ve yüksek işlem hacmi üzerinde yoğunlaşılmıştır. En eski gerçeklemelerden olan [28], tek tur ve yazmaç kullanarak döngü şeklindeki klasik DES gerçeklemesine bir örnektir. Yüksek işlem hacmi için gerçeklemelerin çoğunluğunda işhattı yapısı tercih edilmiştir [17,29-31,33,38]. Standart işhattı yapısında gerçeklenen bu çalışmalarda en yüksek işlem hacmi 10,7 Gb/s (Gigabit /saniye) ile [31]' e aittir. Standart işhattı yapısının özelleştirilmiş şekli

olan iç yazmaçlı işhattı yapısında da gerçeklemeler vardır [15,16,37,39]. Bu gerçeklemeler içinde en yüksek işlem hacmi 21,3 Gb/s ile [16]' ya aittir. Ek olarak işhattı yapısını farklı anahtar üretme yöntemleriyle gerçekleyen tasarımlar da mevcuttur [16,32,36]. İşhattı yapısından farklı olarak asenkron yapıda ASIC tasarımlar da vardır [18,19,35]. Çizelgedeki en yüksek işlem hacmine sahip devre de 38,7 Gb/s değeri ile [18]' deki asenkron ASIC gerçeklemedir.

Son yıllarda düşük güç tüketimi hedefi ile tasarlanan devreler de literatürde yer almaktadır. Güç tüketimi açısından devrelere bakılırsa, daha çok ASIC tasarımlarda hedef olarak düşük güç tüketimini seçildiği görülmektedir. Örneğin çizelgedeki en düşük güç tüketim değeri olan 9,62 mW ile [35] tasarımı asenkron yapıda bir ASIC tasarımdır. Bununla birlikte FPGA teknolojisi kullanılarak gerçeklenen devrelerin çoğunda harcanan güç miktarı kaynaklarda belirtilmemiştir. Bu sebeple var olan iki değer arasında en düşük güç tüketim değeri iç yazmaçlı 64 seviyeli işhattı yapısına sahip [39]' dur.

Bu çalışmada en çok üzerinde durulan bit başına harcanan yani bir biti şifrelemek için gerekli enerji miktarı açısından karşılaştırmaya gelinirse, en düşük enerji tüketimine sahip gerçekleme 11 pJ/bit (pikoJoule/bit) değeri ile asenkron bir ASIC tasarım olan [18]' dir. [18]' de hedef olarak yüksek işlem hacmi alınmıştır ve işlem hacminin çok yüksek değerlere çıkmasıyla bit başına harcanan enerji miktarı oldukça düşmüştür. FPGA teknolojisi ile elde edilen en düşük bit başına harcanan enerji değeri [39]' a aittir. [39]' da iç yazmaçlı 64 seviyeli işhattı yapısı kullanılmıştır ve 18 pJ/bit değerine ulaşılmıştır. Bu çalışmalarda düşük enerji tüketimi hedef olarak alınmamış, sadece harcanan güç ve işlem hacmi değerleri verilmiştir. [18]' de yüksek işlem hacmi, [39]' da ise düşük güç tüketimi amaçlanmıştır. Literatürde düşük enerji tüketimini amaçlayan DES gerçeklemesine rastlanmamıştır.

Son olarak mikroişlemci teknolojisi ile yapılan yazılımsal gerçeklemeler, işlem hacmi, harcanan güç ve bit başına enerji değerlerinde ASIC ve FPGA teknolojilerine göre oldukça gerilerde kalmıştır.

5.2 Farklı DES Yapıları

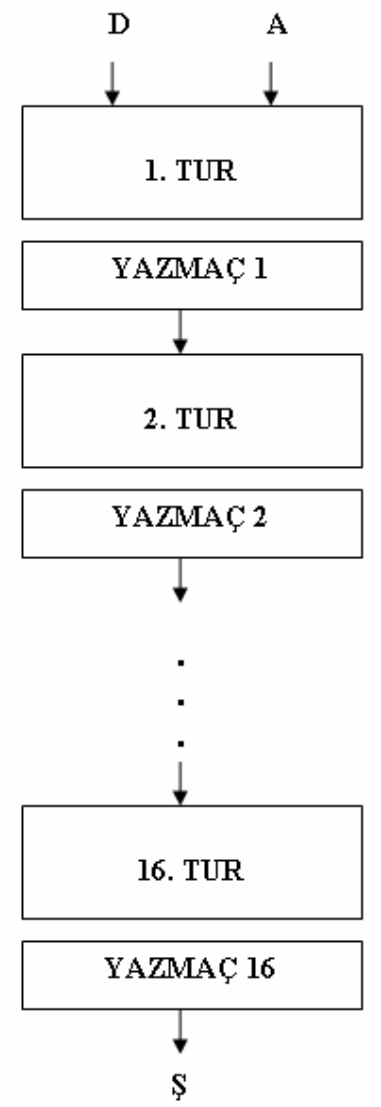
DES blok şifreleme algoritmasını farklı yapılarda gerçeklemek mümkündür ve bu yapıların her birinde farklı işlem hacmi, güç tüketim değeri ve bir biti şifrelemek için gerekli enerji değerleri çıkacağı açıktır. Çalışma sırasında DES blok şifreleme algoritması altı farklı yapıda gerçeklenmiş ve bu yapıların alan, zaman, işlem hacmi, güç tüketim değeri ve bir biti şifrelemek için gerekli enerji değerleri elde edilmiştir. Gerçeklemeler sırasında bölüm 4' te bahsedilen enerji tasarrufu yöntemlerinden yararlanılmıştır. Bu gerçeklemelerde bölüm 2.2' de bahsedilen Xilinx firmasına ait Spartan-3 XC3S5000 FPGA' sı kullanılmıştır. Gerçekleme aşamasında VHDL dili kullanılmış olup, FPGA' da sentezleme, yerleştirme ve yollandırma aşamaları için Xilinx ISE 9.2i programı kullanılmıştır. Ayrıca simülasyon aşamasında Modelsim 6.3c ve güç ölçümleri için de XPOWER aracı kullanılmıştır. Bununla birlikte bölüm 3.3' te anlatılan gerçekleme adımları ve bölüm 3.4' te anlatılan FPGA üzerinde gerçekleme adımları bütün yapılarda izlenmiştir.

Gerçeklenen farklı yapıların bazıları küçük alan ihtiyacı duyulduğunda kullanılmaktayken, bazıları ise yüksek işlem hacmi gerektiği zaman kullanılmaktadır. Bu çalışmada amaç düşük enerji tüketimi olduğundan, enerji açısından da gerçeklenen yapılar alt bölümlerde karşılaştırılmıştır.

5.2.1 İşhattı (pipeline) gerçeklemesi

Çalışmada gerçeklenen ilk yapı işhattı yapısıdır. İşhattı yapısının seçilmesinde sebep, bölüm 4' te bahsedilen enerji tasarrufu yöntemlerinden hem işlem hacmini arttırma hem de güç tasarrufu yöntemleri içinde bulunmasıdır. İşhattı yapısında gerçekleme şekil 5.1' de verilmiştir. Şekilden de görüldüğü gibi işhattı yapısında 16 adet DES tur fonksiyonunun arasında 16 adet yazmaç vardır. Yani yapıda 16 adet tur ve yazmaç bloğu art arda dizilmiştir. Şekilde D girişi düz metni, A girişi anahtar girişini ve Ş çıkışı da şifreli metin çıkışını temsil etmektedir.

İşhattı yapısında ilk şifreli metin 16 saat periyodu sonra oluşur ve yapısı gereği ilk şifreli metinden sonra her bir saat periyodunda bir şifreli metin oluşur. Yani iş hacminin hesaplandığı denklem 5.1' deki adım sayısı 1' dir. Böylelikle işhattı yapısından yüksek işlem hacmi ve dolayısıyla düşük enerji / bit oranı beklenmektedir.



Şekil 5.1 : İşhattı yapısında gerçekleme.

Bölüm 3.3' teki gerçekleme adımları takip edilerek ilk olarak işhattı yapısındaki gerçeklemenin VHDL kodu yazılmıştır. Daha sonra yazılan kodun doğru çalışıp çalışmadığını görmek amacıyla fonksiyonel benzetim (simülasyon) yapılmıştır. ModelSim programı yardımıyla yapılan simülasyon sonucu kodun önceden bilinen üç adet giriş çıkış değeri için doğru çalıştığı gözlenmiştir. Şekil 3.10' da işhattı yapısı için ModelSim programında elde edilen simülasyon sonucu verilmiştir. Yazılan kodun doğru çalıştığına emin olunduktan sonra, tasarım sürecinde bir sonraki adım olan FPGA üzerinde gerçeklemeye geçilmiştir.

Bölüm 3.4' teki FPGA üzerinde gerçekleme adımları takip edilerek ilk olarak Xilinx ISE 9.2i programında işhattı yapısı için yazılan VHDL kodu sentezlenmiştir. Yazılan kodun çalışma amacı yüksek işlem hacmi olduğundan, sentezleme seçeneklerinden optimizasyon hedefi (optimization goal) olarak hız seçilmiştir. Ayrıca optimizasyon girişimi (optimization effort) normalden yükseğe çekilerek FPGA üzerinde en iyi gerçekleme sağlanmıştır. Sentezleme sonucunda devre ile ilgili zaman ve alan bilgileri elde edilmiştir. Fakat bölüm 3.4' te de söylendiği gibi bu değerler sentezleme sırasındaki tahmini değerlerdir, gerçek değerler için yerleştirme ve yollandırma (place & route) sonrasındaki alan ve zaman bilgilerine bakılmıştır.

Sentezleme işleminden sonra işhattı yapısı için FPGA üzerinde dönüştürme (translate), eşleştirme (map), yerleştirme ve yollandırma (place & route) işlemleri gerçekleştirilmiştir. Ayrıca kullanıcı kısıtlamaları olarak saat frekansı verilmiş ve devrenin çalışabileceği en yüksek saat frekansı iterasyonla bulunmuştur. Bu işlemlerin sonucunda oluşan zamanlama ve eşleştirme raporlarından devrenin gerçek alan ve zaman bilgileri elde edilmiştir. İşhattı gerçeklemesi için elde edilen alan ve zaman bilgileri şöyledir:

Design Summary

Logic Utilization:
  Number of Slice Flip Flops: 928 out of 66,560 1%
  Number of 4 input LUTs: 4,125 out of 66,560 6%
Logic Distribution:
  Number of occupied Slices: 2,320 out of 33,280 6%

Timing summary:

Design statistics:
  Minimum period: 9.796ns (Maximum frequency: 102.082MHz)

İşhattı yapısındaki devrenin harcadığı güç ve enerji bilgilerini elde edebilmek için yerleştirme ve yollandırma sonrası simülasyon (post-route simulation) yapılmıştır. Bu simülasyon sırasında devreye MATLAB programı tarafından oluşturulmuş rastgele girişler uygulanmış ve devre mümkün olan en yüksek saat frekansında çalıştırılarak simülasyon sonuçları elde edilmiştir. Burada dikkat edilmesi gereken nokta, yerleştirme ve yollandırma sonrası simülasyonda kullanılacak saat frekansıdır. Yerleştirme ve yollandırma sonucu elde edilen rapordaki en yüksek saat frekansında devre içindeki kapı gecikmelerine bağlı olarak devre doğru çıkışları üretemeyebilir. Çıkışların doğruluğu kontrol edilerek, mümkün olan en yüksek saat frekansı iterasyonla bulunmalıdır. İşhattı yapısındaki devre için saat frekansı 102 MHz verilerek yapılan yerleştirme ve yollandırma sonrası simülasyon sonucu çıkışlar yanlış olmaktadır. Bu sebeple saat frekansı iterasyonla düşürülerek 83 MHz verilmiş ve yapılan simülasyon doğru sonuçları vermiştir. Simülasyon sırasında güç analizi işleminde kullanılmak üzere .vcd uzantılı simülasyon dosyası da oluşturulmuştur. Son olarak elde edilen simülasyon dosyası kullanılarak XPOWER güç ölçüm aracı yardımıyla devrenin harcadığı güç bilgisi elde edilmiştir. İşhattı gerçeklemesi için güç değeri 48,7 mW (mili-watt) çıkmıştır. Böylelikle yazılan VHDL kodunun FPGA üzerinde gerçeklenmesi aşaması tamamlanmıştır.

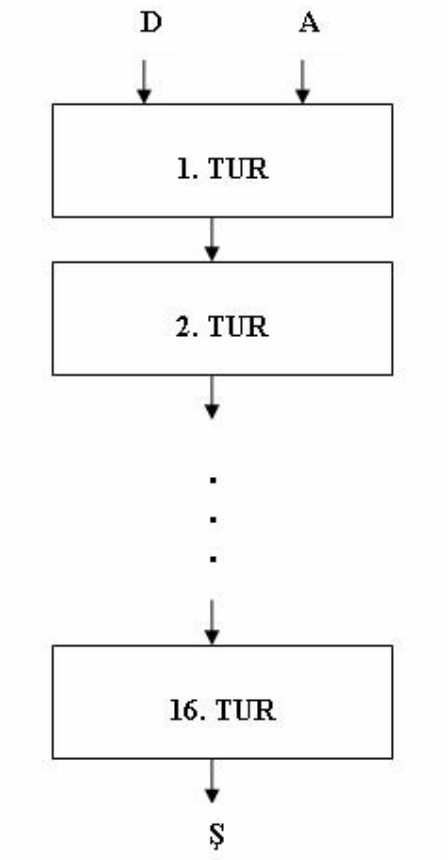
İşhattı yapısındaki devrenin en yüksek işlem hacmi değeri denklem 5.1' den yararlanılarak 5,33 Gb/s olarak bulunmuştur. Ayrıca denklem 5.2' den yararlanılarak bir biti şifrelemek için gerekli enerji miktarı 9,1 pJ/bit olarak bulunmuştur. Bununla birlikte devrenin 64 bitlik bir bloğu şifrelemek için harcadığı enerji miktarı da denklem 5.3' ten yararlanılarak 584,4 pJ olarak bulunmuştur.

$$\text{Enerji} = \text{Güç} * (\text{Saat Periyodu} * \text{Adım Sayısı}) \quad \textbf{(5.3)}$$

Elde edilen işlem hacmi ve bir biti şifrelemek için gerekli enerji miktarı değerleri alt bölümlerde yapılacak olan DES gerçeklemelerinin karşılaştırılması için kullanılacaktır. İlk bakışta çizelge 5.1' de verilen literatürdeki DES gerçeklemeleri arasında en yüksek işlem hacmine sahip olmamasına rağmen, harcadığı güç miktarının az olması sayesinde bir biti şifrelemek için gerekli enerji miktarı değeri en düşük olan devre olmuştur. Bu sonuçta işhattı yapısının hem yüksek işlem hacmi hem de düşük güç tüketimi sağlamasının büyük payı vardır.

5.2.2 Yazmaçsız gerçekleme

Çalışmada gerçeklenen ikinci yapı yazmaçsız yani kombinezonsal yapıdır. Yazmaçsız yapı, bölüm 4' te bahsedilen enerji tasarrufu yöntemlerinin içinde olmamasına rağmen devrelerdeki yazmaçlar olmadan tüketilen enerji miktarını incelemek için gerçeklenmiştir. Yazmaçsız yapıda gerçekleme şekil 5.2' de verilmiştir. Şekilden de görüldüğü gibi yazmaçsız yapıda 16 adet DES tur fonksiyonu art arda dizilmiştir. Şekilde D girişi düz metni, A girişi anahtar girişini ve Ş çıkışı da şifreli metin çıkışını temsil etmektedir.



Şekil 5.2 : Yazmaçsız yapıda gerçekleme.

Yazmaçsız yapıda ilk şifreli metin bir saat periyodu sonra oluşur ve yapısı gereği ilk şifreli metinden sonra her bir saat periyodunda bir şifreli metin oluşur. Yani denklem 5.1' deki adım sayısı 1' dir. Ancak yazmaçsız yapıda kritik yol çok uzundur ve buna bağlı olarak devrede düşük işlem hacmi ve dolayısıyla yüksek enerji / bit oranı beklenmektedir.

Bölüm 3.3' teki gerçekleme adımları takip edilerek ilk olarak yazmaçsız yapıdaki gerçeklemenin VHDL kodu yazılmıştır. Daha sonra yazılan kodun doğru çalışıp çalışmadığını görmek amacıyla fonksiyonel benzetim (simülasyon) yapılmıştır. ModelSim programı yardımıyla yapılan simülasyon sonucu kodun önceden bilinen üç adet giriş çıkış değeri için doğru çalıştığı gözlenmiştir. Yazılan kodun doğru çalıştığına emin olunduktan sonra, tasarım sürecinde bir sonraki adım olan FPGA üzerinde gerçeklemeye geçilmiştir.

Bölüm 3.4' teki FPGA üzerinde gerçekleme adımları takip edilerek ilk olarak Xilinx ISE 9.2i programında yazmaçsız yapı için yazılan VHDL kodu sentezlenmiştir. Yazılan kodun çalışma amacı yüksek işlem hacmi olduğundan, sentezleme seçeneklerinden optimizasyon hedefi olarak hız seçilmiştir. Ayrıca optimizasyon girişimi normalden yükseğe çekilerek FPGA üzerinde en iyi gerçekleme sağlanmıştır.

Sentezleme işleminden sonra yazmaçsız yapı için FPGA üzerinde dönüştürme, eşleştirme, yerleştirme ve yollandırma işlemleri gerçekleştirilmiştir. Bu yapıda yazmaç bulunmadığından kullanıcı kısıtlamaları olarak saat frekansı verilememiştir. Saat frekansı olarak elde edilen kritik yol gecikmesi kullanılmıştır. Bu işlemlerin sonucunda oluşan zamanlama ve eşleştirme raporlarından devrenin gerçek alan ve zaman bilgileri elde edilmiştir. Yazmaçsız gerçekleme için elde edilen alan ve zaman bilgileri şöyledir:

Device utilization summary:

Logic Utilization:
  Number of 4 input LUTs: 3,040 out of 66,560 4%
Logic Distribution:
  Number of occupied Slices: 1,540 out of 33,280 4%

Timing Summary:

Speed Grade: -5

  Minimum period: No path found
  Minimum input arrival time before clock: No path found
  Maximum output required time after clock: No path found
  Maximum combinational path delay: 74.739ns

Yazmaçsız yapıdaki devrenin harcadığı güç ve enerji bilgilerini elde edebilmek için yerleştirme ve yollandırma sonrası simülasyon yapılmıştır. Bu simülasyon sırasında yine devreye MATLAB programı tarafından oluşturulmuş rastgele girişler uygulanmış ve devre mümkün olan en yüksek saat frekansında çalıştırılarak simülasyon sonuçları elde edilmiştir. Yazmaçsız yapıdaki devre için saat frekansı 13,3 MHz' den iterasyonla düşürülmüş ve yapılan simülasyon 10,6 MHz' de doğru sonuçları vermiştir. Simülasyon sırasında güç analizi işleminde kullanılmak üzere .vcd uzantılı simülasyon dosyası da oluşturulmuştur. Son olarak elde edilen simülasyon dosyası kullanılarak XPOWER güç ölçüm aracı yardımıyla devrenin harcadığı güç bilgisi elde edilmiştir. Yazmaçsız gerçekleme için güç değeri 597,8 mW çıkmıştır. Böylelikle yazılan VHDL kodunun FPGA üzerinde gerçeklenmesi aşaması da tamamlanmıştır.

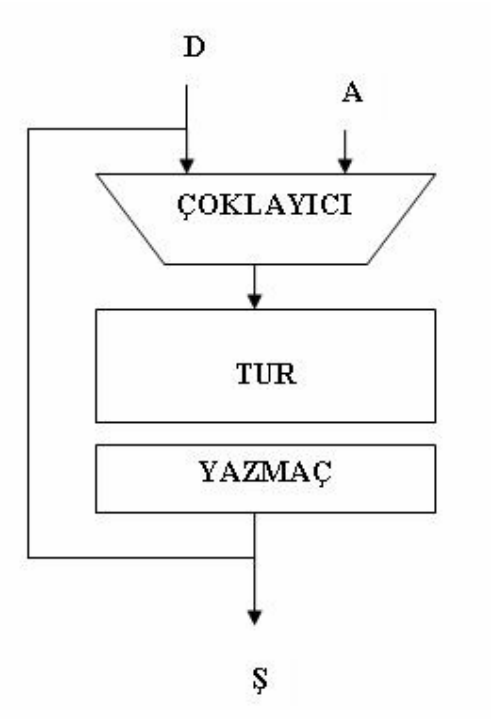
Yazmaçsız yapıdaki devrenin en yüksek işlem hacmi değeri denklem 5.1' den yararlanılarak 678,4 Mb/s olarak bulunmuştur. Ayrıca denklem 5.2' den yararlanılarak bir biti şifrelemek için gerekli enerji miktarı 880,8 pJ/bit olarak bulunmuştur. Bununla birlikte devrenin 64 bitlik bir bloğu şifrelemek için harcadığı enerji miktarı da denklem 5.3' ten yararlanılarak 56,372 nJ olarak bulunmuştur.

Elde edilen işlem hacmi ve bir biti şifrelemek için gerekli enerji miktarı değerleri açısından çizelge 5.1' de verilen literatürdeki DES gerçeklemeleri arasında oldukça gerilerde kalmaktadır. Ayrıca bölüm 5.2.1' de gerçeklenen işhattı yapısına göre de işlem hacmi, harcanan güç ve enerji miktarı ile bir biti şifrelemek için kullanılan enerjide daha kötüdür. Yazmaçsız yapı sadece alan açısından işhattı yapısından daha iyi sonuç vermiştir. Bu sonuçta yazmaçsız yapıda kritik yolun çok uzun olmasının büyük payı vardır ve beklendiği gibi enerji tasarruflu bir devre değildir.

5.2.3 Klasik yapı gerçeklemesi

Çalışmada gerçeklenen üçüncü yapı klasik yapıdır. Klasik yapının seçilmesinde sebep, bölüm 4.2' de bahsedilen güç tasarrufu yöntemlerinden biri olmasıdır. Klasik yapıda gerçekleme şekil 5.3' te verilmiştir. Şekilden de görüldüğü gibi klasik yapıda 1 adet DES tur fonksiyonu ve 1 adet yazmaç vardır. Yapıda 16 tur döngü şeklinde gerçekleşmektedir. Şekilde D girişi düz metni, A girişi anahtar girişini ve Ş çıkışı da şifreli metin çıkışını temsil etmektedir. İlk saat darbesinde çoklayıcı elemanı tarafından D girişi seçilmekte, sonraki 15 saat darbesi boyunca ise yazmaç çıkışından gelen giriş seçilmektedir. Böylelikle döngü şeklinde şifreleme sağlanmaktadır.

Klasik yapıda ilk şifreli metin 16 saat periyodu sonra oluşur ve yapısı gereği ilk şifreli metinden sonra her 16 saat periyodunda bir şifreli metin oluşur. Yani denklem 5.1' deki adım sayısı 16' dır. Alandan tasarruf sağlayan klasik yapıdan düşük işlem hacmi ve düşük güç tüketimi beklenmektedir.



Şekil 5.3 : Klasik yapıda gerçekleme.

Bölüm 3.3' teki gerçekleme adımları takip edilerek ilk olarak klasik yapıdaki gerçeklemenin VHDL kodu yazılmıştır. Daha sonra yazılan kodun doğru çalışıp çalışmadığını görmek amacıyla fonksiyonel benzetim (simülasyon) yapılmıştır. ModelSim programı yardımıyla yapılan simülasyon sonucu kodun önceden bilinen üç adet giriş çıkış değeri için doğru çalıştığı gözlenmiştir. Yazılan kodun doğru çalıştığına emin olunduktan sonra, tasarım sürecinde bir sonraki adım olan FPGA üzerinde gerçeklemeye geçilmiştir.

Bölüm 3.4' teki FPGA üzerinde gerçekleme adımları takip edilerek ilk olarak Xilinx ISE 9.2i programında klasik yapı için yazılan VHDL kodu sentezlenmiştir. Yazılan kodun çalışma amacı düşük güç tüketimi ve küçük alan olduğundan, sentezleme seçeneklerinden optimizasyon hedefi olarak alan seçilmiştir. Ayrıca optimizasyon girişimi normalden yükseğe çekilerek FPGA üzerinde en iyi gerçekleme sağlanmıştır. Sentezleme sonucunda devre ile ilgili tahmini zaman ve alan bilgileri elde edilmiştir.

Sentezleme işleminden sonra klasik yapı için FPGA üzerinde dönüştürme, eşleştirme, yerleştirme ve yollandırma işlemleri gerçekleştirilmiştir. Ayrıca kullanıcı kısıtlamaları olarak saat frekansı verilmiş ve devrenin çalışabileceği en yüksek saat frekansı iterasyonla bulunmuştur. Bu işlemlerin sonucunda oluşan zamanlama ve eşleştirme raporlarından devrenin gerçek alan ve zaman bilgileri elde edilmiştir. Klasik gerçekleme için elde edilen alan ve zaman bilgileri şöyledir:

Device utilization summary:

Logic Utilization:
  Number of Slice Flip Flops: 69 out of 66,560 1%
  Number of 4 input LUTs: 587 out of 66,560 1%
Logic Distribution:
  Number of occupied Slices: 309 out of 33,280 1%

Timing Summary:

Design statistics:
  Minimum period: 14.187ns (Maximum frequency: 70.487MHz)

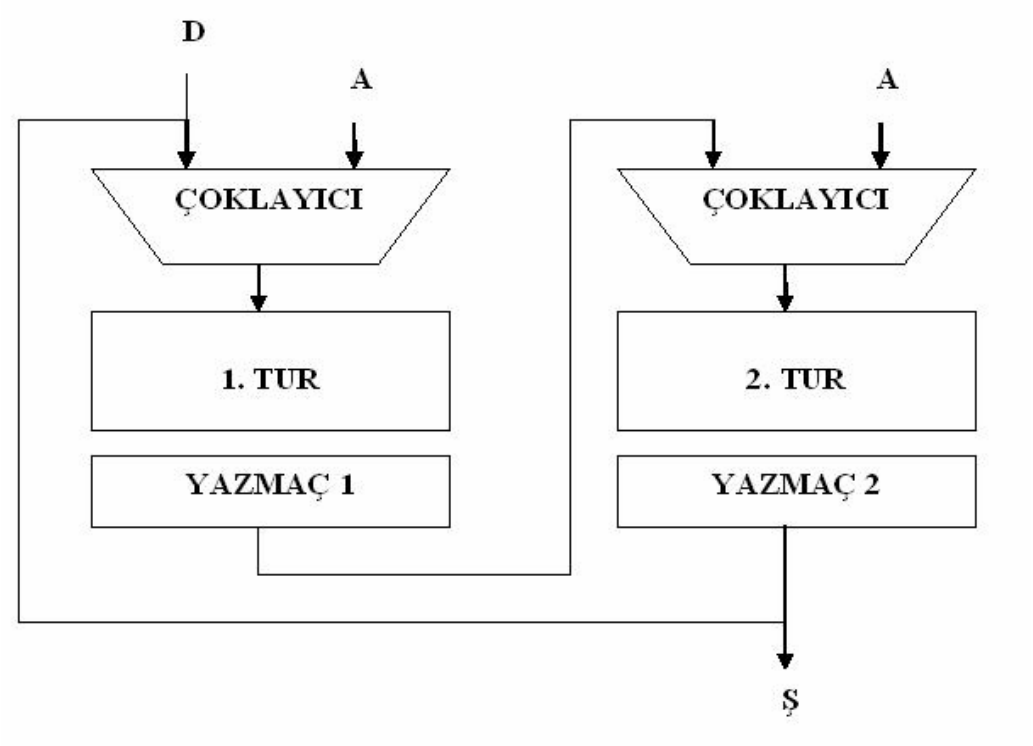
Klasik yapıdaki devrenin harcadığı güç ve enerji bilgilerini elde edebilmek için yerleştirme ve yollandırma sonrası simülasyon yapılmıştır. Bu simülasyon sırasında yine devreye MATLAB programı tarafından oluşturulmuş rastgele girişler uygulanmış ve devre mümkün olan en yüksek saat frekansında çalıştırılarak simülasyon sonuçları elde edilmiştir. Klasik yapıdaki devre için saat frekansı 70,4 MHz' de yapılan simülasyon doğru sonuçları vermiştir. Simülasyon sırasında güç analizi işleminde kullanılmak üzere .vcd uzantılı simülasyon dosyası da oluşturulmuştur. Son olarak elde edilen simülasyon dosyası kullanılarak XPOWER güç ölçüm aracı yardımıyla devrenin harcadığı güç bilgisi elde edilmiştir. Klasik gerçekleme için güç değeri 17,3 mW çıkmıştır. Böylelikle yazılan VHDL kodunun FPGA üzerinde gerçeklenmesi aşaması da tamamlanmıştır.

Klasik yapıdaki devrenin en yüksek işlem hacmi değeri denklem 5.1' den yararlanılarak 285,7 Mb/s olarak bulunmuştur. Ayrıca denklem 5.2' den yararlanılarak bir biti şifrelemek için gerekli enerji miktarı 60,5 pJ/bit olarak bulunmuştur. Bununla birlikte devrenin 64 bitlik bir bloğu şifrelemek için harcadığı enerji miktarı da denklem 5.3' ten yararlanılarak 3,875 nJ olarak bulunmuştur.

Elde edilen işlem hacmi değeri açısından çizelge 5.1' de verilen literatürdeki DES gerçeklemeleri arasında oldukça gerilerde kalmaktadır. Fakat harcanan güç miktarı ve bir biti şifrelemek için harcanan enerji miktarı açısından oldukça iyi bir yerdedir. Ayrıca bölüm 5.2.1' de gerçeklenen işhattı yapısına göre de işlem hacmi ve bir biti şifrelemek için kullanılan enerji değeri daha kötü, fakat harcanan güç ve kapladığı alan bakımından daha iyidir. Bu sonuçta klasik yapıda güç tasarrufu yöntemi olarak alandan tasarruf için şifrelemenin döngü şekline gerçekleştirilmesinin büyük payı vardır ve beklendiği gibi işhattı yapısına kıyasla güç tasarrufu sağlamasına rağmen işlem hacminin çok düşük olması sebebiyle enerji tasarrufu sağlayamamıştır.

5.2.4 Sekiz turda iki S – kutusu ile gerçekleme

Çalışmada gerçeklenen dördüncü yapı sekiz turda iki S-kutusu ile gerçekleme yapısıdır. Sekiz turdaki yapının seçilmesinde sebep, bölüm 4.2' de bahsedilen güç tasarrufu yöntemlerinden biri olmasıdır. DES algoritmasındaki 16 turda gerçekleşen şifreleme, bir saat periyodunda iki şifreleme yapılarak 8 turda gerçeklenmiştir. Böylelikle saat frekansı düşmesine rağmen şifreleme süresi kısaldığından iş hacminde fazla kayıp yaşanmaması beklenmektedir. Sekiz turda iki S-kutusu ile gerçekleme şekil 5.4' te verilmiştir. Şekilden de görüldüğü gibi bu yapıda 2 adet DES tur fonksiyonu ve 2 adet yazmaç vardır. Yapıda 16 tur döngü şeklinde 8 saat periyodunda gerçekleşmektedir. Şekilde D girişi düz metni, A girişi anahtar girişini ve Ş çıkışı da şifreli metin çıkışını temsil etmektedir. İlk saat darbesinde çoklayıcı elemanı tarafından D girişi seçilmekte, sonraki 7 saat darbesi boyunca ise ikinci yazmaç çıkışından gelen giriş seçilmektedir. Böylelikle döngü şeklinde şifreleme sağlanmaktadır.



Şekil 5.4 : Sekiz turda iki S-kutusu ile gerçekleme.

Sekiz turda iki S-kutusu ile gerçekleme yapısında ilk şifreli metin 8 saat periyodu sonra oluşur ve yapısı gereği ilk şifreli metinden sonra her 8 saat periyodunda bir şifreli metin oluşur. Yani işlem hacminin elde edildiği denklem 5.1' deki adım sayısı 8' dir. Şifreleme süresini klasik yapıya göre kısaltan sekiz turda iki S-kutusu ile gerçeklemede düşük enerji ve düşük güç tüketimi beklenmektedir.

Bölüm 3.3' teki gerçekleme adımları takip edilerek ilk olarak sekiz turda iki S-kutusu ile gerçeklemenin VHDL kodu yazılmıştır. Bir turda iki şifreleme yapılması gerektiğinden iki ayrı döngü tanımlanarak saat işaretinin yükselen ve düşen kenarlarında işlemler yapılmıştır. VHDL' de bir döngü içerisinde iki farklı değer bir saat periyodunda aynı yazmaca yazılamadığından iki ayrı döngü kullanılması zorunlu hale gelmiştir. Daha sonra yazılan kodun doğru çalışıp çalışmadığını görmek amacıyla fonksiyonel benzetim (simülasyon) yapılmıştır. ModelSim programı yardımıyla yapılan simülasyon sonucu kodun önceden bilinen üç adet giriş çıkış değeri için doğru çalıştığı gözlenmiştir. Yazılan kodun doğru çalıştığına emin olunduktan sonra, tasarım sürecinde bir sonraki adım olan FPGA üzerinde gerçeklemeye geçilmiştir.

Bölüm 3.4' teki FPGA üzerinde gerçekleme adımları takip edilerek ilk olarak Xilinx ISE 9.2i programında sekiz turda iki S-kutusu ile gerçekleme için yazılan VHDL kodu sentezlenmiştir. Yazılan kodun çalışma amacı düşük güç tüketimi ve küçük alan olduğundan, sentezleme seçeneklerinden optimizasyon hedefi olarak alan seçilmiştir. Ayrıca optimizasyon girişimi normalden yükseğe çekilerek FPGA üzerinde en iyi gerçekleme sağlanmıştır. Sentezleme sonucunda devre ile ilgili tahmini zaman ve alan bilgileri elde edilmiştir.

Sentezleme işleminden sonra sekiz turda iki S-kutusu ile gerçekleme yapısı için FPGA üzerinde dönüştürme, eşleştirme, yerleştirme ve yollandırma işlemleri gerçekleştirilmiştir. Ayrıca kullanıcı kısıtlamaları olarak saat frekansı verilmiş ve devrenin çalışabileceği en yüksek saat frekansı iterasyonla bulunmuştur. Bu işlemlerin sonucunda oluşan zamanlama ve eşleştirme raporlarından devrenin gerçek alan ve zaman bilgileri elde edilmiştir. Sekiz turda iki S-kutusu ile gerçekleme yapısı için elde edilen alan ve zaman bilgileri şöyledir:

Device utilization summary:

Logic Utilization:
  Number of Slice Flip Flops: 71 out of 66,560 1%
  Number of 4 input LUTs: 717 out of 66,560 1%
Logic Distribution:
  Number of occupied Slices: 367 out of 33,280 1%

Timing Summary:

Design statistics:
  Minimum period: 20.552ns (Maximum frequency: 48.657MHz)

Sekiz turda iki S-kutusu ile gerçekleme yapısındaki devrenin harcadığı güç ve enerji bilgilerini elde edebilmek için yerleştirme ve yollandırma sonrası simülasyon yapılmıştır. Bu simülasyon sırasında yine devreye MATLAB programı tarafından oluşturulmuş rastgele girişler uygulanmış ve devre mümkün olan en yüksek saat frekansında çalıştırılarak simülasyon sonuçları elde edilmiştir. Sekiz turda iki S-kutusu ile gerçekleme yapısındaki devre için saat frekansı 38,4 MHz' de yapılan simülasyon doğru sonuçları vermiştir. Simülasyon sırasında güç analizi işleminde kullanılmak üzere .vcd uzantılı simülasyon dosyası da oluşturulmuştur. Son olarak elde edilen simülasyon dosyası kullanılarak XPOWER güç ölçüm aracı yardımıyla devrenin harcadığı güç bilgisi elde edilmiştir. Sekiz turda iki S-kutusu ile gerçekleme için güç değeri 15,1 mW çıkmıştır. Böylelikle yazılan VHDL kodunun FPGA üzerinde gerçeklenmesi aşaması da tamamlanmıştır.

Sekiz turda iki S-kutusu ile gerçekleme yapısındaki devrenin en yüksek işlem hacmi değeri denklem 5.1' den yararlanılarak 273,5 Mb/s olarak bulunmuştur. Ayrıca denklem 5.2' den yararlanılarak bir biti şifrelemek için gerekli enerji miktarı 55,2 pJ/bit olarak bulunmuştur. Bununla birlikte devrenin 64 bitlik bir bloğu şifrelemek için harcadığı enerji miktarı da denklem 5.3' ten yararlanılarak 3,533 nJ olarak bulunmuştur.

Elde edilen işlem hacmi değeri açısından çizelge 5.1' de verilen literatürdeki DES gerçeklemeleri arasında oldukça gerilerde kalmaktadır. Fakat harcanan güç miktarı ve bir biti şifrelemek için harcanan enerji miktarı açısından oldukça iyi bir yerdedir. Ayrıca bölüm 5.2.1' de gerçeklenen işhattı yapısına göre de işlem hacmi ve bir biti şifrelemek için kullanılan enerji değeri daha kötü, fakat harcanan güç ve kapladığı alan bakımından daha iyidir. Elde edilen sonuçlara göre güç ve enerji tüketiminde klasik yapıya göre az da olsa iyileştirme sağlanmıştır. Bu sonuçta sekiz turda iki S-kutusu ile gerçekleme yapısında güç tasarrufu yöntemi olarak tur sayısını azaltmanın yani bir turda iki şifreleme yapılmasının büyük payı vardır. Beklendiği gibi işhattı yapısına kıyasla güç tasarrufu sağlamasına rağmen işlem hacminin çok düşük olması sebebiyle enerji tasarrufu sağlayamamıştır. Hala en düşük enerji tüketimine sahip devre işhattı yapısıdır.

5.2.5 Sekiz turda tek S – kutusu ile gerçekleme

Çalışmada gerçeklenen beşinci yapı sekiz turda tek S-kutusu ile gerçekleme yapısıdır. Bu yapı, düşük güç tüketimine örnek olarak [26]' dan alınmıştır. Bölüm 5.2.4' te anlatılan sekiz turda iki S-kutusu ile gerçeklemeden tek farkı bir turda yapılan iki şifrelemede aynı S-kutusunun kullanılmasıdır. Bunda amaç, alandan tasarruf sağlayarak güç ve enerji tüketimini azaltmak yani bir önceki devreyi iyileştirmektir. Yine DES algoritmasındaki 16 turda gerçekleşen şifreleme, bir saat periyodunda iki şifreleme yapılarak 8 turda gerçeklenmiştir. Böylelikle saat frekansı düşmesine rağmen şifreleme süresi kısaldığından iş hacminde fazla kayıp yaşanmaması beklenmektedir.

Sekiz turda tek S-kutusu ile gerçekleme yapısında ilk şifreli metin 8 saat periyodu sonra oluşur ve yapısı gereği ilk şifreli metinden sonra her 8 saat periyodunda bir şifreli metin oluşur. Yani işlem hacminin elde edildiği denklem 5.1' deki adım sayısı 8' dir. Şifreleme süresini klasik yapıya göre kısaltan sekiz turda tek S-kutusu ile gerçeklemede düşük enerji ve düşük güç tüketimi beklenmektedir. Ayrıca iki S-kutusu ile gerçeklemeye göre de az da olsa güç ve enerji tüketiminde iyileşme beklenmektedir çünkü daha az alan kaplaması beklenmektedir.

Bölüm 3.3' teki gerçekleme adımları takip edilerek ilk olarak sekiz turda tek S-kutusu ile gerçeklemenin VHDL kodu yazılmıştır. Bir önceki yapıdan farklı olarak sadece S-kutularının girişinde çoklayıcı yardımıyla yükselen veya düşen kenar işlemi olduğunu belirten bir seçme işlemi yapılmıştır. Daha sonra yazılan kodun doğru çalışıp çalışmadığını görmek amacıyla fonksiyonel benzetim (simülasyon) yapılmıştır. ModelSim programı yardımıyla yapılan simülasyon sonucu kodun önceden bilinen üç adet giriş çıkış değeri için doğru çalıştığı gözlenmiştir. Yazılan kodun doğru çalıştığına emin olunduktan sonra, tasarım sürecinde bir sonraki adım olan FPGA üzerinde gerçeklemeye geçilmiştir.

Bölüm 3.4' teki FPGA üzerinde gerçekleme adımları takip edilerek ilk olarak Xilinx ISE 9.2i programında sekiz turda tek S-kutusu ile gerçekleme için yazılan VHDL kodu sentezlenmiştir. Yazılan kodun çalışma amacı düşük güç tüketimi ve küçük alan olduğundan, sentezleme seçeneklerinden optimizasyon hedefi olarak alan seçilmiştir. Ayrıca optimizasyon girişimi normalden yükseğe çekilerek FPGA üzerinde en iyi gerçekleme sağlanmıştır. Sentezleme sonucunda devre ile ilgili tahmini zaman ve alan bilgileri elde edilmiştir.

Sentezleme işleminden sonra sekiz turda tek S-kutusu ile gerçekleme yapısı için FPGA üzerinde dönüştürme, eşleştirme, yerleştirme ve yollandırma işlemleri gerçekleştirilmiştir. Ayrıca kullanıcı kısıtlamaları olarak saat frekansı verilmiş ve devrenin çalışabileceği en yüksek saat frekansı iterasyonla bulunmuştur. Bu işlemlerin sonucunda oluşan zamanlama ve eşleştirme raporlarından devrenin gerçek alan ve zaman bilgileri elde edilmiştir. Sekiz turda tek S-kutusu ile gerçekleme yapısı için elde edilen alan ve zaman bilgileri şöyledir:

Device utilization summary:

Logic Utilization:
  Number of Slice Flip Flops: 71 out of 66,560 1%
  Number of 4 input LUTs: 652 out of 66,560 1%
Logic Distribution:
  Number of occupied Slices: 341 out of 33,280 1%

Timing Summary:

Design statistics:
  Minimum period: 21.384ns (Maximum frequency: 46.764MHz)

Sekiz turda tek S-kutusu ile gerçekleme yapısındaki devrenin harcadığı güç ve enerji bilgilerini elde edebilmek için yerleştirme ve yollandırma sonrası simülasyon yapılmıştır. Bu simülasyon sırasında yine devreye MATLAB programı tarafından oluşturulmuş rastgele girişler uygulanmış ve devre mümkün olan en yüksek saat frekansında çalıştırılarak simülasyon sonuçları elde edilmiştir. Sekiz turda tek S-kutusu ile gerçekleme yapısındaki devre için saat frekansı 37 MHz' de yapılan simülasyon doğru sonuçları vermiştir. Simülasyon sırasında güç analizi işleminde kullanılmak üzere .vcd uzantılı simülasyon dosyası da oluşturulmuştur. Son olarak elde edilen simülasyon dosyası kullanılarak XPOWER güç ölçüm aracı yardımıyla devrenin harcadığı güç bilgisi elde edilmiştir. Sekiz turda tek S-kutusu ile gerçekleme için güç değeri 14 mW çıkmıştır. Böylelikle yazılan VHDL kodunun FPGA üzerinde gerçeklenmesi aşaması da tamamlanmıştır.

Sekiz turda tek S-kutusu ile gerçekleme yapısındaki devrenin en yüksek işlem hacmi değeri denklem 5.1' den yararlanılarak 263,3 Mb/s olarak bulunmuştur. Ayrıca denklem 5.2' den yararlanılarak bir biti şifrelemek için gerekli enerji miktarı 53,1 pJ/bit olarak bulunmuştur. Bununla birlikte devrenin 64 bitlik bir bloğu şifrelemek için harcadığı enerji miktarı da denklem 5.3' ten yararlanılarak 3,402 nJ olarak bulunmuştur.

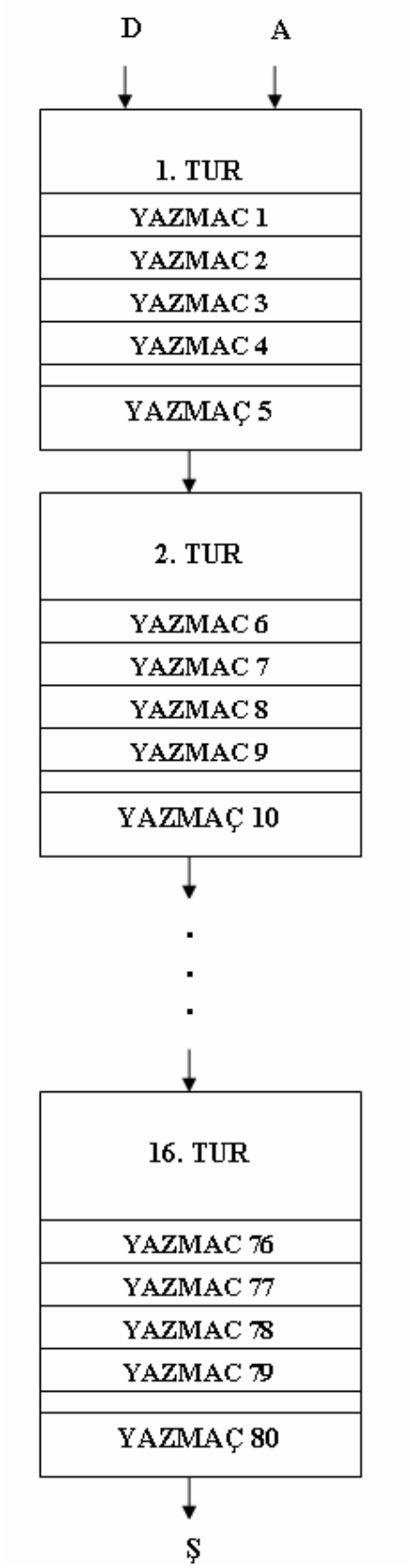
Elde edilen sonuçlara göre güç ve enerji tüketiminde sekiz turda iki S-kutusu ile gerçeklemeye göre az da olsa iyileştirme sağlanmıştır. Bu sonuçta sekiz turda tek S-kutusu ile gerçekleme yapısında güç tasarrufu yöntemi olarak alandan tasarrufun büyük payı vardır. Beklendiği gibi işhattı yapısına kıyasla güç tasarrufu sağlamasına rağmen işlem hacminin çok düşük olması sebebiyle enerji tasarrufu sağlayamamıştır. Hala en düşük enerji tüketimine sahip devre işhattı yapısıdır. Elde edilen işlem hacmi değeri açısından çizelge 5.1' de verilen literatürdeki DES gerçeklemeleri arasında oldukça gerilerde kalmaktadır. Fakat harcanan güç miktarı ve bir biti şifrelemek için harcanan enerji miktarı açısından oldukça iyi bir yerdedir.

5.2.6 İç yazmaçlı işhattı yapısında gerçekleme

Çalışmada gerçeklenen altıncı ve son yapı iç yazmaçlı işhattı gerçekleme yapısıdır. Şu ana kadar gerçeklenen devreler arasında en düşük enerji tüketimine sahip olan yapı işhattı yapısı olduğundan bu yapı iyileştirilmeye çalışılmıştır. İç yazmaçlı işhattı yapısının seçilmesinde sebep, bölüm 4.1' de bahsedilen işlem hacmini arttırma yöntemlerinden biri olmasıdır. DES algoritmasının tasarımında tur içindeki kritik yolları azaltmak amacıyla, tur içine dört adet yazmaç eklenmiştir. Dört adet iç yazmacı olan bu devre 64 seviyeli işhattı yapısına sahiptir. Şekil 3.4' teki DES f fonksiyonunda E fonksiyonu çıkışına, S – kutularının girişlerine ve çıkışlarına, permütasyon fonksiyonunun çıkışına iç yazmaçlar eklenmiştir. Böylelikle devredeki kritik yolların iyice kısalmasıyla devrenin saat frekansı ve dolayısıyla işlem hacminin artması beklenmektedir. İç yazmaçlı işhattı yapısında gerçekleme şekil 5.5' te verilmiştir. Şekilden de görüldüğü gibi işhattı yapısında 16 adet DES tur fonksiyonunun arasında 80 adet yazmaç vardır. Yani yapıda 16 adet tur art arda dizilmiştir ve her turda 5 adet yazmaç bloğu vardır. Şekilde D girişi düz metni, A girişi anahtar girişini ve Ş çıkışı da şifreli metin çıkışını temsil etmektedir.

İç yazmaçlı işhattı yapısında ilk şifreli metin 80 saat periyodu sonra oluşur ve yapısı gereği ilk şifreli metinden sonra her bir saat periyodunda bir şifreli metin oluşur. Yani denklem 5.1' deki adım sayısı 1' dir. Böylelikle iç yazmaçlı işhattı yapısından yüksek işlem hacmi ve dolayısıyla yüksek güç harcaması beklenmektedir. Düşük enerji tüketimi için hem yüksek işlem hacmi hem de düşük güç tüketimi gerektiğinden bu yapıdaki enerji tüketimi önceden kestirilememiştir. İşlem hacmindeki artış oranına bağlı olarak enerji tüketiminde artış veya azalma görülecektir.

Bölüm 3.3' teki gerçekleme adımları takip edilerek ilk olarak iç yazmaçlı işhattı yapısının VHDL kodu yazılmıştır. Daha sonra yazılan kodun doğru çalışıp çalışmadığını görmek amacıyla fonksiyonel benzetim (simülasyon) yapılmıştır. ModelSim programı yardımıyla yapılan simülasyon sonucu kodun önceden bilinen üç adet giriş çıkış değeri için doğru çalıştığı gözlenmiştir. Yazılan kodun doğru çalıştığına emin olunduktan sonra, tasarım sürecinde bir sonraki adım olan FPGA üzerinde gerçeklemeye geçilmiştir.



Şekil 5.5 : İç yazmaçlı işhattı yapısı.

Bölüm 3.4' teki FPGA üzerinde gerçekleme adımları takip edilerek ilk olarak Xilinx ISE 9.2i programında iç yazmaçlı işhattı yapısı için yazılan VHDL kodu sentezlenmiştir. Yazılan kodun çalışma amacı yüksek işlem hacmi olduğundan, sentezleme seçeneklerinden optimizasyon hedefi olarak hız seçilmiştir. Ayrıca optimizasyon girişimi normalden yükseğe çekilerek FPGA üzerinde en iyi gerçekleme sağlanmıştır.

Sentezleme işleminden sonra iç yazmaçlı işhattı yapısı için FPGA üzerinde dönüştürme, eşleştirme, yerleştirme ve yollandırma işlemleri gerçekleştirilmiştir. Eşleştirme sırasında blok RAM kullanımı da bir seçenek olarak denenmiş fakat kullanılan FPGA' nın halihazırda blok RAM' ları kullanması özelliğinden dolayı işlem hacmini arttırmada etkili olmamıştır. Ayrıca kullanıcı kısıtlamaları olarak saat frekansı verilmiş ve devrenin çalışabileceği en yüksek saat frekansı iterasyonla bulunmuştur. Bu işlemlerin sonucunda oluşan zamanlama ve eşleştirme raporlarından devrenin gerçek alan ve zaman bilgileri elde edilmiştir. İç yazmaçlı işhattı yapısı için elde edilen alan ve zaman bilgileri şöyledir:

```
Design Summary
--------------
Logic Utilization:
  Number of Slice Flip Flops:       6,784 out of  66,560   10%
  Number of 4 input LUTs:           3,341 out of  66,560    5%
Logic Distribution:
  Number of occupied Slices:        4,226 out of  33,280   12%

Timing summary:
---------------

Design statistics:
   Minimum period:   5.754ns   (Maximum frequency: 173.792MHz)
```

İç yazmaçlı işhattı yapısındaki devrenin harcadığı güç ve enerji bilgilerini elde edebilmek için yerleştirme ve yollandırma sonrası simülasyon yapılmıştır. Bu simülasyon sırasında yine devreye MATLAB programı tarafından oluşturulmuş rastgele girişler uygulanmış ve devre mümkün olan en yüksek saat frekansında çalıştırılarak simülasyon sonuçları elde edilmiştir. İç yazmaçlı işhattı yapısındaki devre için saat frekansı 125 MHz' de yapılan simülasyon doğru sonuçları vermiştir. Simülasyon sırasında güç analizi işleminde kullanılmak üzere .vcd uzantılı simülasyon dosyası da oluşturulmuştur. Son olarak elde edilen simülasyon dosyası kullanılarak XPOWER güç ölçüm aracı yardımıyla devrenin harcadığı güç bilgisi

elde edilmiştir. İç yazmaçlı işhattı yapısı için güç değeri 369,1 mW çıkmıştır. Böylelikle yazılan VHDL kodunun FPGA üzerinde gerçeklenmesi aşaması da tamamlanmıştır.

İç yazmaçlı işhattı yapısındaki devrenin en yüksek işlem hacmi değeri denklem 5.1' den yararlanılarak 8 Gb/s olarak bulunmuştur. Ayrıca denklem 5.2' den yararlanılarak bir biti şifrelemek için gerekli enerji miktarı 46,1 pJ/bit olarak bulunmuştur. Bununla birlikte devrenin 64 bitlik bir bloğu şifrelemek için harcadığı enerji miktarı da denklem 5.3' ten yararlanılarak 2,952 nJ olarak bulunmuştur.

Elde edilen sonuçlara göre güç ve enerji tüketiminde işhattı yapısına göre iyileştirme sağlanamamıştır. Bu sonuçta güç tüketimindeki artışı karşılayabilecek kadar işlem hacminin artmaması büyük pay sahibidir. Beklendiği gibi işhattı yapısına kıyasla işlem hacminde az da olsa iyileştirme sağlanmıştır. Hala en düşük enerji tüketimine sahip devre açık farkla işhattı yapısıdır. Elde edilen işlem hacmi değeri açısından çizelge 5.1' de verilen literatürdeki DES gerçeklemeleri arasında oldukça iyi bir yerdedir. Fakat harcanan güç miktarı ve bir biti şifrelemek için harcanan enerji miktarı açısından sıralamada gerilerdedir.

5.3 Farklı DES Yapılarının Karşılaştırılması

Öncelikle bölüm 5.2' de anlatılan ve bu çalışmada gerçeklenmiş olan altı farklı yapıdaki DES devresinin elde edilen alan, zaman, iş hacmi, güç ve enerji bilgileri çizelge 5.2' de birleştirilmiştir.

Çizelge 5.2 : Çalışmada gerçeklenen DES yapılarının karşılaştırılması.

Kaynak	Saat Frekansı (MHz)	İşlem Hacmi (Mb/s)	Güç (mW)	Enerji (nJ)	Enerji/bit (nJ/bit)	LUT sayısı	Flip-Flop sayısı
DES1	83,3	5333,3	48,7	0,584	0,0091	4125	928
DES2	10,6	678,4	597,8	56,372	0,8808	3040	-
DES3	70,4	285,7	17,3	3,875	0,0605	587	69
DES4	38,4	273,5	15,1	3,533	0,0552	717	71
DES5	37	263,3	14	3,402	0,0531	652	71
DES6	125	8000	369,1	2,952	0,0461	3341	6784

Çizelgede ilk sütunda verilen DES yapıları bölüm 5.2' deki anlatım sırasına göre dizilmiştir. DES1 işhattı yapısını, DES2 yazmaçsız yapıyı, DES3 klasik yapıyı, DES4 sekiz turda iki S-kutusu ile gerçeklemeyi, DES5 sekiz turda tek S-kutusu ile gerçeklemeyi ve DES6 da iç yazmaçlı işhattı yapısını temsil etmektedir.

İlk olarak gerçeklemeler kapladıkları alan açısından karşılaştırılırsa, en küçük alana sahip yapı klasik yapıdır. Klasik yapıyı sekiz turda tek S-kutulu gerçekleme ve iki S-kutulu gerçekleme takip etmektedir. Klasik yapının en küçük alanlı çıkmasında şifrelemenin döngü şeklinde ve tek tur ve yazmaç bloğu kullanılarak yapılmasının etkisi büyüktür.

Zaman açısından kıyaslamada ise işhattı yapıları ön plana çıkmaktadır. En yüksek işlem hacmine sahip devre iç yazmaçlı işhattı yapısındaki devredir. Daha sonra işhattı yapısındaki devre gelmektedir. İşhattı yapısının en büyük avantajı olan kısa kritik yollar bu sonucu doğurmuştur.

Devrelerin harcadıkları güç açısından karşılaştırmada ise sekiz turda tek S-kutusu ile gerçekleme en düşük güç tüketimiyle ilk sıradadır. İkinci sırada sekiz turda iki S-kutusu ile gerçekleme, üçüncü sırada ise klasik yapı vardır. Bu sonuçta alandan tasarruf sağlayan devrelerin güç tasarrufu da sağlamasının büyük payı vardır. İşhattı yapıları ise yüksek işlem hacmine bağlı olarak yüksek güç tüketimi yapmaktadır.

Son olarak bu çalışmada en çok üzerinde durulan parametre olan enerji tüketimi açısından kıyaslamaya gelinirse, bir biti şifrelemek için gerekli enerji miktarı en düşük olan yapı işhattı yapısıdır. Enerji tüketimi açısından diğer yapılara göre çok daha iyi bir sonuç vermiştir. Bu sonuçta işhattı yapısında hem yüksek işlem hacmi hem de düşük güç tüketiminin sağlanması etkili olmuştur. Enerji tüketimini azaltmak için diğer yapılar ya güç tüketimini azaltırken işlem hacmini de çok düşürmüşler ya da işlem hacmini arttırırken güç tüketimini de çok arttırmışlardır. Güç tüketimi ve işlem hacmi arasındaki dengeyi en iyi sağlayan devre olan işhattı yapısı, yapısı gereği her saat periyodunda bir şifreli metin üretebilmesinin de avantajıyla bu çalışmada gerçeklenen yapılar arasında enerji tüketimi açısından en iyi sonucu vermiştir.

Bu çalışmada gerçeklenen yapıların karşılaştırılmasından sonra bütün DES gerçeklemelerinin karşılaştırılmasına geçilmiştir. Çizelge 5.1' de verilen literatürdeki

DES gerçeklemeleri ile çizelge 5.2' de verilen bu çalışmadaki DES yapıları birleştirilerek çizelge 5.3 oluşturulmuştur.

Çizelge 5.3 : Bütün DES yapılarının karşılaştırılması.

Kaynak	Teknoloji	Saat Frekansı (MHz)	İşlem Hacmi (Mb/s)	Güç (mW)	Enerji/bit (nJ/bit)
DES1	FPGA	83,3	5333,3	48,7	0,0091
[18]	ASIC	-	38780	420,8	0,011
[39]	FPGA	125	8000	148,6	0,018
DES6	FPGA	125	8000	369,1	0,0461
DES5	FPGA	37	263,3	14	0,0531
DES4	FPGA	38,4	273,5	15,1	0,0552
DES3	FPGA	70,4	285,7	17,3	0,0605
[26]	ASIC	52,19	417,5	25,88	0,061
[19]	ASIC	13,56	102,4	15,97	0,156
[35]	ASIC	-	51,2	9,62	0,188
[31]	FPGA	168	10752	3200	0,297
[17]	ASIC	105	9280	6500	0,700
DES2	FPGA	10,6	678,4	597,8	0,8808
[25]	Mikroişlemci	8	0,079	55,53	703
[40]	Mikroişlemci	-	-	-	725
[34]	Mikroişlemci	-	-	-	2080
[16]	FPGA	333	21300	-	-
[15]	FPGA	237	15100	-	-
[37]	FPGA	188,6	12000	-	-
[36]	FPGA	111,8	7160	-	-
[38]	FPGA	60	3870	-	-
[32]	FPGA	59,5	3808	-	-
[30]	FPGA	47,7	3052	-	-
[29]	FPGA	25,1	402,7	-	-
[33]	Mikroişlemci	300	137	-	-
[28]	FPGA	10	26,7	-	-

Çizelgede ilk sütunda kaynak bilgisi, ikinci sütunda kullanılan teknoloji, üçüncü sütunda devrenin çalışma frekansı, dördüncü sütunda işlem hacmi, beşinci sütunda güç bilgisi ve son sütunda bir bit şifreleme için gerekli enerji miktarları verilmiştir. Ayrıca çizelgedeki sıralama enerji açısından en iyiden en kötüye doğru yukarıdan aşağıya yapılmıştır. Enerji bilgisi olmayan gerçeklemeler ise işlem hacmi değerlerine göre sıralanmıştır.

Çizelge 5.3' ten de görüldüğü gibi en düşük enerji tüketimine sahip devre, bu çalışmada gerçeklenen ve bölüm 5.2.1' de anlatılan FPGA teknolojisi kullanılarak gerçeklenen işhattı yapısındaki devre olmuştur. İkinci sırada [18]' de verilen asenkron yapıdaki ASIC tasarım vardır. Üçüncü sırada ise [39]' da verilen 64 seviyeli iç yazmaçlı işhattı yapısına sahip FPGA üzerinde tasarım vardır. Daha sonra ise bu çalışmada gerçeklenen iç yazmaçlı işhattı yapısı, sekiz turda tek S-kutulu yapı, sekiz turda iki S-kutulu yapı ve klasik yapı sıralanmıştır. Bu çalışmada enerji tüketimi açısından en kötü gerçekleme olan yazmaçsız gerçekleme, 16 devre arasında 13. olabilmiştir. Bölüm 5.1' de de söylendiği gibi mikroişlemci teknolojisi kullanılarak gerçeklenen devreler FPGA ve ASIC tasarımlara göre enerji tüketimi açısından oldukça kötü sonuçlar vermiştir.

Bir biti şifrelemek için gerekli enerji miktarı en düşük çıkan işhattı yapısının enerji tüketimini etkileyen iki faktör yani güç tüketimi ve işlem hacmi arasında çok iyi bir denge sağlamasının bu sonuçta büyük etkisi vardır. Düşük enerji tüketimi için hem düşük güç tüketimi hem de yüksek işlem hacmi gerekmektedir. Diğer yapılar ya işlem hacmini arttırırken güç tüketimini çok arttırmışlar ya da güç tüketimini düşürürken işlem hacmini de çok düşürmüşlerdir. Bu iki faktör arasında en iyi dengeye sahip olan işhattı yapısındaki gerçekleme, yapısı gereği her saat periyodunda bir şifreli metin üreterek en düşük enerji tüketen devre olmuştur.

6. SONUÇLAR VE TARTIŞMA

DES blok şifreleme algoritmasının FPGA üzerinde düşük enerjili tasarımı yüksek lisans tezinde öncelikle çalışmanın amacından ve FPGA' lardan bahsedilmiştir. Daha sonra DES blok şifreleme algoritması ve gerçeklenmesi ile enerji tasarrufu yöntemleri anlatılmıştır. Son olarak, DES blok şifreleme algoritmasının düşük enerjili tasarımı geniş bir biçimde anlatılmıştır. Bu çalışmada güç tasarrufu yöntemlerinden ve işlem hacmini arttırma yöntemlerinden yararlanılarak, altı farklı DES yapısı gerçeklenmiştir. Literatürdeki diğer DES gerçeklemeleri de incelenerek, en düşük enerji tüketen DES devresi elde edilmiştir. Tüm bu tasarımlar sırasında FPGA teknolojisi kullanılmıştır.

Literatürde, DES gerçeklemelerinde genellikle yüksek hız hedef olarak alınmıştır. Düşük güç veya düşük enerji hedefiyle tasarım yok denecek kadar azdır. Bu çalışma ile hem literatürdeki DES gerçeklemeleri işlem hacmi, güç ve enerji açılarından karşılaştırılmış hem de enerji tüketimi en düşük devre gerçeklemesi sunulmuştur. Ayrıca düşük güç tüketen devre tasarımı için gerekli yapısal değişiklikler ve yüksek hızlı devre için gerekli yapısal değişiklikler üzerinde durulmuştur. Bununla birlikte, DES algoritmasını gerçeklemek isteyen tasarımcılar için kapsamlı bir kaynak hazırlanmaya çalışılmıştır.

Bu çalışmada gerçeklenen altı farklı yapı arasında en düşük enerji tüketimine işhattı yapısındaki gerçekleme sahiptir. Ayrıca literatürdeki önceki çalışmalar ile karşılaştırılınca da bir bitin şifrelenmesi için gerekli enerji bakımından en düşük enerji tüketimine sahip devre yine işhattı yapısındaki devre olmuştur. İşhattı yapısında gerçeklemede bir bit şifreleme için 9,1 pJ harcanmaktadır. Literatürdeki çalışmalarda bulunan en düşük değer 11 pJ/bit ile [18]' deki asenkron yapıdaki ASIC tasarımdır. Bu çalışmada gerçeklenen işhattı yapısındaki devre literatürdeki en düşük enerji tüketen DES gerçeklemesi haline gelmiştir.

Düşük enerji tüketimine sahip devre tasarımı için hem yüksek işlem hacmi hem de düşük güç tüketimi gerekmektedir. Bu iki parametreyi birden en dengeli şekilde sağlayan devre yapısı işhattı yapısı olduğundan en düşük enerji tüketen devre olmuştur. Ayrıca işhattı yapısındaki devrede, devrenin yapısı gereği her saat periyodunda bir şifrelenmiş metin oluşmaktadır. Diğer yapılarda ise 8 veya 16 saat darbesinde bir şifreli metin oluşmaktadır. Bu yapısal avantaj da işhattı yapısının en düşük enerji tüketen devre olmasında rol oynamıştır. İç yazmaçlı işhattı yapısı da her saat periyodunda bir şifreli metin oluşturmaktadır fakat fazla alan kaplamasına bağlı olarak harcadığı güç miktarı işhattı yapısına göre çok fazladır. İşlem hacmi açısından daha iyi olmasına rağmen enerji tüketimi açısından işhattı yapısının gerisinde kalmıştır.

Güç tasarrufu yöntemlerinden faydalanılarak tasarlanan klasik yapı, sekiz turda iki S-kutulu yapı ve sekiz turda tek S-kutulu yapı ise beklendiği gibi işhattı yapısından daha düşük güç tüketimine sahiptir. Ancak işlem hacmi açısından işhattı yapısının çok gerisinde kalmalarına bağlı olarak enerji tüketiminde işhattı yapısına göre daha kötü durumdadırlar. Daha önce de söylendiği gibi düşük enerji tüketimi için işlem hacmi ve güç tüketimi arasında çok iyi bir denge kurulmalıdır.

Bu çalışmada FPGA teknolojisi kullanılarak en düşük enerji tüketimine sahip devre olan işhattı yapısındaki devrenin gerçeklenmesi anlatılmıştır. Gelecek çalışma olarak bu devrenin farklı teknolojiler kullanılarak tasarımı ve enerji tüketimi açısından bu farklı yapıların karşılaştırılması düşünülebilir. Ayrıca gerçeklenen devrelerin güvenliği ve güvenlik yöntemleri yardımıyla iyileştirilmesi de çalışılabilecek konular arasındadır.

KAYNAKLAR

[1] **Savaş, E.,** 1994. Dizi Şifreleme Sistemleri ve Doğrusal Karmaşıklık, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[2] **Acar, S.,** 2005. Eliptik Eğri Kriptografisinde Skaler Çarpma Bloğunun VHDL ile Tasarımı, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[3] **Doğan, A.Y.,** 2008. AES Algoritmasının FPGA Üzerinde Düşük Güçlü Tasarımı, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[4] **FIPS 46-3,** 1999. Data Encryption Standard. National Institute of Standarts and Technology (NIST).

[5] **Lagger, A.,** 2002. Implementation of DES Algorithm Using FPGA Technology, *Semester Project,* EPFL, Lozan.

[6] **Berna, A.,** 1998. Sahada Programlanabilir Kapı Dizileri ile Lojik Devre Tasarımı ve VHDL Kullanılarak Bazı Devrelerin Gerçekleştirilmesi, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[7] **Topçu, İ.H.,** 2002. Sahada Programlanabilir Kapı Dizileri Kullanılarak Sayısal Tasarım Kartı Gerçeklenmesi, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[8] **Ordu, L,** 2006. AES Algoritmasının FPGA Uzerinde Gerceklenmesi ve Yan Kanal Analizi Saldırılarına Karsı Guclendirilmesi, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[9] **Xilinx**. Spartan – 3 Field Programmable Gate Arrey Family Data Sheet. http://www.xilinx.com/support/documentation/data_sheets/spartan3

[10] **Kaplan, T.,** 2006. Trivium Dizi Şifreleme Algoritmasının FPGA Üzerinde Gerçeklenmesi, *Lisans Bitirme Ödevi,* İ.T.Ü. Elektrik-Elektronik Fakültesi, İstanbul.

[11] **Şahinoğlu, M.,** 2008. AES Algoritmasının FPGA Üzerinde Gerçeklemesine Elektromanyetik Alan Saldırısı, *Yüksek Lisans Tezi,* İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[12] **Stinson, D.R.,** 2002. Cryptography, Chapman & Hall/CRC Press Company, sf. 95-102, United States of America.

[13] **Hsu, Y.C., Tsai, K.F., Liu J.T. and Lin, E.S.,** 1995. *VHDL Modeling For Digital Synthesis*, Kluwer Academic Publishers, Riverside.

[14] **Koç, Ç.K. and Paar, C.,** 2000. A 12 Gbps DES Encryptor/Decryptor Core in an FPGA, *CHES 2000, LNCS Vol. **1965***, pp. 156 - 163.

[15] **Pasham, V. and Trimberger, S.,** 2001. High-Speed DES and Triple DES Encryptor/Decryptor,*Xilinx*, www.xilinx.com/support/documentation/application_notes/xapp270.pdf

[16] **Rouvroy, G., Standaert, F.X., Quisquater, J.J and Legat, J.D.,** 2003. Efficient Uses of FPGAs for Implementations of DES and Its Experimental Linear Cryptanalysis, *IEEE Transactions On Computers, Vol. **52**, No. 4*, pp. 473 - 482.

[17] **Wilcox, D.C., Pierson, L.G., Robertson, P.J., Witzke, E.L. and Gass, K.,** 1999. A DES ASIC Suitable for Network Encryption at 10 Gbps and Beyond, *CHES'99, LNCS Vol. **1717***, pp. 37 - 48.

[18] **Sotiriou, C.P. and Papaefstathiou, Y.,** 2003. Design-Space Exploration of a Cryptography Algorithm, *ICECS 2003, Vol.**2***, pp. 858 - 861.

[19] **Siu, P.L., Choy, C.S., Butas, J. and Chan, C.F.,** 2001. A low power asynchronous DES, *ISCAS 2001,vol. **4***, pp. 538 - 541.

[20] **Kaps, J.P.,** 2006. Cryptography for Ultra-Low Power Devices, *Ph.D. thesis*, Worcester Polytechnic Institue.

[21] **Rabaey, J.M. and Pedram, M.,** 1996. *Low Power Design Methodologies*, Kluwer Academic Publishers, Norwell, Massachusetts, 1st edition.

[22] **Rabaey, J.M. and Pedram, M.,** 1996. *Low Power Design Methodologies*, Kluwer Academic Publishers, Norwell, Massachusetts, 2nd edition.

[23] **Keating, M., Flynn, D., Aitken, R., Gibbons, A. and Shi, K.,** 2007. *Low Power Design Methodology Manual For System-on-Chip Design*, Springer, 1st edition.

[24] **Kitsos, P., Koufopavlou, O., Selimis., G. and Sklavos, N.,** 2005. Low Power Cryptography, *Journal of Physics: Conference Series **10***, pp. 343 - 347.

[25] **Kolstad, J.,** 2004. A Comparative Analysis of Energy Usage Between Equal-Strength Cryptographic Algorithms, *Survey Report*, http://islab.oregonstate.edu/koc/ece575/04Project1/Kolstad/report.doc.

[26] **Lim, Y.W.,** 2000. Efficient 8-Cycle DES Implementation, *AP-ASIC 2000, Proceedings of the Second IEEE Asia Pacific Conference on*, pp. 175 - 178.

[27] **Ghosh, A., Devadas, S., Keutzer, K. and White, J.,** 1992. Estimation of average switching activity in combinational and sequential circuits, *DAC '92: Proceedings of the 29th ACM/IEEE conference on Design automation*, pp. 253 - 259.

[28] **Wong, K., Wark, M., Dawson, E.,** 1998. A Single-Chip FPGA Implementation of the Data Encryption Standard (DES) Algorithm, *IEEE proc. GLOBECOM 98, vol.* ***2***, pp. 827 - 832.

[29] **Kaps, J.P., Paar, C.** 1998. Fast DES Implementation for FPGAs and its Application to a Universal Key-Search Machine, *Proc. 5th Annual Workshop on Selected Areas in Cryptography, SAC '98, Vol.* ***1556****,* pp. 234 - 247.

[30] **Free-DES Core,** 2000. http://www.free-ip.com/DES/.

[31] **Patterson, C.,** 2000. High performance DES Encryption in Virtex FPGAs using Jbits, *IEEE Symposium on Field-Programmable Custom Computing Machines, FCCM '00*, pp. 113 - 121.

[32] **McLoone, M., McCanny, J.V.,** 2000. High-performance FPGA Implementation of DES Using a Novel Method for Implementing the Key Schedule, *IEEE Workshop on Signal Processing Systems (SiPS 2000)*, Vol. **150**, pp. 374 - 383.

[33] **Biham, E.,** 1997. A Fast New DES Implementation in Software, *Prov. 4th Int. Workshop on Fast Software Encryption*, Vol. **1267**, pp. 260 - 271.

[34] **Potlapally, N.R., Ravi, S., Raghunathin, A., Jha, N.K.,** 2003. Analyzing the Energy Consumption of Security Protocols, *International Symposium on Low Power Electronics and Design*, pp. 25 - 27.

[35] **Kessels, J., Kramer, T., Besten, G., Peeters, A., Timm, V.,** 1999. Applying Asynchronous Circuits in Contactless Smart Cards, *Intl. Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 36 - 44.

[36] **Patel, V., Joshi, R.C., Saxena, A.K.,** 2005. FPGA Implementation of DES Using Pipelining Concept with Skew Core Key-Scheduling, *Journal of Theoretical and Applied Information Technology*, Vol. **150**, pp. 295 - 300.

[37] **Trimberger, S., Pang, R., Singh, A.,** 2000. A 12 Gbps DES Encryptor/Decryptor Core in an FPGA, *CHES 2000, Springer-Verlag*, Vol. **1965**, pp. 155 - 163.

[38] **McLoone, M., McCanny, J.V.,** 2000. A High Performance FPGA Implementation of DES, *IEEE 2000,* pp. 374 – 383.

[39] **Katashita, T., Maeda, A. and Yamaguchi, Y.,** 2007. A Low-Power Design Method for FPGA Using Extra Flip-Flops Driven by Phase-Shifted Clock, *Electronics and Communications in Japan, Vol. **90***, pp. 1132 - 1142.

[40] **Saputra, H., Vijaykrishnan, N., Kandemir, M., Irwin, M.J., Brooks, R., Kim, S. and Zhang, W.,** 2003. Masking the Energy Behavior of DES Encryption, *Proceedings of the Design,Automation and Test in Europe Conference and Exhibition*, pp. 1530 - 1536.

ÖZGEÇMİŞ

Ad Soyad: Tarık KAPLAN

Doğum Yeri ve Tarihi: Antalya - 1983

Lisans Üniversite: İstanbul Teknik Üniversitesi, Elektronik Mühendisliği, 2006