İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK – ELEKTRONİK FAKÜLTESİ

CEP TELEFONLARINDAKİ DİZİ ŞİFRELEMENİN SAHADA PROGRAMLANABİLİR KAPI DİZİLERİ İLE GERÇEKLENMESİ

BİTİRME ÖDEVİ

Furkan DAYI

040010136

Bölümü: Elektronik ve Haberleşme Mühendisliği

Programı: Elektronik Mühendisliği

Danışmanı: Yrd. Doç. Dr. Sıddıka Berna Örs YALÇIN

MAYIS 2006

ÖNSÖZ

Bitirme ödevi boyunca; bilgilerinden faydalandığım, değerli zamanını aldığım, bana fikirleri ve yönlendirmeleri ile büyük destek veren Sayın Yrd. Doç. Dr. Sıddıka Berna Örs Yalçın'a sonsuz saygı ve teşekkürlerimi sunarım.

Mayıs, 2006 Furkan DAYI

İÇİNDEKİLER

ÖZET

İçinde bulunduğumuz bilgi çağında, bilginin taşınması sırasında güvenliğinin de sağlanması önem arz etmektedir. Kullanılan bilgi iletim kanallarının herkesin erişimine açık olması bu önemin sebebidir. İnternet üzerinden banka işlemlerini üçüncü şahısların saldırısından en az etkilenecek şekilde yapmak, cep telefonlarından başkalarının anlayamayacağı konuşmalar yapmak ancak kriptografi'nin gelişmesiyle mümkün olmuştur. Kriptografi; bir bilginin, herkesin erişimine açık elektronik haberleşme kanallarından iletilebilmesi için bir takım dönüşümler sonucunda değişikliğe uğratılarak üçüncü şahıslar için anlaşılmaz hale getirilmesi amacıyla yapılan işlemlerdir.

Kullanımı günümüzde oldukça artan cep telefonları sayesinde, kapsama alanı dâhilinde istenilen yerden kablosuz olarak görüşme yapma imkânına kavuşuldu. Fakat cep telefonunun iletim ortamının (hava) herkese açık olması, telefondan çıkan bilgi işaretlerinin başkaları tarafından erişilebilme sorununu doğurmaktadır. Bu yüzden havaya gönderilecek işaretlerin şifrelenmiş olarak gönderilmesi gerekmektedir. Cep telefonlarında, bu şifrelemeyi sağlamak için A5 algoritması kullanılmaktadır.

Bu çalışmada, örnek olarak A5'in de bulunduğu şifreleme çeşitlerinden dizi şifreleme (stream cipher) incelenmiştir. Dizi şifreleme açıklanmış, genel çalışma prensipleri üzerinde durulmuştur. Daha sonra özel olarak A5 algoritmasından bahsedilmiştir. Programlanabilir lojik özelliği olan sahada programlanabilir kapı dizileri (FPGA – Field Programmable Gate Arrays) hakkında bilgi verilmiştir. Çalışmanın pratik kısmında ise cep telefonlarındaki güvenlik algoritması A5, sayısal donanım tasarlama dili olan VHDL (Very high speed integrated circuit Hardware Description Language) ile tasarlanmış ve FPGA donanımı üzerinde gerçeklenmiştir. Bölüm 4'de bu gerçekleme adımları anlatılmıştır.

SUMMARY

In today's information age, providing information security is a very important thing during information transportation between two locations. The reason of this importance is that everyone can reach to this information transportation channels. Doing banking operations over internet with minimum effects of attacks, doing conversations over mobile phones that nobody can understand it, is provided thanks to the developments in cryptography. Cryptography is the processes that are made for the secure information transportation so that nobody can understand the contents of the transported signals.

With today's most widely used mobile phones, wireless communication is provided within the boundaries of the coverage area. However, because the transmission field of mobile phone is open to everybody (air), everybody can reach the signals which come from mobile phones. Hence, the signals should be encrypted. In mobile phones, this encryption is provided with A5 algorithm.

In this work, stream ciphers which is a kind of encryption is examined. Stream ciphers and their working principles are explained. Then, a specific stream cipher, A5 algorithm is mentioned. Information about the field programmable gate arrays (FPGA) is given. On the practical side of this work, mobile phones' security algorithm A5 is designed with VHDL (Very high speed integrated circuit Hardware Description Language) and implemented on the field programmable gate arrays.

1. GİRİŞ

Kriptografi, Yunancada “gizli” ve “yazmak” anlamına gelen köklerin birleşmesi ile oluşmuş, günümüz gizli haberleşmesini tanımlayan temel kelime olmuştur. Eski zamanlardan beri bilgi iletimi amacı ile farklı şekillerde kullanılmıştır. Örneğin Sezar’ın kendi komutanlarına emirler göndermek amacıyla çapı değişen bir silindirin üzerine sarılmış ince kâğıda silindir ekseni doğrultusunda yazı yazarak kullandığı ve bu mesajın ancak aynı çapta bir silindir kullanılarak okunabildiği bilinmektedir [1]. Bu tip basit sistemler 1. ve 2. dünya savaşlarında hızla gelişmiş ve günümüz karmaşık sistemleri ortaya çıkmıştır.

1970’lere kadar daha çok askeri amaçlı olarak kullanılan kripto sistemleri bu tarihten sonra sivil amaçlı olarak da kullanılmaya başlanmıştır. Bunun en önemli sebeplerinden biri ise artan haberleşme trafiğinin güvenliği ve özellikle yüksek ekonomik önem taşıyan bazı verilerin sivil haberleşme hatları kullanılarak iletilmesidir.

Kriptolojik terminolojide şifrelenmemiş mesaja “açık metin” (plain text), şifrelenmiş mesaja ise “şifre metin” (cipher text) adı verilmektedir. Tüm kriptolojik sistemler açık metinden şifreli metni elde edebilmek için özgün dönüşümler kullanırlar. Bu dönüşümler bir “anahtar” yardımıyla yapılır. Bu anahtar gizli ya da açık (public key) olabilir. Gizli tutulan anahtarda, önceden taraflar anlaşmaya varmış olup anahtar üçüncü şahıslara bildirilmez. Üçüncü bir kişinin tüm anahtarları deneyerek şifreli mesajdan açık mesajı elde etmesini (şifreyi kırmak) önlemek için anahtarın seçildiği küme olası derecede geniş tutulmalıdır. Alıcı tarafa ulaşan şifreli mesaj şifre çözme anahtarı kullanılarak çözülür ve içeriğine ulaşılır. Anahtar olmadan şifreli mesajı çözmek ya da kullanılan anahtarı bulmak için yapılan tüm çalışmalara birden “kripto-analiz” denir. Günümüzde kripto-analiz çalışmaları sadece bir mesaja saldırmak için değil aynı zamanda tasarlanan kripto sistemlerinin güvenilirliğini test etmek için de kullanılan çok önemli bir disiplin haline gelmiştir.

Gizli anahtarlı şifreleme sistemleri blok şifreleme (block ciphers) ve dizi şifreleme (stream ciphers) olmak üzere ikiye ayrılırlar. Blok şifrelemede, genel

olarak açık metin sabit boyda bloklara bölünür ve her bir blok üzerinde diğer bloklardan bağımsız dönüşümler uygulanır. Blok şifreleme yöntemini kullanan sistemler basit yerine koyma dönüşümlerini kullanırlar. Bu nedenle bu sistemlerde kaba kuvvet (brute force = kümedeki tüm anahtarları denemek) yaklaşımını zorlaştırmak amacıyla geniş bir anahtar kümesi kullanılır.

Dizi şifreleme yönteminde ise açık metin daha önceden belirlenmiş bir boyutta birimlere (ya da daha çok uygulandığı üzere tek bitlik sayılara) bölünür ve her bir birim zamanla değişen bir fonksiyonla şifrelenir. Dizi şifrelemede her birim için kullanılan anahtar rastgele olarak belirlendiğinden aynı açık mesaj birimleri, zamana bağlı olarak farklı şifreli mesaj birimlerinin oluşmasına neden olurlar. Dizi şifrelemede birim olarak ya Latin alfabesinden bir harf ya da bir basamaklık ikili bir sayı seçilir.

Dizi şifreleme sistemlerinde genelde açık mesajdaki bir bit, şifreli mesajda yalnızca bir biti etkiler. Bu kriptografik olarak olası en kötü yayılımdır. Ancak buna karşılık kullanılan gizli anahtardaki her bir bit şifreli mesajdaki birçok biti etkileyebilir. Böylece anahtar yayılımı oldukça iyi olabilir. Cep telefonlarında kullanılan A5 algoritmasında da durum böyledir. Konuşma sonucu oluşan ses işaretlerinden bir bit yalnızca algoritma sonucu üretilen bir bit ile exorlanır. Fakat algoritma sonucu oluşan bit, anahtarın son derece efektif kullanılmasıyla oldukça rastlantısal hale getirilmiştir. Bu sayede A5 algoritmasının anahtar yayılımının iyi olduğu söylenebilir.

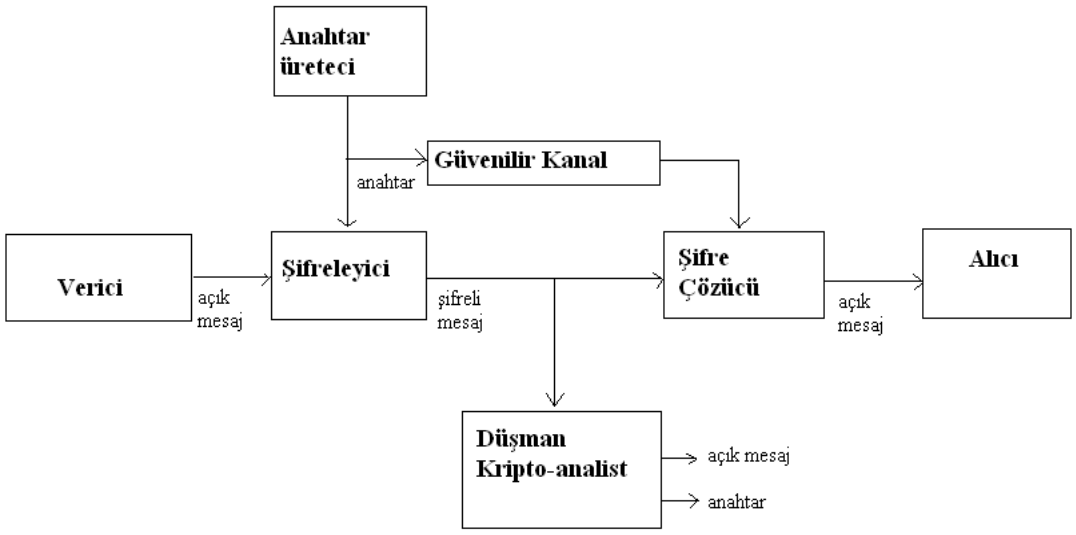
Çalışmanın ileriki bölümlerinde dizi şifreleme mercek altına alınacak, dizi şifrelemede kullanılan elemanlar incelenecek ve bu elemanların hangi özelliklere sahip olması gerektiği belirtilecektir. Daha sonra özellikle A5 algoritması üzerinde durulacak, çalışma prensipleri ve güvenilirliği incelenecektir. Çalışmanın pratik kısmında yapılan A5 algoritmasının gerçeklenmesi aşama aşama anlatılacak ve elde edilen sonuçlar yorumlanacaktır. Pratik çalışma, sahada programlanabilir kapı dizileri üzerinde gerçeklendiği için bu tümdevreyle ilgili bilgiler de pratik çalışmanın anlaşılabilmesi için gerçeklemenin anlatımından önce sunulacaktır.

2. DİZİ ŞİFRELEME SİSTEMLERİ

2.1. Temel Yapısı

Kripto 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" kripto sistemleri olarak ta bilinirler. Açık anahtarlı ya da anti-simetrik kripto sistemlerinde şifreleme ve şifre çözme işlemleri, gizli-anahtarlı sistemlerin aksine, farklı anahtarlar kullanılarak gerçekleştirilir. Bu nedenle açık anahtarlı sistemler simetrik olmayan bir yapıya sahiptir.

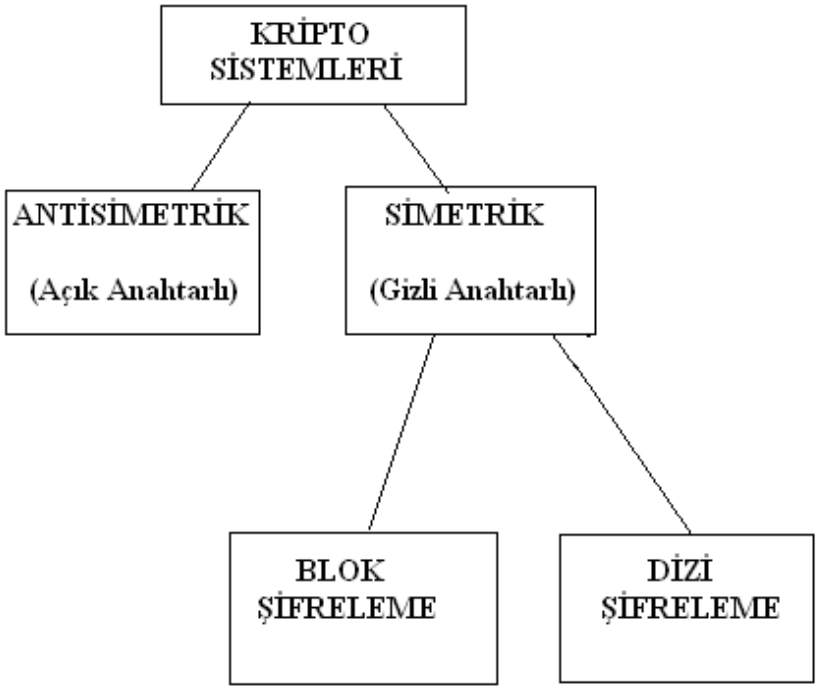
Simetrik yapılı gizli anahtarlı kripto sistemlerinde şifreleme ve şifre çözme anahtarları aynıdır. Şekil 2.1'de gizli anahtarlı sistemin genel yapısı görülmektedir.



Şekil 2. 1 Gizli anahtarlı sistemin genel yapısı

Simetrik sistemler Şekil 2.2'de görülebileceği üzere blok şifreleme ve dizi şifreleme olmak üzere ikiye ayrılır. Blok şifreleme sistemlerinde mesaj sabit uzunluklu bloklara bölünür ve her bir blok diğerlerinden bağımsız bir şekilde şifrelenir. Bu şekilde her bir bloğa karşılık aynı alfabeden aynı boyutta başka bir blok karşılık düşürülür. Bu nedenle blok şifreleme sistemleri basit yerine koyma işlemidir [2]. Kripto analist eğer anahtarı bilmiyorsa eline geçirdiği şifreli mesajı

çözmek için şifrelemede kullanılmış olması olası tüm anahtarları denemek zorundadır. Buna, kaba kuvvet yaklaşımı (brute force) denir.



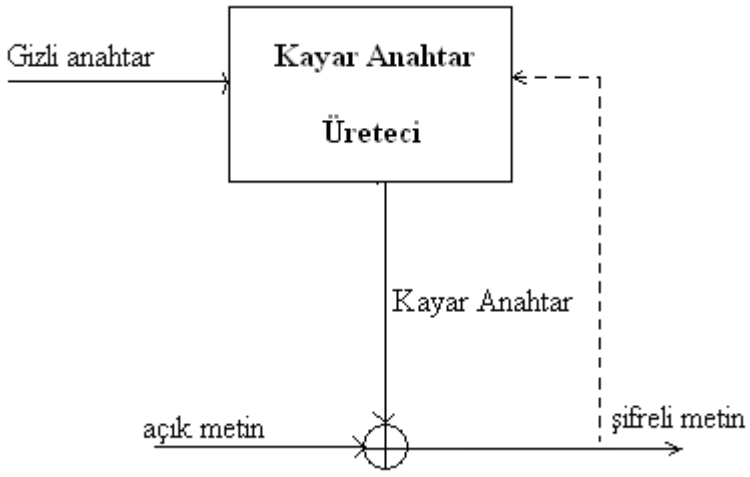
Şekil 2. 2 Kripto Sistemleri

Dizi şifreleme işlemleri ise, her bir mesaj birimini, zamanla değişen bir fonksiyon kullanarak şifreler. Dizi şifreleme sisteminin içyapısı zamana bağlı olarak durum değiştiren bir makine olarak düşünülebilir. Dolayısıyla şifrelemede kullanılan fonksiyonun zamana bağımlılığı makinenin durum değiştirme bağıntısıyla tanımlanır.

Bir dizi şifreleme sisteminde mesaj basamakları tek tek şifrelendiğinden mesajlar genelde bir dizi şeklinde düşünülür. Mesaj dizisinin her bir basamağı 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.

Dizi şifreleme sistemi "kayar anahtar üreteci" (KAÜ) olarak adlandırılan çıkışında sözde-rastgele (pseudorandom) diziler üreten bir sonlu durumlu makineden oluşur. Mesajın ikili bir sayı dizisi olduğu düşünülürse sonlu durumlu makinenin çıkışındaki basamaklar da ikili sayı dizisi olmalıdır. Bu dizinin bitleri

Şekil 2.3'de gösterildiği gibi ardışıl olarak açık mesaj dizisindeki bitlerle exorlanır. Sonuçta çıkan dizi şifreli metin dizisi olarak adlandırılır. Bazı durumlarda şifreli metin de KAÜ'de kullanılır, fakat eğer şifreli metinin bir biti yanlış üretildiyse sonraki bitleri de hep hatalı olacaktır.



Şekil 2. 3 İkili Toplamsal Dizi Şifreleme

Anahtar dizisi de denilen KAÜ'nin çıkışındaki dizinin rastgele görünüşlü olmasındaki amaç, açık mesaj dizisindeki bitlerle şifreli mesaj bitleri arasındaki karşılıklı bilginin (iki büyüklüğün birbiri hakkındaki taşıdığı bilgi) sıfıra inmesini sağlamaktır [2]. Böylece bir dizi, tamamen rastgele bir başka diziyle exorlanırsa sonuçta çıkacak dizi kendisinden istatiksel olarak bağımsız olacaktır. Bu ise tüm kripto sistemlerinin ana amacıdır.

Bilindiği gibi tamamen deterministik yollarla gerçek rastgele dizilerin üretilmesi olanaklı değildir. Bu yüzden amaç KAÜ'nin olabildiğince rastgele görünüme sahip bit dizileri üretmesini sağlamak olmalıdır. Genel olarak, eğer bir dizi içerisinde yapısal bir düzenlilik bulunmuyorsa, dizi elemanları hakkında önceki terimlere bakarak hiçbir öngörü ya da dizi üzerinde herhangi bir tanımlama yapılamıyorsa, o dizinin rastgeleliğinden söz edilebilir. Burada tanımlama sözüyle göreceli olarak az sayıda terimden dizinin yeniden üretilebilmesini sağlayacak bir kural ya da bağıntı kastedilmektedir. Gerçek bir rastgele dizinin yeniden üretilebilmesi için bir bağıntı bulmak imkânsızdır. Diğer yandan deterministik yollarla üretilen sözde rastgele dizilerin gerçek rastgele diziler kadar güçlü olacağını düşünmek yanlış olur. Dolayısıyla çok zor ve karmaşık ta olsa bir bağıntı bulunabilir. O halde yapılması gereken sistem tasarlanırken bu bağıntının yeterince

karmaşık olmasını sağlamak, böylece eldeki hesaplama olanaklarıyla bulunmasının imkânsız hale getirmek olacaktır.

Dizi şifreleme sistemlerinde en çok kullanılan ikili toplamsal dizi şifrelemede (Şekil 2.3) gerçek anahtar, kayar anahtardan farklıdır. Gizli anahtar KAÜ'ni kontrol ederek asıl anahtar dizisinin üretilmesini sağlar. Gerçek gizli anahtarsa hiçbir zaman şifreleme işlemlerinde doğrudan kullanılmaz. Gizli anahtarın uzunluğu kayar anahtarın uzunluğu yanında oldukça küçüktür. Şifreli metin bitleri, ikili açık metin bitlerinin kayar anahtar dizisiyle terim terim modülo-2 toplanmasıyla (exor) elde edilirler. Şifre çözmek içinse aynı işlem şifreli metin için yapılır ve açık metin elde edilir.

2.2. Güvenilirlik

2.2.1. Teorik ve Pratik Güvenilirlik

Teorik güvenilirlik düşman kripto-analistin elinde sonsuz bir hesaplama gücü ve sınırsız bir zamanı olduğunda söz konusu sistemin ne kadar güvenilir olabileceği sorusuyla yakından ilgilidir. Teorik olarak güvenilir bir sistemin inşa edilebilmesi için gerekli anahtar miktarları pratikte uygulanabilir sistemlerin kurulabilmesini imkânsız hale getirmiştir [2].

Pratik güvenilirlik ile kastedilen sınırlı miktarda hesaplama gücüne ve zamana sahip bir kripto-analistin saldırısına karşı sistemin ne kadar güvenilir olduğudur. Örnek olarak açık anahtarlı sistemler teorik değil, pratik açıdan güvenilir sistemlerdir. Bu tür sistemler hep karşı tarafın sınırlı imkânlarla donatılmış olduğu göz önüne alınarak tasarlanırlar.

2.2.2. Mükemmel Gizlilik

Şifreli mesaj biliniyor, fakat şifreli mesaj bitlerinden açık mesaj bitini elde etme olasılığı ½'yi geçemiyorsa sistem teorik açıdan mükemmel gizlilik sağlıyor demektir. Bu durumda kripto-analistin sadece şifreli mesajı ele geçirmiş olması anahtarı bulmasını sağlamaz.

Sistemin mükemmel gizliliği sağlaması için açık mesaj dizisindeki basamak sayısı kadar gerçek rastgele anahtar basamağının kullanılması gerekmektedir. Fakat şifreleme işleminin gerçek rastgele bir anahtarla yapılmasından sonra şifre çözme işlemini gerçekleştirmek için bu rastgele dizinin alıcıya nasıl ulaştırılacağı büyük bir

sorundur. Şu andaki teknoloji çok büyük boyutlu rastgele dizilerin saklanmasını, iletilmesini ve üzerinde işlem yapılmasını pratik uygulama açısından olanaklı kılacak durumda değildir. Bu nedenle tek seferlik dizilerin kullanıldığı alanlar ancak gizliliğin maksimum derecede gerektiği yerlerdir.

Tek seferlik dizilerin pratik olarak gerçekleştirilebilmesi ve işletilmesinin oldukça zor olmasına rağmen gündeme getirdikleri teorik güvenilirlik kavramı bazı değişikliklerle de olsa benzer sistemlerin geliştirilmesine olanak sağlamıştır. Dizi şifreleme sistemleri tamamen tek seferlik diziden yola çıkılarak tasarlanmış sistemlerdir. Ancak daha önce de belirtildiği gibi, sözü geçen rastgele dizi KAÜ denilen makineyle deterministik yollarla alıcı ve vericide eş zamanlı olarak üretilirler. Bu dizi kayar anahtar dizisi olarak adlandırılır ve sistemin tüm güvenilirliği bu dizinin ne kadar rastgele olduğuna bağlıdır. Eğer kripto-analist anahtar dizisinin bir kısmına erişir ve sonraki bitler üzerinde ½'den daha iyi bir tahminde bulunabilirse bu sistemin şifreleme açısından sağlam bir sistem olmadığı açıkça görülür. Kripto-analist şifreyi kırma yolunda önemli bir adım atmış sayılır.

Anahtar dizisindeki öngörülebilirliği azaltmak için ilk akla gelen önlem, dizinin periyodunu olabildiğince büyük yapmaya çalışmaktır. Periyot doğrusal yinelemeli (linear recursive) bir bağıntıyı tanımlar; bu nedenle periyodik bir kayar anahtar dizisinin bir periyodunu bilmek diğer kısmını belirlemek için yeterlidir.

Bir anahtar dizisinin öngörülebilirliğini azaltmak için yapılacak en önemli iş doğrusal karmaşıklığının (bir diziyi üretecek en kısa doğrusal geribeslemeli ötelemeli yazıcının (DGÖY) uzunluğu) olabildiğince büyük tutulmasıdır. Bununla beraber DGÖY hücreleri içindeki bitlerin iyi bir şekilde dağılması gerekir.

2.3. Kayar Anahtar Üreteci

Kayar anahtar üreteci dizi şifreleme sistemlerinde bulunan en önemli elemandır. Şifreleme için kullanılacak bit dizisini üretir. Yapısı sürücü ve birleştirici kısım olarak iki bölüme ayrılabilir. Sürücü kısım KAÜ'nin iç durumunu yönetir ve büyük periyotlu, iyi istatiksel dağılıma sahip dizilerin üretilmesinden sorumludur. Buna karşılık bu dizilerin doğrusal karmaşıklığı büyük olmaz. Genelde sürücü kısım sözde-gürültü dizileri üreten uygun bağlanmış DGÖY'lardan oluşur. Bu DGÖY'ların çıkışındaki dizilerin doğrusal karmaşıklığını arttırmak işi de birleştirici

kısma aittir. Birleştirici kısım sürücünün çıkışındaki diziler üzerinde dönüşümler uygulayarak, bu dizilerin uzun periyodunu ve iyi dağılım özelliklerini bozmadan doğrusal karmaşıklığı büyük ve kriptografik açıdan güçlü dizilerin üretilmesini sağlar [2].

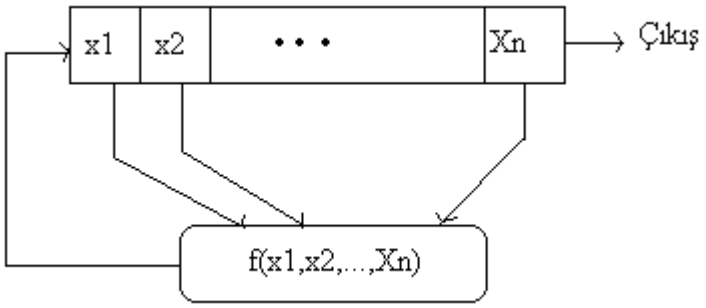
KAÜ için seçilecek birleştirici fonksiyonun sağlaması istenen koşullar şu şekilde sıralanabilir:

- Fonksiyon periyodik sürücü dizisinin tüm istatiksel özelliklerini KAÜ'nin çıkışındaki kayar anahtar dizisine taşımalıdır.
- Fonksiyon, kayar anahtar dizisinin periyodunu girişindeki sürücü dizilerinin periyotlarına göre maksimum yapmalıdır.
- Fonksiyon, kayar anahtar dizisinin doğrusal karmaşıklığını sürücü dizilerin doğrusal karmaşıklığına göre maksimum yapmalıdır.
- Fonksiyon, kayar anahtar dizisi ile sürücü diziler arasında enformasyon sızıntısına neden olmamalıdır.
- Fonksiyonu gerçekleştirmek kolay ve hızlı olmalıdır.
- Eğer olanak varsa gizli anahtarın fonksiyonu da kontrol etmesi sağlanmalıdır [2].

2.3.1. Ötelemeli Yazıcılar

Elektronik ve haberleşme teknolojisinde oldukça yaygın olarak kullanım alanı bulan ötelemeli yazıcılar basit yapıları ve kolay uygulanabilirlikleri ile şifreleme sistemlerinde de sıkça kullanılmaktadırlar. Ötelemeli yazıcı dizilerinin belirli davranış özellikleri göstermesi ötelemeli yazıcıların şifrelemede daha etkin kullanımına sebep olmuştur. Bu davranış özelliklerinden bir tanesi belirli yapılardaki ötelemeli yazıcıların çıkışında rastgele görünüşlü dizilerin elde edilebilmesidir. Daha önce de belirtildiği gibi kripto sistemleri açısından rastgele dizilerin deterministik yollarla elde edilmesi büyük önem taşımaktadır.

Bir dizi şifreleme sisteminin içerisinde rastgele dizileri üreten KAÜ bulunur. Birçok KAÜ'nin temel elemanı olarak ötelemeli yazıcılar kullanılmaktadır. Bu ötelemeli yazıcılar belirli bir fonksiyonu sağlayacak şekilde birbirine bağlanır. Seçilen fonksiyonun ötelemeli yazıcı çıkışlarındaki dizileri nasıl etkileyeceği bilinmektedir. Bu yolla kayar anahtar üretecinin ürettiği dizinin istenen özellikte olması sağlanmış olur. Üstelik kurulan yapı çok basit ve analizi kolaydır.



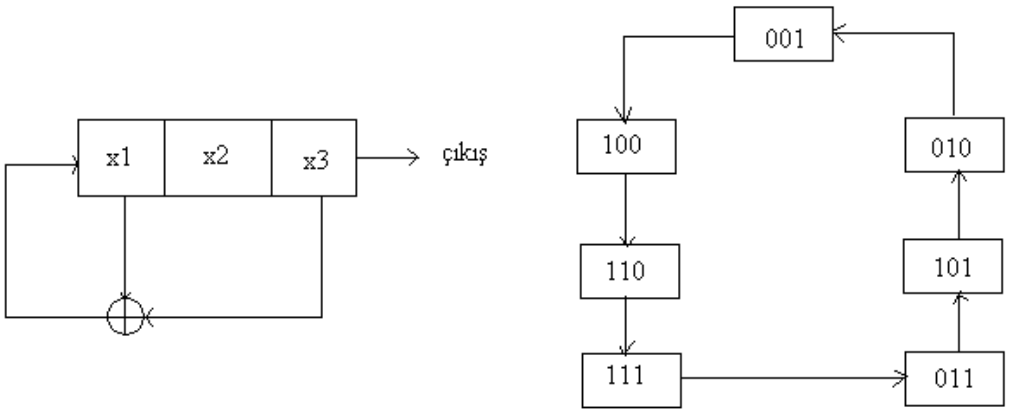
Şekil 2. 4 Geribeslemeli ötelemeli yazıcı

Şekil 2.4'de görüldüğü üzere bir geribeslemeli ötelemeli yazıcı ardarda bağlanmış hücrelerden ve bir geribesleme fonksiyonundan oluşmaktadır. Her hücrenin içerisinde bir sayı bulunur. Her hücrenin de saat girişi vardır. Her saat darbesiyle hücre içerikleri sağa doğru kaymakta, geribesleme fonksiyonun ürettiği değer en soldaki hücreye yerleşmektedir.

Belirli bir andaki hücre içeriklerinin tümü, ötelemeli yazıcının o anda bulunduğu durumu belirler. n uzunluklu bir ötelemeli yazıcının 2^n farklı durumu olacağı açıktır. Ötelemeli yazıcı eğer bu durumlarından bir tanesinden başlarsa, yazıcı belirli durumlardan geçerek bir süre sonra belirli bir durum çevrimine oturur. Ötelemeli yazıcı her zaman başladığı duruma dönmeyebilir.

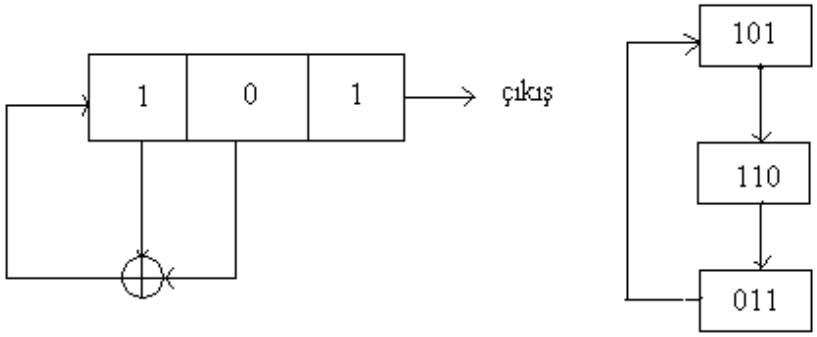
Şekil 2.4'deki fonksiyon, $f(x_1,x_2,\ldots,x_n)=c_1x_1 \oplus c_2x_2 \oplus \ldots \oplus c_nx_n$ şeklinde tanımlanırsa bu fonksiyona doğrusal; bu türden bir geribesleme fonksiyonuna sahip ötelemeli yazıcılara da doğrusal geribeslemeli ötelemeli yazıcılar (DGÖY) denir. Eğer basamaklar üzerinden işlem yapılıyorsa $\oplus$ işlemi modülo-2 toplamayı göstermekte olup x ve c terimleri {0,1} kümesinin elemanlarıdır.

Uzunluğu n olan bir ötelemeli yazıcı 2^n mümkün duruma sahip sonlu durumlu bir makinedir. Herhangi bir ötelemeli yazıcının davranışını belirlemek için durum diyagramı çizilmesi gerekir. Şekil 2.5'de uzunluğu 3, geribesleme fonksiyonu $f(x_1,x_2,x_3)=x_1 \oplus x_3$ olan ötelemeli yazıcının "001" başlangıç durumuna sahip durum diyagramı çizilmiştir [2].



Şekil 2. 5 Bir çeşit ötelemeli yazıcı ve durum diyagramı

Şekil 2.5'den görüldüğü üzere DGÖY bir süre sonra başladığı duruma geri dönmüştür. Bu da DGÖY'nın çıkışındaki dizinin periyodik olduğunu gösterir. Bu dizinin ilk periyodu "1001110" a eşittir. Bu şekilde kendini periyodik olarak tekrar eden dizilere yarı-sonsuz dizi denir.



Şekil 2. 6 Bir çeşit ötelemeli yazıcı ve durum diyagramı

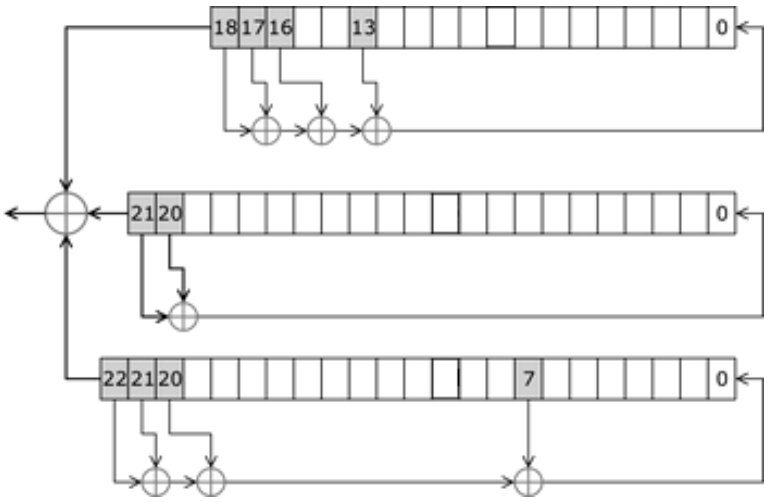
Şekil 2.6'daki örnekte ise "101" başlangıç durumu için DGÖY periyodik bir dizi üretmektedir. Ancak "111" başlangıç durumu için DGÖY başladığı duruma geri dönmemektedir. Bu nedenle durum diyagramı bir süre sonra periyodik bir çevrime otursa da çıkıştaki dizi yukarıdaki anlamda periyodik olmayacaktır.

Yukarıdaki iki örnekten anlaşılacağı üzere ötelemeli yazıcının başlangıç durumunun ve geribesleme fonksiyonu seçimi, oluşturulan dizilerin periyodunun ne kadar uzun olacağı bakımından son derece önemlidir.

2.4. A5 Algoritması

GSM (Global System for Mobile Communications) bugünün dünyasında en geniş şekilde kullanılan kablosuz haberleşme teknolojisidir. GSM'de güvenli haberleşme için birçok kriptografik algoritma kullanılır. A3 doğrulama (authentication) algoritması, A8 ise anahtar paylaşım (key agreement) algoritmasıdır. İki algoritma da abone kimlik modülü kartı (SIM card – Subscriber Identity Module card) içindedir ve bu yüzden operatör kontrolündedir [3]. A5 algoritması ise bu algoritmaların aksine cep telefonun içinde gerçeklenmektedir. A5/1, A5/2, A5/3 olmak üzere türleri mevcuttur. Fakat dünyada genellikle A5/1 algoritması kullanılır. Bu yüzden bu çalışmada A5/1 algoritması anlatılacaktır. Anlatımda A5 ile kastedilen A5/1 algoritmasıdır.

A5, 1980'li yıllarda büyük bir gizlilik içinde geliştirilmiştir. Algoritmanın özellikleri hiçbir zaman için açıklanmamıştır. Buna rağmen, 1999 yılında M. Bricenco, I. Goldberg ve D. Wagner tersine mühendislik (reverse engineering) ile algoritmanın özelliklerini bulmayı başarmışlardır.



Şekil 2. 7 A5'in genel yapısı

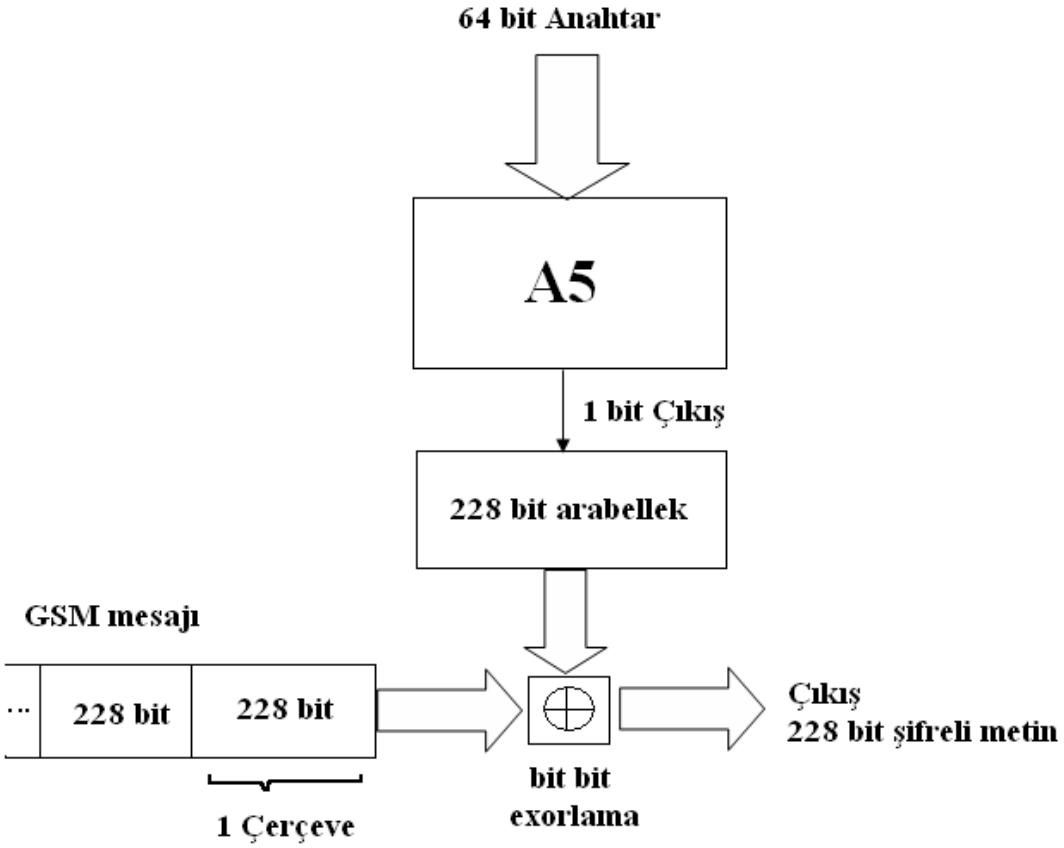
A5'in genel yapısı Şekil 2.7'de görüldüğü gibi 3 adet doğrusal geribeslemeli ötelemeli yazıcıdan oluşmaktadır. İlk DGÖY 19, ikincisi 22, üçüncüsü ise 23 bitten oluşmaktadır. DGÖY'lardaki bitler her saat darbesiyle en anlamlı bit tarafına doğru kayarlar. Bu kayma sonucu oluşan en az anlamlı hücredeki boşluğa ise geribesleme fonksiyonu sonucu oluşan bit yerleştirilir. DGÖY'dan çıkan en anlamlı bitler ise birbirleriyle exorlanarak çıkış üretilir. DGÖY'larda kullanılan geribesleme fonksiyonları Tablo 2.1'de gösterilmiştir.

Tablo 2. 1 A5'deki geribesleme fonksiyonları

1. DGÖY	$f_1(x_{18}, x_{17}, x_{16,} x_{13}) = x_{18} \oplus x_{17} \oplus x_{16} \oplus x_{13} \oplus x_{13}$
2. DGÖY	$f_2(x_{21}, x_{20}) = x_{21} \oplus x_{20}$
3. DGÖY	$f_3(x_{22}, x_{21}, x_{20,} x_7) = x_{22} \oplus x_{21} \oplus x_{20} \oplus x_7$

Tablo 2.1'den de anlaşılacağı üzere birinci ve üçüncü DGÖY'larda dörder bit, ikinci DGÖY'da ise iki bit geribesleme fonksiyonlarında kullanılır.

Genel olarak bakıldığında her DGÖY kendi içinde geribesleme fonksiyonuna göre çalışmaktadır. Öteleme sonucu dışarı çıkan en anlamlı bit ise şifrelemede kullanılacaktır. Şifreleme için kullanılacak kayar anahtar dizisi, bu üç DGÖY'dan çıkan en anlamlı bitlerin exorlanmasıyla oluşur.



Şekil 2. 8 A5'in cep telefonunda kullanım şeması

A5 algoritmasının anahtar uzunluğu DGÖY'ların toplam hücre sayısı olan 64'e eşittir. GSM çerçevesi her iki yönde de 114 bit olmak üzere toplam 228 bitten oluşur. Şekil 2.8'de belirtildiği üzere A5'in çıkışındaki 228 bit dizi arabellekte tutulur. GSM mesajının (açık metin) bir çerçevesini oluşturan 228 bit ile arabellek bit bit exorlanır ve çıkışta 228 bit şifreli metin elde edilir.

3. SAHADA PROGRAMLANABİLİR KAPI DİZİLERİ

Sahada programlanabilir kapı dizileri, programlanabilir mantıksal bloklar ve bu bloklar arasında programlanabilir ara bağlantılar içeren sayısal tümdevrelerdir. Kısaca, FPGA olarak (Field Programmable Gate Arrays) ifade edilir.

3.1. Programlanabilir Devre Elemanlarının Gelişimi

Programlanabilen yapılar, uygulamaya yönelik olarak programlanabilen, genel amaçlı tümleştirilmiş devrelerdir. Günümüzde sayısal donanım tasarımında anahtar rol oynayan programlanabilir yapıların ilk ve uzun bir süre yaygın olarak kullanılan türü programlanabilir salt okunur belleklerdir (PROM – Programmable Read Only Memory). PROM'lar bir kez programlanabilen iki boyutlu bellek elemanlarından oluşmuş yapılardır. PROM'lar Boole fonksiyonlarının gerçeklenmesinde kullanılabilseler de, yaygın olarak mikroişlemcili yapılarda bellek elemanı olarak kullanılır. PROM'ların programlanabilme özelliği geliştirilerek silinebilir programlanabilir salt okunur bellek (EPROM – Erasable Programmable Read Only Memory) ve elektriksel olarak silinebilir programlanabilir salt okunur bellekler (EEPROM – Electrically Erasable Programmable Read Only Memory) üretilmeye başlanmıştır.

Programlanabilir yapıların sonraki türleri "programlanabilir sayısal devre"lerdir (PLD – Programmable Logic Device). PLD'ler "programlanabilir dizi lojiği" (PAL – Programmable Array Logic) ve "programlanabilir lojik dizi" (PLA - Programmable Logic Array) olmak üzere iki kısma ayrılırlar. Bir PLA, VE ile VEYA matrislerinden oluşur. PLA içerisindeki VE ve VEYA matrislerinin her ikisinin de programlanabilir olması üretim maliyetlerini ve yollandırma gecikmelerini arttırmaktadır. PAL yapısında ise programlanabilir VE matrisi ve sabit bir VEYA matrisi bulunur. Bu özelliğiyle PAL, PLA'dan daha ucuz ve performansı daha iyidir. PAL ve PLA yapıları basit programlanabilir sayısal devreler (SPLD – Simple PLD) olarak da adlandırılır. PLD'lerin dezavantajı bir

fonksiyonu çarpımlar toplamı biçiminde gerçekleştirdiklerinden, yüksek çarpım terimleri içeren fonksiyonları gerçekleyememeleridir [4].

SPLD'lerin sınırlı kapasiteleri daha yüksek kapasiteli programlanabilir devreler olan CPLD'lerin (Complex PLD) doğmasına neden olmuştur. CPLD'ler, SPLD benzeri birçok bloğun bir araya gelmesiyle oluşmuş yapılardır.

3.2. FPGA'in Ortaya Çıkışındaki Sebepler

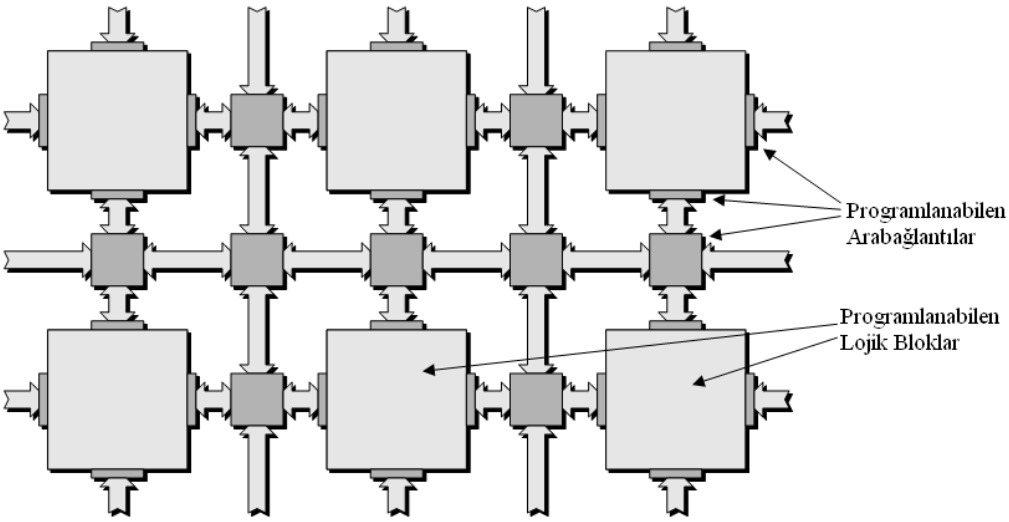
1980'lerde sayısal tümdevre teknolojileri arasında bir boşluk olduğu gün yüzüne çıkmıştı. Bir tarafta SPLD ve CPLD gibi programlanabilen ve hızlı tasarım sürelerine sahip mantıksal elemanlar bulunuyordu, fakat bu elemanlar büyük ve karmaşık fonksiyonları gerçekleyemiyordu. Diğer tarafta da uygulamaya yönelik tümdevreler (ASIC – Application Specific Integrated Circuit) vardı. ASIC çok büyük ve karmaşık fonksiyonları gerçekler fakat çok pahalıya üretilirler ve tasarım süreleri de daha uzundur. Ek olarak bir kez tasarlandıklarında bir daha programlanarak tasarım değiştirilemez. Sayısal tümdevre eleman çeşitleri arasındaki bu boşluğu 1984 yılında Xilinx firması FPGA'i üreterek kapatmıştır [5].

FPGA ilk olarak ortaya çıktığında, sadece orta seviyedeki mantıksal işlevleri yerine getirebiliyordu. 1990'lı yıllardan sonra FPGA teknolojisindeki gelişmeler sayesinde FPGA'in kullanım alanları genişlemiştir. FPGA genel olarak ASIC tasarımların prototipi olarak veya yeni algoritmaların gerçeklenmesinde fiziksel bir düzenek olarak kullanılır. Fakat son yıllarda düşük geliştirme maliyetleri ve kısa zamanda piyasaya sürebilme imkânı sayesinde son ürün olarak da bulunma olanağına kavuşmuştur [5].

FPGA genel olarak dört ana piyasa kesimine rakip olmuştur. Bunlar ASIC, sayısal işaret işleme, gömülü mikrokontrolörler ve haberleşmenin fiziksel katmanıdır. Bunların dışında FPGA yeniden yapılandırılabilir hesaplama (Reconfigurable Computing) adıyla yeni bir alan da oluşturmuştur. Bu alanda şirketler algoritmalarını yazılımdan FPGA'e geçirerek algoritmaların işleyişlerini hızlandırmaktadırlar. Bu sayede donanım benzetimlerinden, kriptoloji analizlerine varıncaya kadar birçok uygulama gerçekleştirilmektedir.

3.3. FPGA Mimarisi

FPGA'ler son kullanıcı tarafından hiçbir üretim basamağına gerek duyulmadan, istenilen sayısal işlevleri yerine getirecek şekilde programlanabilen yapılardır. Genel yapısı Şekil 3.1'de gösterilmiştir. FPGA üç temel bloktan oluşur; lojik bloklar, giriş çıkış blokları ve bağlantı blokları. Bu blokların dışında aritmetik işlem blokları ve bellek blokları gibi özel bloklar içeren FPGA yapıları da mevcuttur.

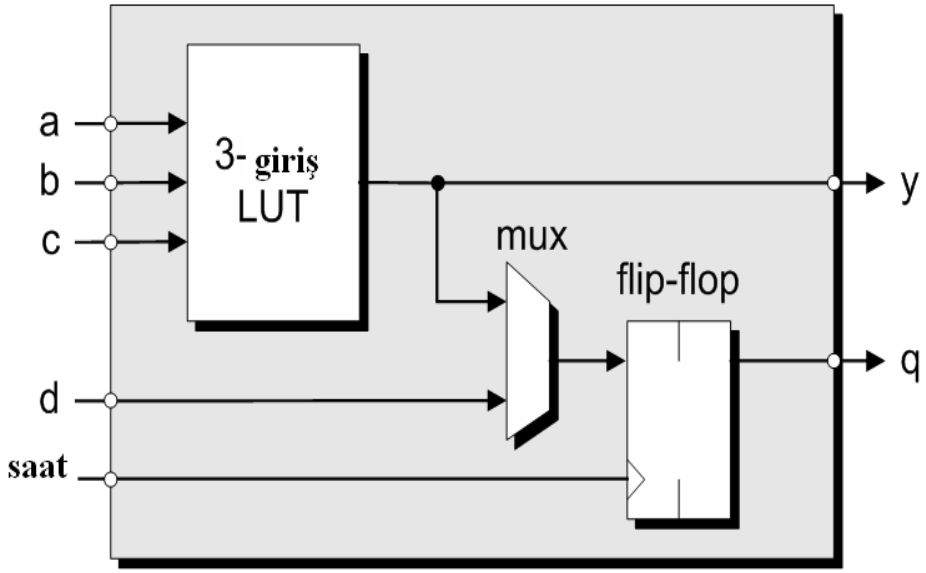


Şekil 3. 1 FPGA genel yapısı

Lojik bloklar Boole fonksiyonlarının gerçeklendiği yapılardır. Küçük taneli (fine-grain) ve kaba taneli (coarse - grain) olarak adlandırılan iki sınıfa ayrılırlar [6]. Bu sınıflandırmada; lojik bloğun oluşumunda kullanılan tranzistor sayısı, lojik bloğun gerçekleyebileceği Boole fonksiyonu sayısı veya lojik bloğun giriş çıkış sayısı, büyüklük ölçüsü olarak kullanılabilir.

Küçük taneli lojik bloklar, genellikle iki girişli bir lojik kapıya veya birkaç girişli çoklayıcıya eşlik eden saklama elemanından oluşur. Bu bloklar karmaşık lojik fonksiyonların gerçeklenmesinde yüksek verimli yapı taşları olarak kullanılır. Ancak, bu gerçekleme sırasında kullanılan bağlantı kanallarının ve programlanabilir anahtarların sayısı oldukça fazladır. Kaba taneli lojik blokların yapıları çok çeşitlilik göstermelerine karşın yaygın olarak doğruluk tablosu veya çoklayıcı gibi daha büyük yapılardan ve saklama elemanlarından oluşur. Kaba taneli lojik blok örneği Şekil 3.2'de gösterilmiştir. Bu tür lojik bloklar, karmaşık lojik fonksiyonların gerçeklenmesinde küçük boyutlu lojik bloklar kadar verimli olmasalar da,

gerçekleme sırasında kullanılan bağlantı kanallarının ve programlanabilir anahtarların sayısı daha azdır.



Şekil 3. 2 Kaba taneli lojik blok örneği

Bağlantı blokları; lojik bloklar ile giriş çıkış blokları arasındaki bağlantıyı sağlayan yapılardır. Bu yapılar, yollandırma kanalları ve programlanabilir anahtarlardan oluşur. Yollandırma kanalları, farklı uzaklıktaki lojik blokların en etkin şekilde bağlanması amacıyla farklı uzunlukta tasarlanır. FPGA içinde gerçeklenebilecek lojik fonksiyon miktarı, yollandırma kanallarının sayısı ile doğru orantılıdır. Yollandırma kanallarının sayısı az olan FPGA yapılarında, lojik blok sayısı fazla olsa bile, gerçeklenebilecek lojik fonksiyon sayısı, lojik blok sayısına oranla kısıtlıdır. Programlanabilir anahtarlar; statik RAM hücresi ile kontrol edilen geçiş tranzistoru, antifuse, EPROM ve EEPROM teknolojileri kullanılarak oluşturulur.

3.4. FPGA'lerin Programlama Teknolojileri

FPGA'lerin programlanabilme özellikleri, içerdikleri programlanabilir anahtarlardan ve lojik bloklardan gelmektedir. Programlanabilir anahtarın gerçeklenme teknolojilerine göre FPGA'lerin programlanma teknolojileri belirlenir.

Programlanabilir anahtar matrisleri üretildikleri teknolojiden bağımsız olarak açık veya kapalı olmak üzere iki konumda bulunabilirler. Programlanabilir anahtarların istenilen özellikleri aşağıdaki gibi sıralanabilir:

- Yarı iletken üzerinde en küçük alanı kaplamaları
- İletim durumundayken küçük bir direnç ve kesim durumundayken de yüksek bir direnç göstermeleri
- İşlev gördükleri noktalarda yollandırma kanallarına minimum kapasite getirmeleri
- Çok sayıda programlanabilir anahtarlar entegre içinde güvenilir bir şekilde gerçeklenebilmeli [6].

3.4.1. Statik RAM Programlama Teknolojisi

SRAM programlama teknolojisinde, programlanabilir bağlantılar SRAM hücresi tarafından kontrol edilen geçiş tranzistoru, iletim kapısı veya çoklayıcı yapısında gerçeklenmektedir. SRAM hücreleri uçucu olduklarından, bu teknolojide FPGA'lerin tümdevreye her gerilim uygulandığında yeniden programlanması gerekmektedir. Programlama verisinin dış bellekte saklanması gerekliliği, bu verinin kopyalanmasını önlemek amacıyla, çeşitli şifreleme yöntemleri kullanılmasını zorunlu kılmıştır [6].

SRAM hücrelerinin yarı iletken üzerinde kapladığı alan büyük olmasına rağmen, FPGA'lerin sistem üzerindeyken tekrar programlanabilmesi, bu teknoloji için büyük bir üstünlük sağlamaktadır.

3.4.2. ANTIFUSE Programlama Teknolojisi

Antifuse teknolojisinde, FPGA programlanmadan önce, yollandırma kanalları arasındaki bağlantılar kurulmamış durumdadır. Programlama sırasında uygulanan gerilimle gerekli bağlantılar oluşturulmuş olur.

Programlama için gerekli gerilimin entegre içinde dağıtılmasını sağlayan tranzistorlar yarıiletken üzerinde geniş alan kaplamalarına karşın, diğer teknolojilerle karşılaştırıldığında antifuse'ler daha küçük bir alana ihtiyaç duyarlar.

Antifuse programlama teknolojisini kullanarak programlanan FPGA'lar bir kez programlanabilme özelliğine sahip olduklarından ilk örnek üretimi için pahalı bir çözüm olmaktadır.

3.4.3. EPROM ve EEPROM Programlama Teknolojisi

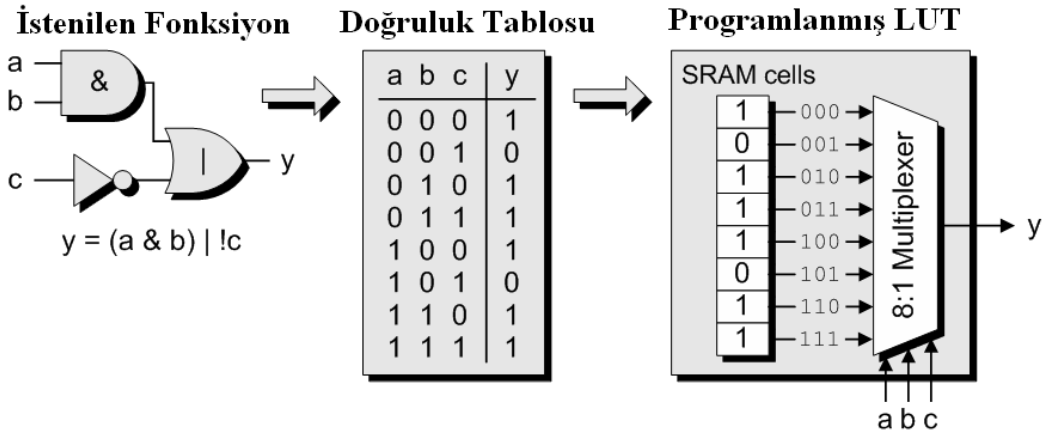
EPROM ve EEPROM programlamada kullanılan yapı EPROM belleklerde kullanılan yapıya benzer. Bir EPROM tranzistorda, MOS tranzistordan farklı olarak seçme geçidi (select gate) ve yüzen geçit (floating gate) olmak üzere iki geçit bulunur. Yüzen geçidin hiçbir elektriksel bağlantısı yoktur ve programlanmadığı durumda üzerinde yük bulunmaz. Tranzistor programlanmak istendiğinde, kaynak ve savak arasında bir akım akıtılır. Bu, geçit ile dielektrik arasında yük tutulmasını ve tranzistorun sürekli iletimde kalmasını sağlar. Yüzen geçitteki yükün boşaltılmasıyla tranzistor yeniden programlanabilir. Yüzen geçitten yükün boşaltılması, elektriksel veya mor ötesi ışıkla gerçekleştirilebilir.

3.5. FPGA'lerın Lojik Hücre Yapısı

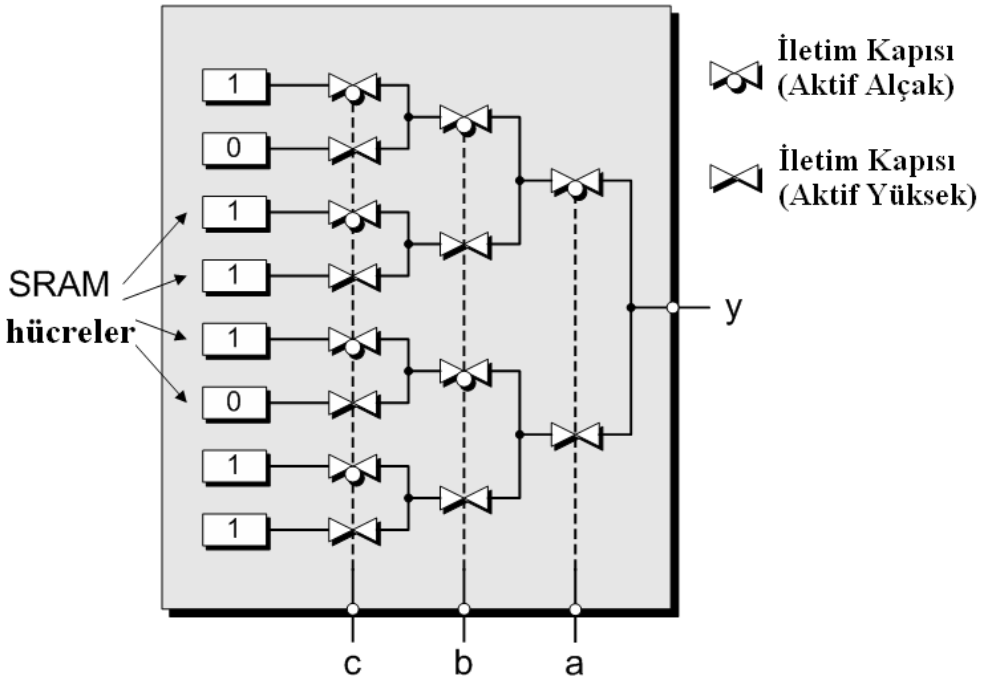
FPGA'ların hücre yapısı doğruluk tablosu tabanlı veya çoklayıcı tabanlı olmak üzere iki sınıfta incelenebilir.

3.5.1. Doğruluk Tablosu Tabanlı Yapı

Doğruluk tablosu tabanlı yapının temel bloğu, LUT (Look Up Table) adı verilen ve m (m>1) değişkenli her Boole fonksiyonunu gerçekleyen bir devredir. Bu yapı statik RAM ile gerçekleştirilir. Şekil 3.3'de üç değişkenli bir fonksiyon verilmiştir. Bu fonksiyonun doğruluk tablosu çıkarılmıştır. Doğruluk tablosuna göre fonksiyon sonuçları SRAM hücrelerindedir. Çoklayıcı sayesinde değişken girişlere karşı olan fonksiyonun cevabı direkt olarak verilmektedir [5]. Şekil 3.3'de kullanılan çoğullayıcı basitleştirilmiş gösterimdir. Şekil 3.4'de programlanmış LUT'un daha gerçekçi modeli gösterilmiştir.



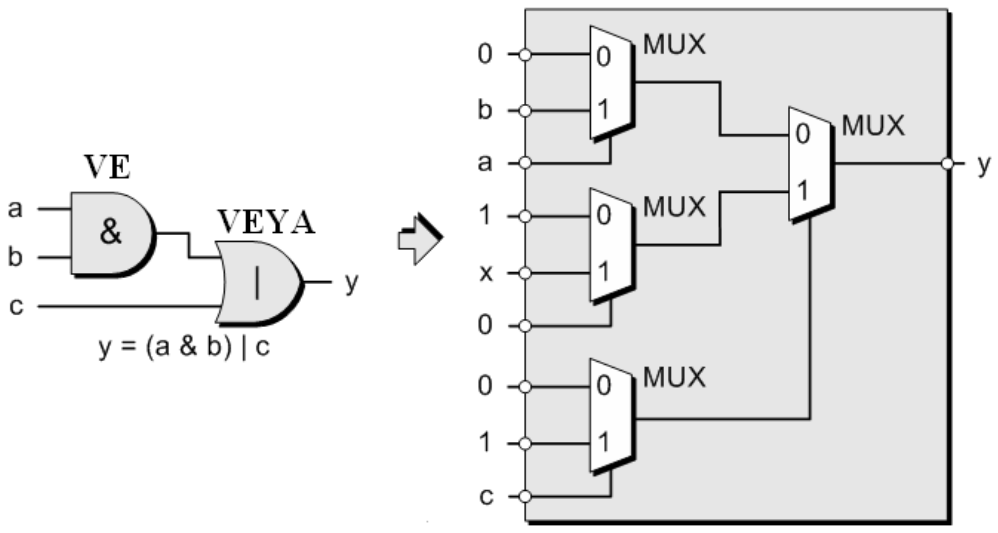
Şekil 3. 3 Bir fonksiyona göre düzenlenmiş LUT'un basit gösterimi



Şekil 3. 4 Programlanmış LUT'un daha gerçekçi gösterimi

3.5.2. Çoklayıcı Tabanlı Yapı

Çoklayıcı tabanlı yapının temel bloğu çeşitli konfigürasyonlardan ve olabildiğince az VE ve VEYA gibi lojik kapılardan oluşur. Bu yapıdaki FPGA'ların içinde latch ve flip-flop bulunmadığından çoklayıcı kullanılarak bu elemanların gerçeklenmesi gerekmektedir. Şekil 3.5'de bir lojik fonksiyonun çoklayıcı tabanlı yapı kullanılarak gerçeklenmesi gösterilmiştir [5].



Şekil 3. 5 Bir lojik fonksiyonun çoklayıcı tabanlı yapı kullanılarak gerçeklenmesi

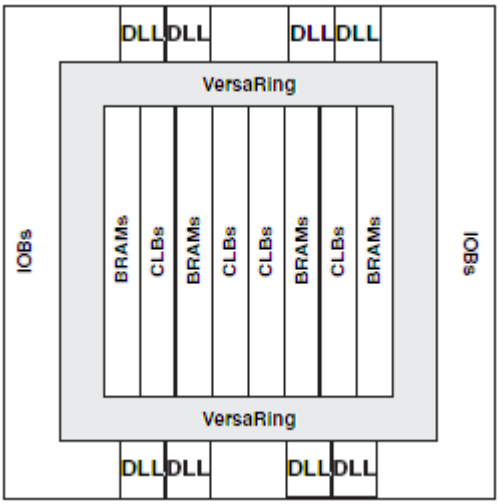
3.6. Bitirme Ödevinde Kullanılan FPGA Özellikleri

Bitirme ödevi için kullanılan FPGA, Xilinx firmasının ürettiği Virtex-E (XCV1000E) dir. SRAM programlama teknolojisine, doğruluk tablosu tabanlı yapıya sahiptir. Nicel özellikleri Tablo 3.1'de verilmiştir [7].

Tablo 3. 1 Virtex -E XCV1000E eleman miktarları

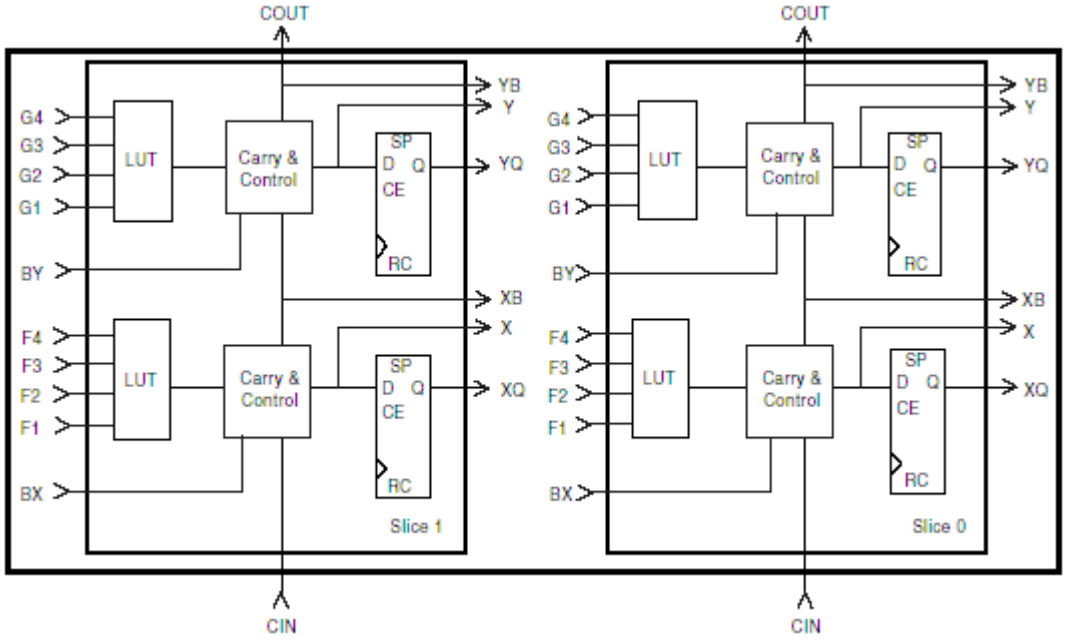
Sistem Kapıları	Lojik Kapılar	CLB Dizile-ri	Lojik Hücre-ler	Diferan-siyel I/O Çiftleri	Kulla-nıcı I/O	Blok RAM bit	Dağıl-mış RAM bit
1569178	331776	64X96	27648	281	660	393216	393216

Virtex–E; ayarlanabilen lojik bloklar (CLB – Configurable Logic Block), CLB etrafını saran giriş çıkış blokları (IOB – Input Output Block) ve programlanabilir ara bağlantılardan oluşur. Genel yönlendirme matrisleri (GRM - General Routing Matrix) düşey ve yatay ara bağlantıların kesiştiği noktalarda anahtarlama görevini görür. Her CLB; aynı zamanda GRM'ye bağlanmasını sağlayan VersaBlock™ içine yuvalanmıştır [7]. Genel yapı Şekil 3.6'da görülmektedir.



Şekil 3. 6 Virtex-E genel yapısı

Her lojik blok 4 giriş fonksiyonu üretecinden, elde lojiğinden ve bellek elemanından oluşur. Her CLB Şekil 3.7'de görüleceği gibi iki dilime ayrılmış toplam dört lojik bloktan oluşur. LUT içindeki doğruluk tablosu 4 bitten (m=4) oluşur. Dolayısıyla 16 (2^m) SRAM hücresine de sahiptir.



Şekil 3. 7 Virtex-E CLB içyapısı

4. A5 TASARIM VE GERÇEKLEMESİ

4.1. FPGA Kullanılarak Gerçeklenecek Devrelerin Tasarım Süreci

Tasarım süreci, gerçeklenecek devre fonksiyonlarının, sözle veya şematik olarak tanımlanması olarak başlar. Sözle tanımlamada, genellikle VHDL veya Verilog gibi yüksek seviyeli donanım tanımlama dilleri kullanılır. Şematik tanımlamada ise birçok firma tarafından geliştirilmiş şematik çizim programlarından faydalanılır. Tanımlama ne şekilde olursa olsun, derleme işlemi sonrasında tüm tanımlamalar, standart bağlantı listesi (netlist) biçimine çevrilir. Yapılan tanımlamaların, istenilen fonksiyonları yerine getirip getirmediği davranışsal benzetim (behavioral simulation) yapılarak test edilir. Benzetim sonucuna göre, tanımlamada gerekli değişiklikler yapılır.

Tanımlamanın doğruluğu test edildikten sonra lojik sentezleme adımına geçilir. Lojik sentezleme sırasında, tanımlanan fonksiyonların en iyi şekilde gerçeklenebilmesi için gerekli lojik indirgemeler yapılır. Lojik iyileştirme (logic optimization) FPGA'a bağlı değildir. Lojik iyileştirmenin ardından, elde edilen lojik fonksiyonların FPGA içindeki lojik bloklara eşleştirilmesi işlemi yapılır. Teknoloji haritalaması (Technology Mapping) olarak da adlandırılan bu işlem sırasında, haritalama işlemini yapan program, kullanılacak lojik blok sayısını ve lojik derinliği azaltmayı hedefleyen bir yol izler.

Sentezleme sonrası yerleştirme ve yollandırma (Placement and Routing) işlemleri yapılır. Bu adımda devre fonksiyonları ile eşleştirilmiş lojik bloklar FPGA içine yerleştirilir ve bu bloklar arasındaki bağlantılar oluşturulur. Yerleştirme ve yollandırma adımının tamamlanması ile birlikte, devreye ait gerçek kapı ve bağlantı gecikmeleri elde edilmiş olur. Bu veriler kullanılarak devrenin gerçeğe çok yakın benzetimleri yapılabilir ve hız açısından kritik olan yollar saptanabilir [4].

Yerleştirme ve yollandırma sonrası benzetimlerde istenilen sonuçlar elde edildikten sonra FPGA'i programlama işlemine geçilir. Programlama işlemi FPGA'deki programlanabilir elemanların yapısına göre farklılık gösterir. Her FPGA

üretici firma, kullandığı teknolojiye bağlı olarak kendi programlama sürecini belirlemiştir.

FPGA kullanılarak yapılan tasarım sürecinin her adımında, bilgisayar destekli tasarım programları kullanılır. Tüm adımları gerçekleyen paket programlar olduğu gibi her adım için ayrı tasarlanmış programlar da mevcuttur.

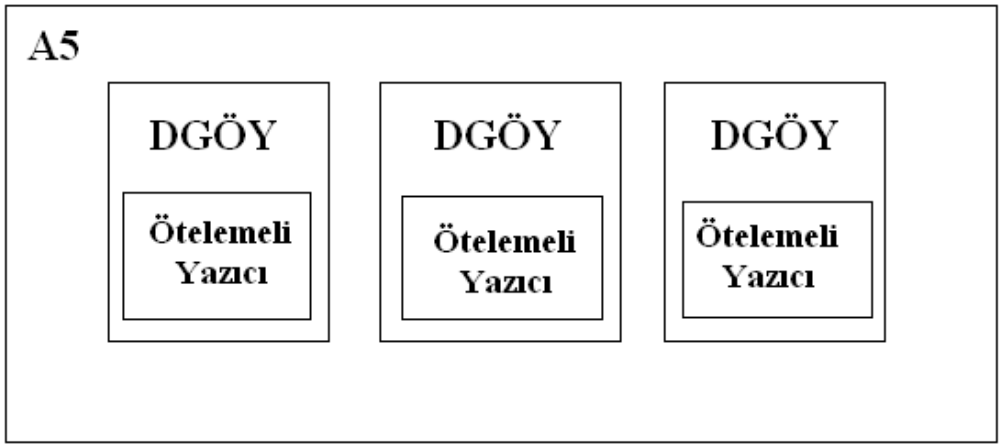
Bu çalışmadaki tasarımda, sistem VHDL kullanılarak sözle tanımlanmıştır. Benzetimler için Modelsim, sentezleme içinse Xilinx ISE programı kullanılmıştır.

4.2. A5 Algoritmasının Tasarımı

A5 algoritması Bölüm 2.4'de belirtildiği gibi 3 adet farklı bit uzunluklarına sahip ötelemeli yazıcılardan oluşmaktadır. Bu yazıcılardan da farklı hücrelerdeki bitler exorlanmakta, en düşük anlamlı hücreye yerleştirilmekte ve en anlamlı bitler ise birbiriyle exorlanıp çıkışa verilmekteydi.

Tasarımda en başta en genel haliyle çift yönlü, parametreli ötelemeli yazıcı tasarlanmıştır. Daha sonra bu yazıcı kullanılarak hücre sayısı ve exorlanacak bitler parametreli olarak oluşturulan genel halde DGÖY tasarlanmıştır. Daha sonra bu DGÖY parametrelerine A5 algoritmasındaki 3 DGÖY'nın özellikleri girilerek A5 algoritması tasarlanmıştır.

Tasarımda ötelemeli yazıcıyı, DGÖY'yı ve A5'i tanımlamak için toplam 3 adet, giriş ve çıkışları olan varlıklar (entity) kullanılmıştır. Şekil 4.1'de gösterildiği gibi, A5 içinde toplam 3 DGÖY için 3 DGÖY parçası (component) ve bu DGÖY'lerin içinde de bir ötelemeli yazıcı parçası kullanılmıştır.



Şekil 4. 1 A5 tasarım mantığı

4.2.1. En Genel Halde Ötelemeli Yazıcı Tasarımı

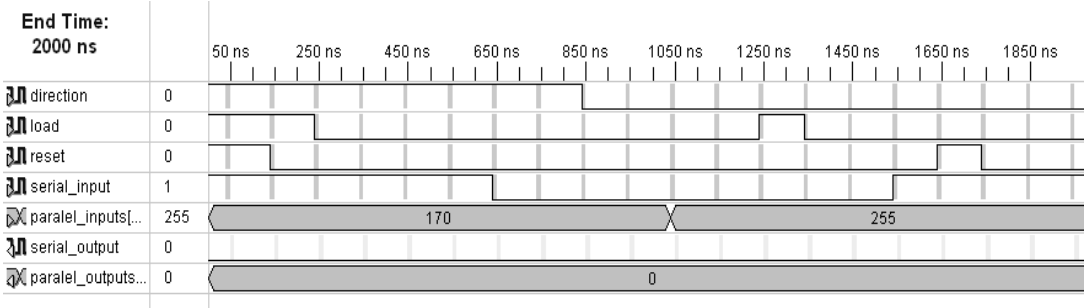
Paralel ve seri girişli; paralel ve seri çıkışlı; öteleme yönü değişken; bit sayısı değişken olan en genel halde ötelemeli yazıcı tasarlanmıştır. Devrenin giriş ve çıkışları programda yazıldığı şekliyle şöyledir:

- Clock (saat)
- Reset (sıfırlama)
- Load (yükleme)
- Serial_input (seri giriş)
- Paralel_inputs (paralel giriş – N bit)
- Direction (öteleme yönü belirleme)
- Paralel_outputs (paralal çıkış – N bit)
- Serial_output (seri çıkış)

Saat ile eşzamanlı çalışan bu devrede yazıcının hücrelerine paralel yükleme yapabilmek için 'Load' lojik 1 yapılır, N bitlik paralel hattaki değer ötelemeli yazıcıya kaydedilir. Ötelemeli yazıcıya girecek seri giriş ise öteleme yönüne göre ya en anlamlı hücreye ya da en düşük anlamlı hücreye girer. Direction (yön) lojik 1 ise ötelemeli yazıcı sola doğru kayar ve en anlamlı bit dışarıya çıkar, en düşük anlamlı bite giriş olur; direction lojik 0 ise tam tersi olur.

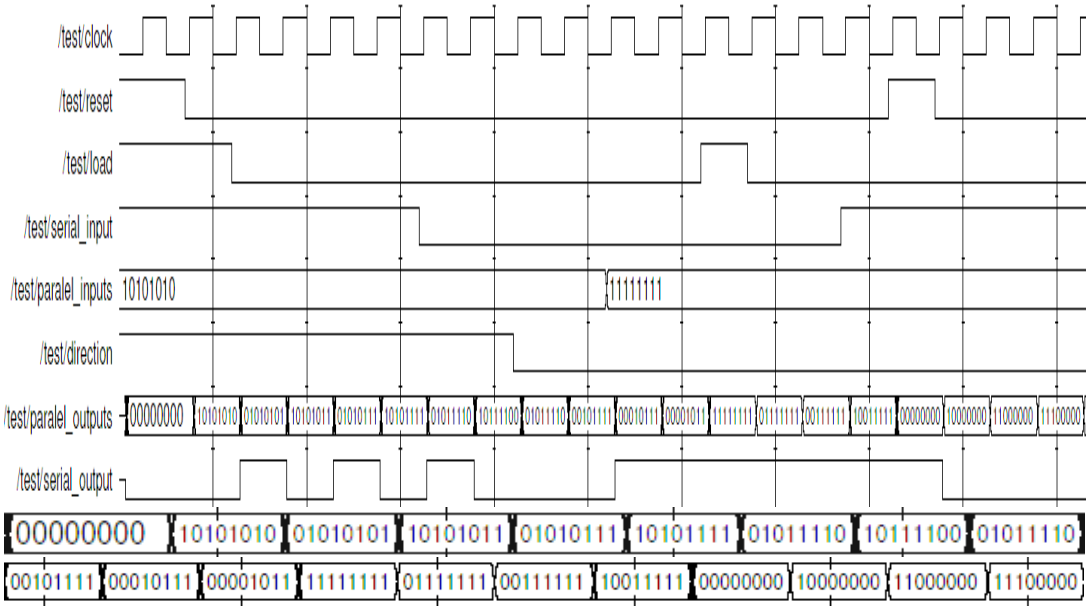
Ötelemeli yazıcı için yazılan kod başarı ile sentezlenmiştir (N=8 bit için). Davranışsal benzetim, teknoloji haritalaması sonrası benzetim, yerleştirme ve yollandırma sonrası benzetimleri yapılmıştır. Sentezlemede ötelemeli yazıcının çalışabileceği en yüksek frekans 242.365MHz (4.126ns periyotlu) çıkmıştır.

Yerleştirme ve yollandırma sonrası benzetim için Şekil 4.2'deki gibi bir test düzeneği (test bench) oluşturulmuştur. Saatin lojik 1 seviyesinde kalma süresi ve lojik 0 seviyesinde kalma süresi 50ns olarak verilmiştir.



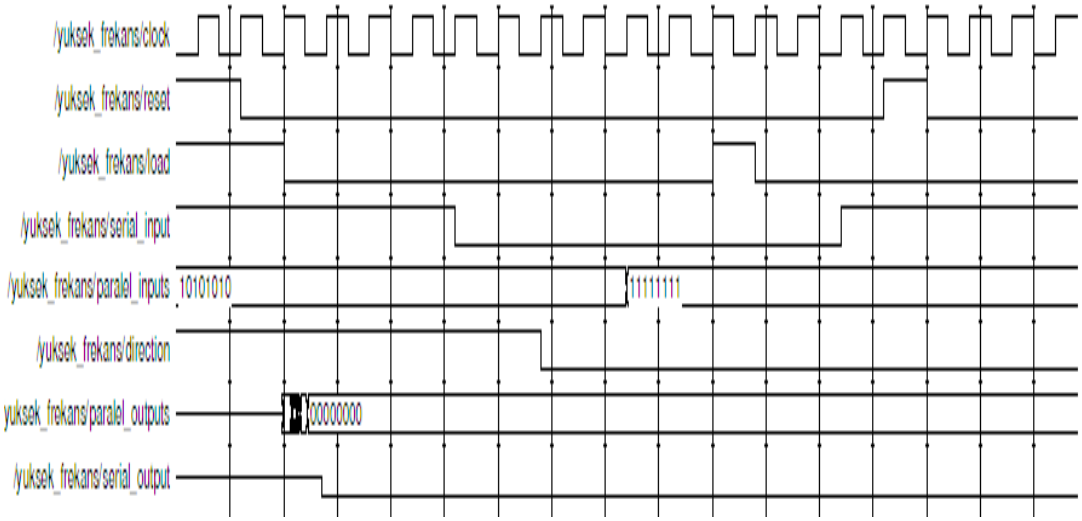
Şekil 4. 2 En genel halde ötelemeli yazıcı için test düzeneği

Bu durumda yapılan yerleştirme ve yollandırma sonrası benzetimine göre (bu benzetim, FPGA modeli hesaba katılmış en son yapılan benzetimdir) elde edilen sonuç Şekil 4.3'deki gibidir. Şeklin alt kısmında paralel çıkış değerleri rahat okunabilmesi için büyültülmüş olarak verilmiştir.



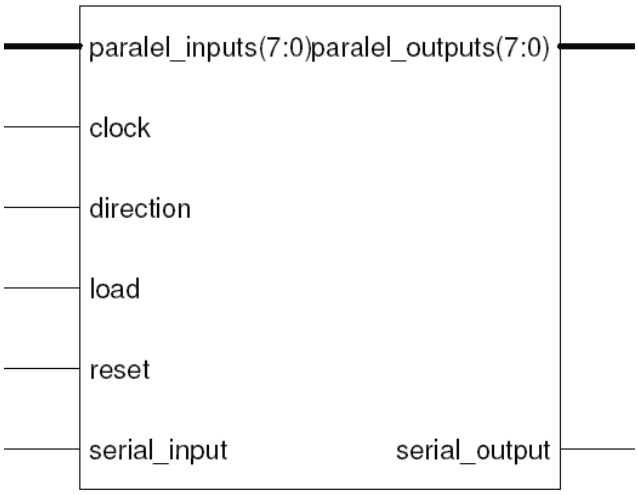
Şekil 4. 3 En genel halde ötelemeli yazıcının yerleştirme ve yollandırma sonrası benzetimi

Eğer saat periyodu ötelemeli yazıcının minimum periyodu olan 4.126 ns den daha az bir süre olursa sistemin yanlış çalışacağı beklenir. Bunu görmek için saat periyodu 4ns olarak atanmıştır. Elde edilen benzetim sonucu Şekil 4.4'de belirtildiği üzere, tasarlanılan ötelemeli yazıcının davranışı görülmemektedir. Dolayısıyla saat periyodunu 4.126ns'den büyük seçmek gereklidir.

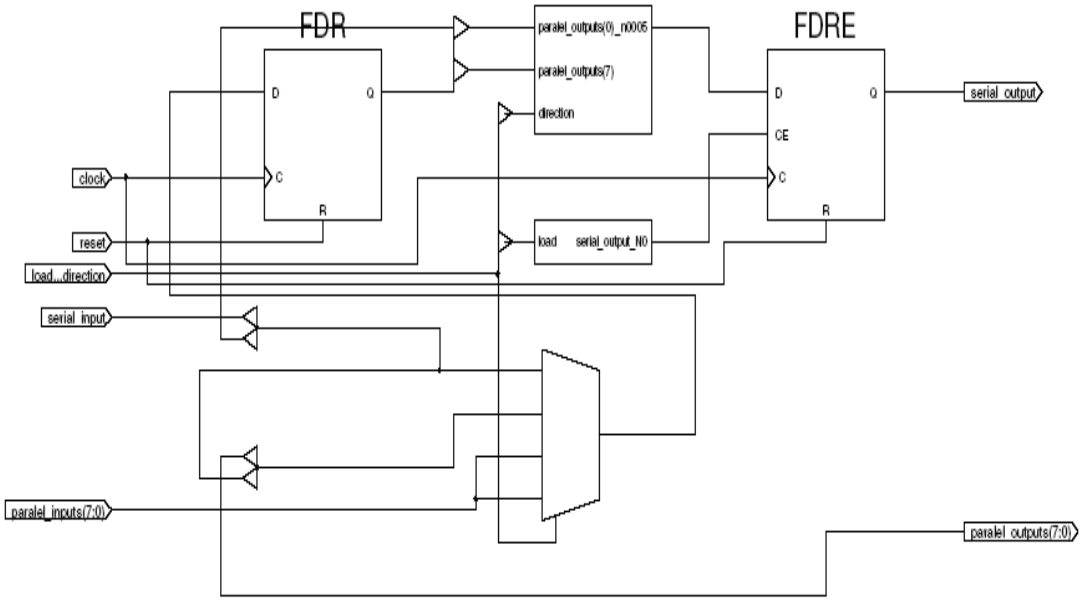


Şekil 4. 4 En yüksek frekanstan büyük saat darbesi uyguladığı zamanki durum

8 bitlik ötelemeli yazıcının genel yapısı Şekil 4.5'de, yazıcı transfer lojiği (Register Transfer Logic) yapısı ise Şekil 4.6'da verilmiştir.

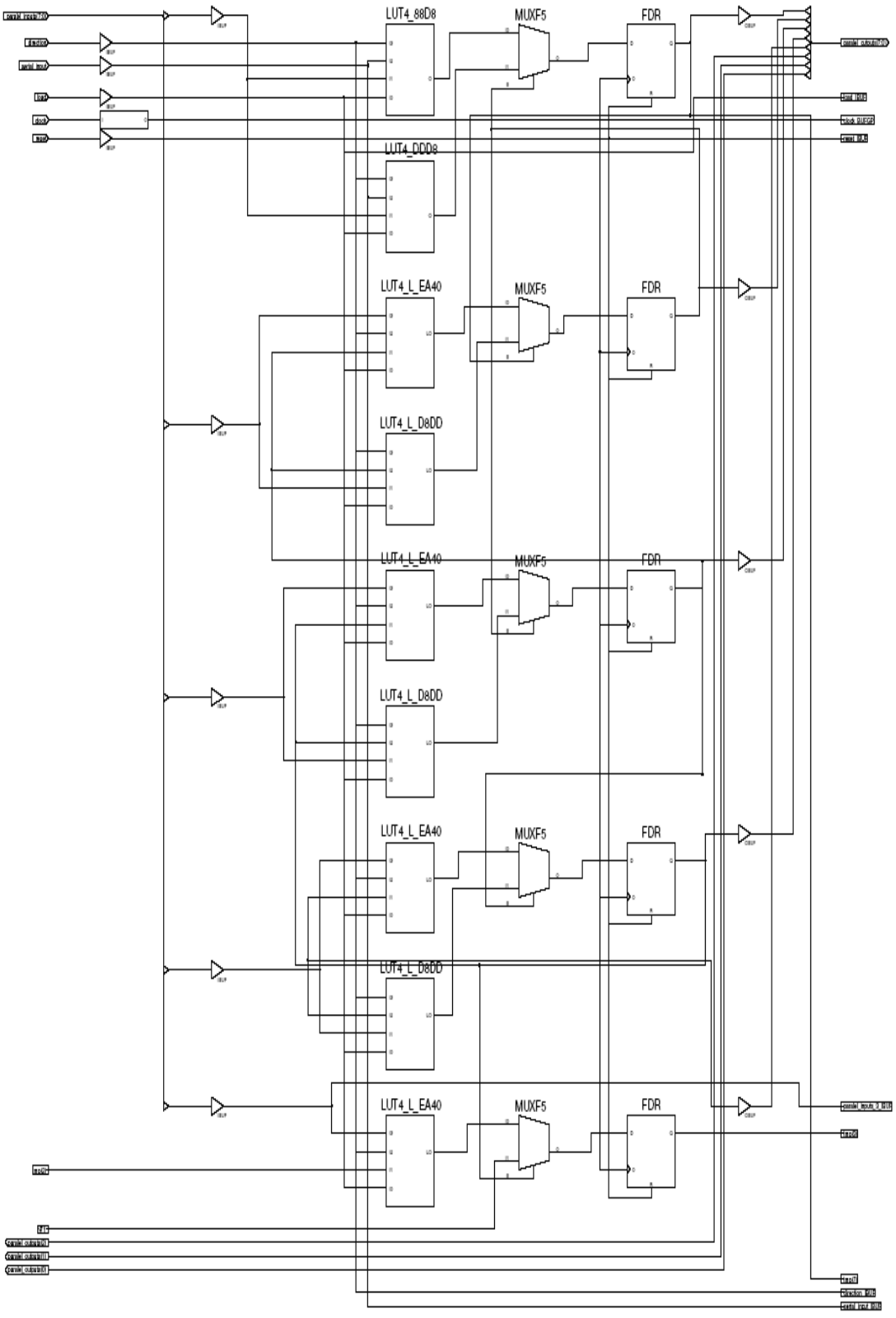


Şekil 4. 5 Ötelemeli yazıcının genel yapısı

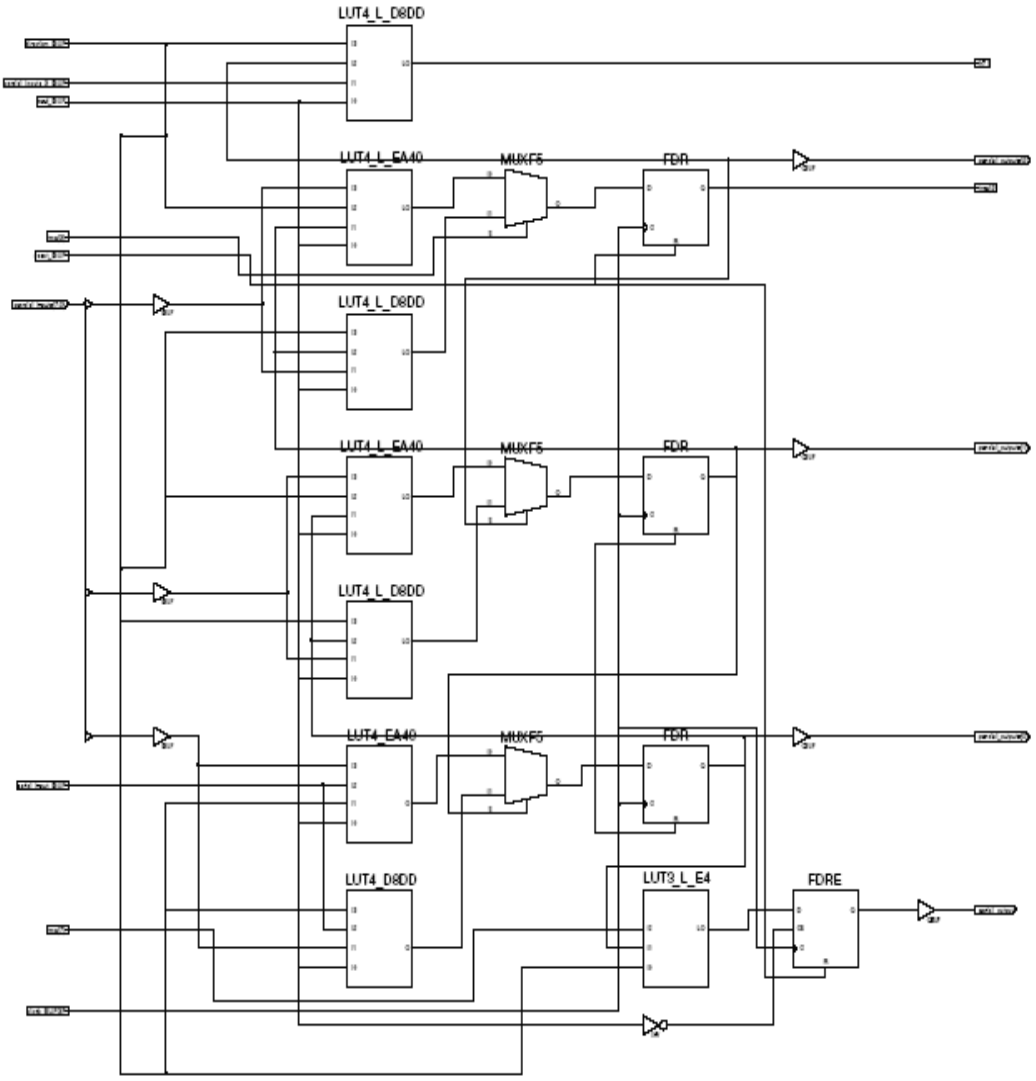


Şekil 4. 6 Ötelemeli yazıcının yazıcı transfer lojiği yapısı

Ötelemeli yazıcının FPGA içinde yerleşimi Şekil 4.7a ve 4.7b'deki teknoloji yapısında gösterilmiştir. Bu yapıda FPGA içinde kullanılan doğruluk tabloları da (LUT) görülmektedir.

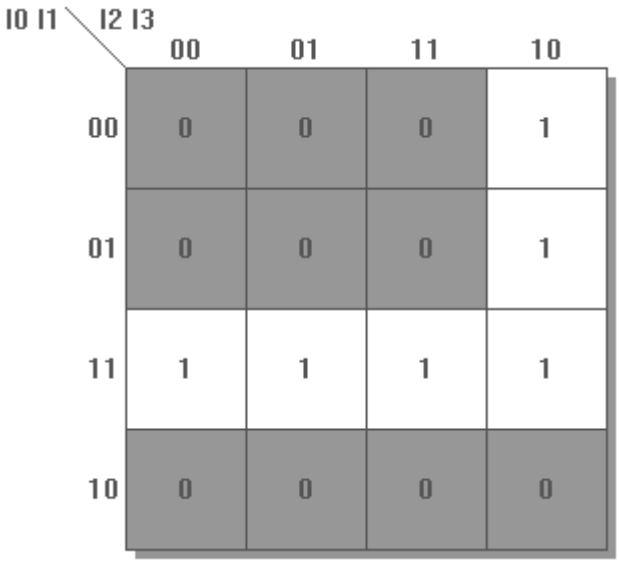


Şekil 4. 7a Ötelemeli yazıcının teknoloji yapısı (ilk kısım)

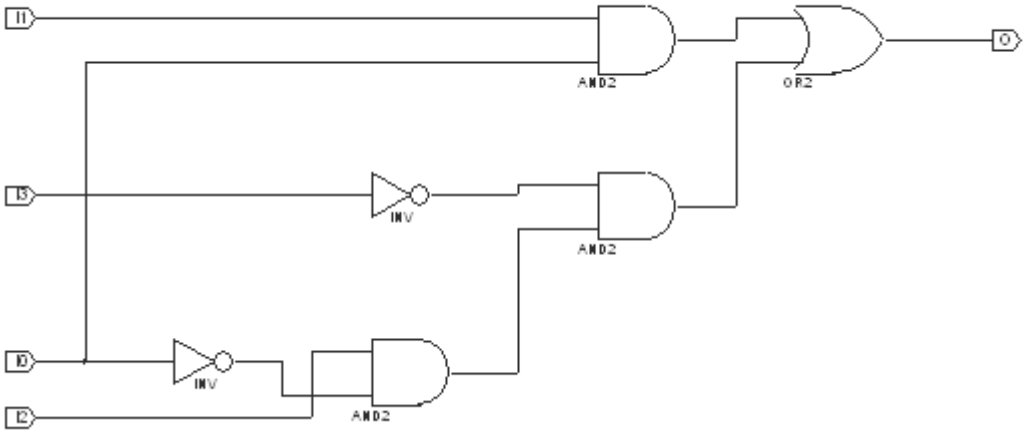


Şekil 4.7b Ötelemeli yazıcının teknoloji yapısı (ikinci kısım)

Kullanılan FPGA'de doğruluk tablolarının hepsi 4 girişlidir. Yukarıdaki yapıda da bu görülmektedir. Bu doğruluk tabloları belirli girişlere karşı belirli çıkışları üretecek bir yapıya sahiptir. Örneğin Şekil 4.7a'da bulunan en üstteki doğruluk tablosunun Karnaugh diyagramı ve bu diyagramı gerçekleyen kapılar Şekil 4.8 ve Şekil 4.9'da gösterilmiştir.



Şekil 4.8 İlgili LUT'un Karnough diyagramı



Şekil 4.9 İlgili LUT'u gerçekleyen kapılar

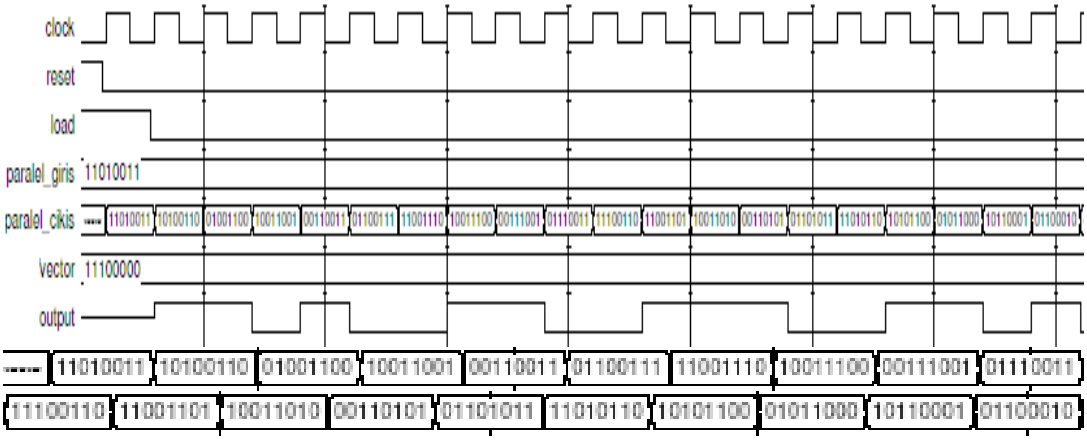
4.2.2. DGÖY tasarımı

Tasarlanan DGÖY'da her ötelemede en anlamlı bit dışarıya çıkar. Hangi bitlerin exorlanacağı bir vektörle belirlenir. DGÖY ile aynı bit sayısına sahip oluşturulan bu vektörün lojik 1 olan hücreleri DGÖY'daki aynı konumda bulunan hücreleri işaret eder. DGÖY'daki bu hücreler birbirleriyle exorlanarak DGÖY'nın en düşük anlamlı hücresine yerleştirilir. Daha önce tasarlanan ötelemeli yazıcı parçası (component) kullanılır. DGÖY'nın giriş ve çıkışları şunlardır:

- Clock (saat)
- Reset (sıfırlama)
- Load (yükleme)
- Paralel_giris (paralel giriş – N bit)
- Output (seri çıkış)

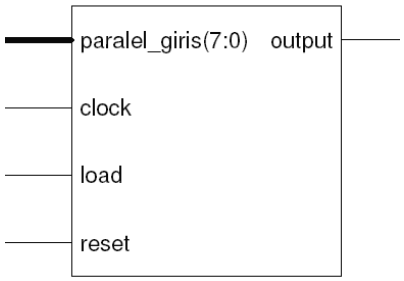
DGÖY'nın ilk tasarımında sonuçların doğruluğunun görülebilmesi için paralel çıkış da bir çıkış olarak tanımlanmıştır. İlk tasarımda hangi bitlerin exorlanacağını gösteren vektör ise giriş olarak belirtilmiştir. Fakat son tasarımda bu vektör genel olarak (generic) tanımlanmıştır, dolayısıyla girişte yer almamaktadır.

Şekil 4.10'da DGÖY'nin doğru çalıştığı, paralel çıkış sonuçlarından görüldükten sonra kullanılması gereksiz olan bu çıkış kaldırılmıştır. Şekilde ötelemeli yazıcın en anlamlı 3 biti exorlanıp, bir sonraki saat darbesinde en düşük anlamlı hücreye yerleştiği görülmektedir. Seri çıkışın ise en anlamlı bit olduğu görülmektedir.



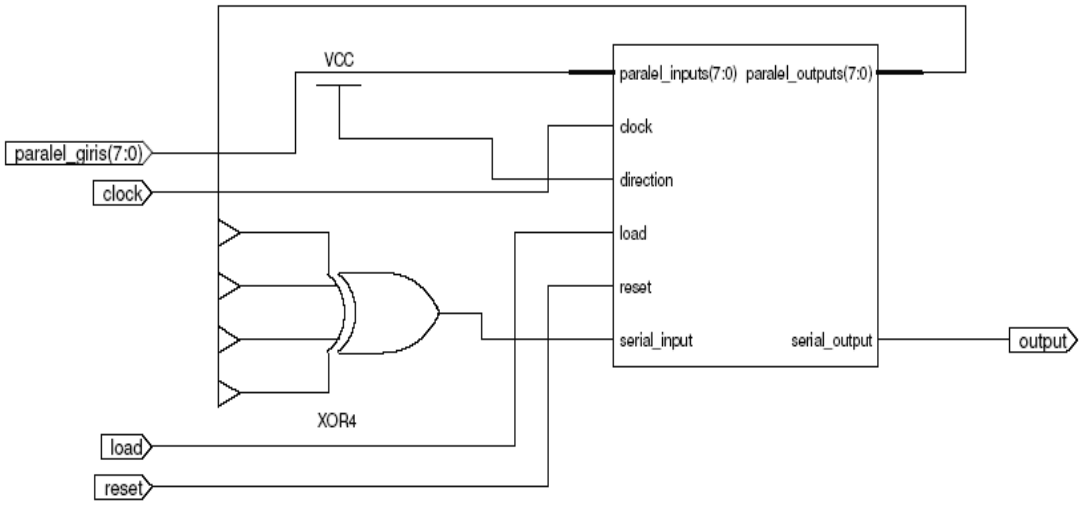
Şekil 4.10 DGÖY'nin benzetim sonucu

En son düzenlemeler yapılıp tasarlanan 8 bitlik DGÖY'nın genel yapısı Şekil 4.11'de görülmektedir.



Şekil 4.11 DGÖY'nin genel yapısı

Şekil 4.12'de ise DGÖY'nın lojik yapısı görülmektedir. Şekilden anlaşılacağı üzere tasarımda en genel halde yazıcı kullanılmıştır. En genel yazıcının yön (direction) girişine sürekli lojik 1 (V_{cc}) verilmektedir, çünkü DGÖY sürekli sol tarafa doğru kayacaktır. Genel ötelemeli yazıcının seri girişine, paralel çıkışın bazı işlemlerden geçtikten sonra (istenilen bitler exorlandıktan sonra) girdiği görülmektedir. Dolayısıyla geribesleme olduğu açıktır.

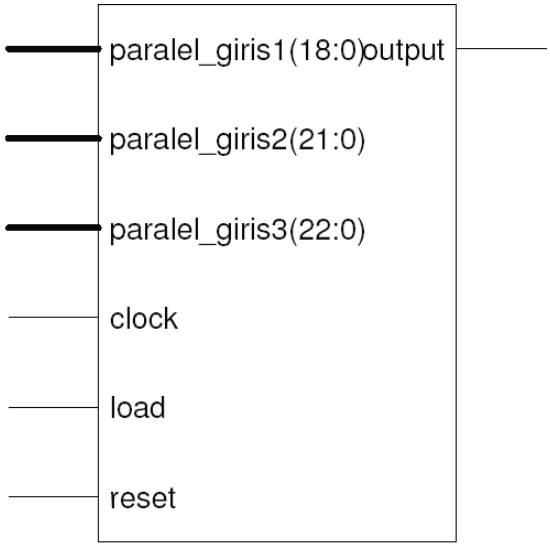


Şekil 4.12 DGÖY'nin lojik yapısı

4.2.3. A5 Tasarımı

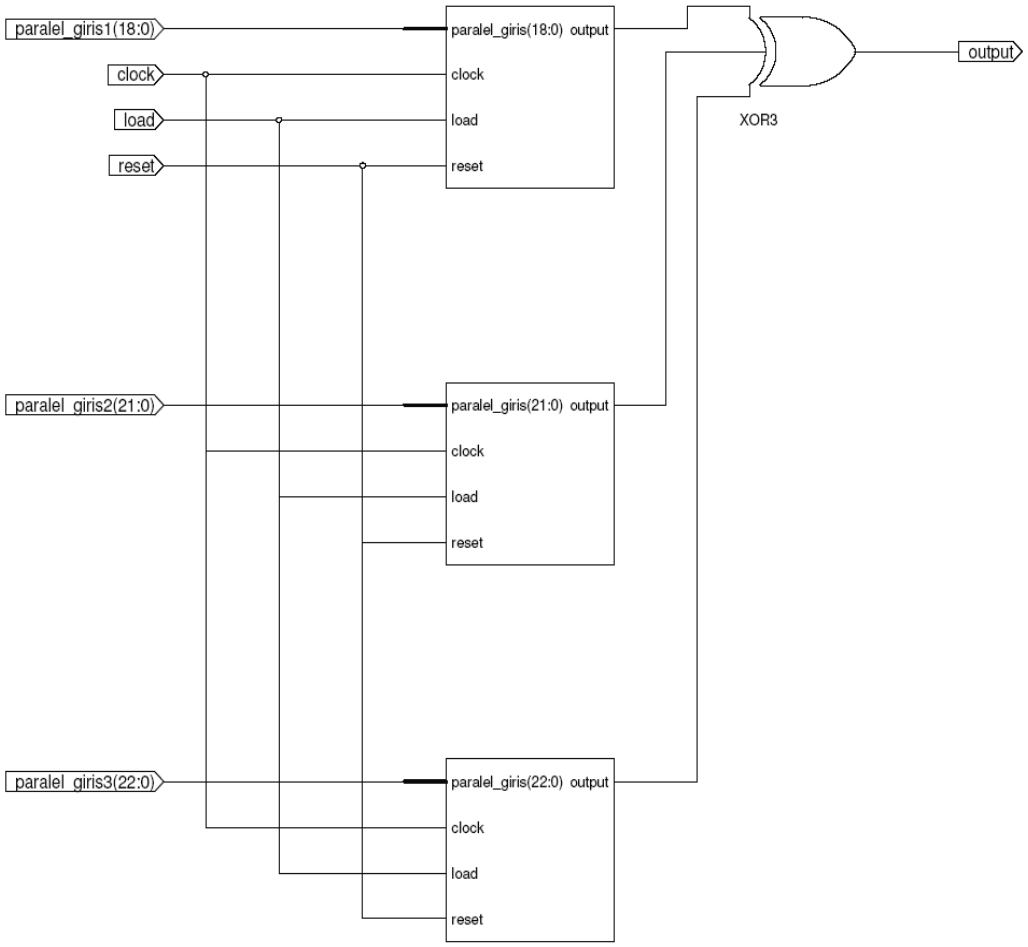
A5 algoritmasını gerçeklemek için yapılan bu adımda 3 adet DGÖY parçası kullanılır. Bu DGÖY'ların bit sayısının kaç olacağı ve hangi bitlerinin exorlanacağı bu kısımda yazılmıştır. Şekil 4.13'den de görüleceği gibi A5'in altı giriş, 1 çıkışı vardır. Bunlar:

- Paralel_giris1 (19 bitlik ilk DGÖY'nın girişi)
- Paralel_giris2 (22 bitlik ikinci DGÖY'nın girişi)
- Paralel_giris3 (23 bitlik üçüncü DGÖY'nın girişi)
- Clock (saat)
- Load (yükleme işareti girişi)
- Reset (sıfırlama)
- Output (şifrelemede kullanılacak kayar anahtar dizisi)



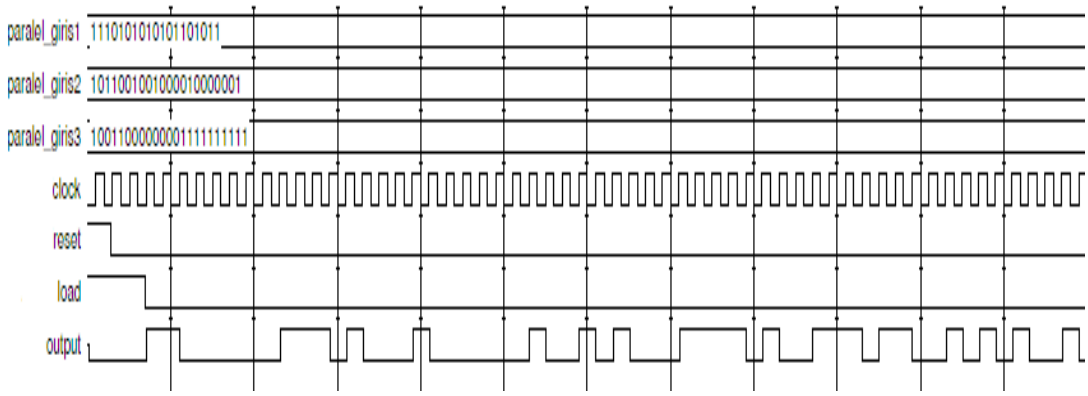
Şekil 4.13 A5'in genel yapısı

Şekil 4.14'deki lojik yapıdan görüleceği üzere 3 DGÖY'nın çıkışı exorlanmakta ve kayar anahtar dizisi üretilmektedir.

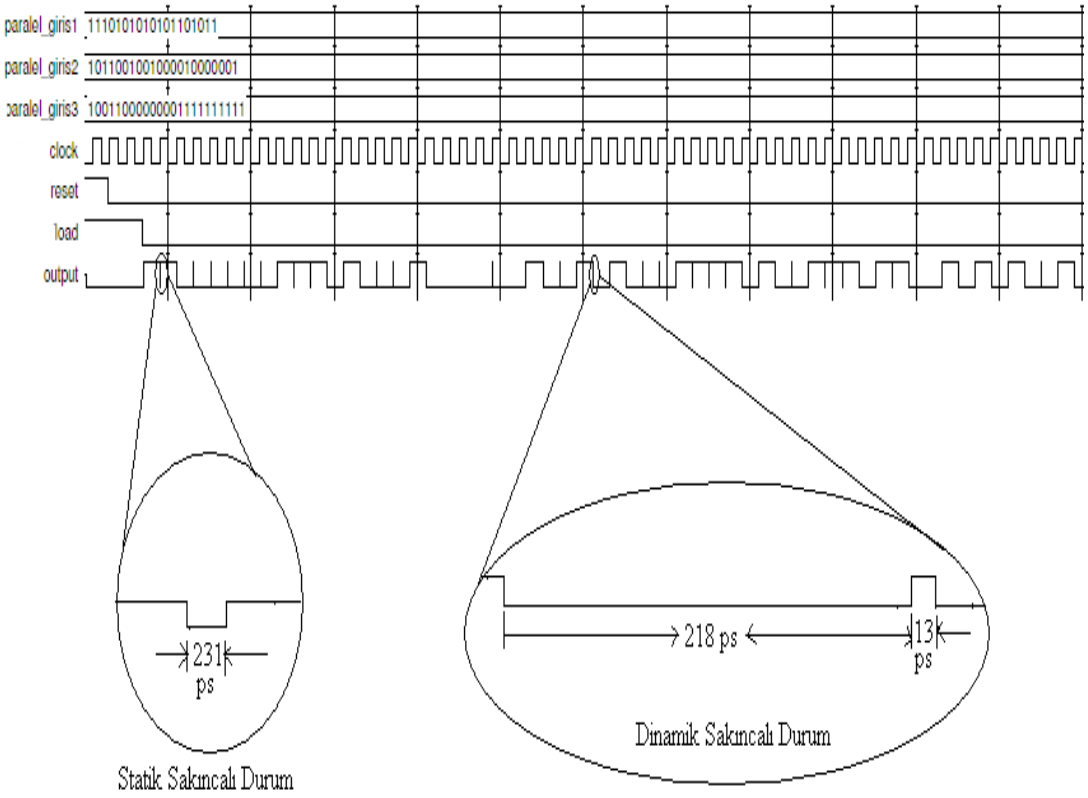


Şekil 4.14 A5'in lojik yapısı

A5'in teknoloji haritalaması sonrası benzetimi Şekil 4.15'de verilmiştir. FPGA elemanlarının yerleştirilmesi yapılıp, gerçekçi gecikmeler uygulanınca, yerleştirme ve yollandırma sonrası benzetimi Şekil 4.16'da görülmektedir. Şekilde görüleceği üzere piko saniyeler mertebesinde çıkış işaretinde sıçramalar oluşmuştur. Bu sıçramalara sakıncalı durum (hazard) denir. Fakat bu sakıncalı durumlar eşzamanlı (senkron) devrenin işleyişi bakımından bir sorun olarak görülmez. Konuyu daha iyi anlamak için sakıncalı durumlara değinmekte fayda vardır.



Şekil 4.15 A5'in teknoloji haritalaması sonrası benzetimi sonucu



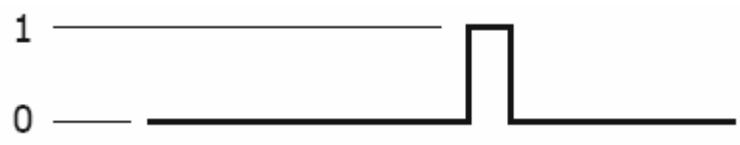
Şekil 4.16 A5'in yerleştirme ve yollandırma sonrası benzetimi sonucu

4.2.3.1. Sakıncalı Durumlar

Sakıncalı durumlar, değişik yolların değişik yayılma gecikmelerine neden olmasından ötürü devre çıkışında ortaya çıkan ve anahtarlamaların yol açtığı istenmeyen geçici durumlardır [8]. Sakıncalı durumlar kombinezonsal devrelerde meydana geldiğinde geçici olarak hatalı çıkışlar üretirler. Bu durum asenkron ardışıl devrelerde hatalı bir kararlı duruma geçişe yol açabilir.

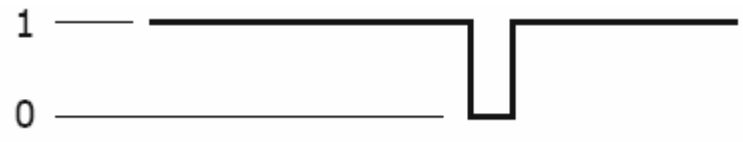
- Statik sakıncalı durum:

Eğer bir devrenin çıkışı, lojik olarak sıfır durumundayken giriş değişimlerine bağlı olarak yine sıfırda kalması gerekirse, fakat çok kısa süreliğine lojik bir olursa bu duruma statik 0 sakıncalı durum denir. (Şekil 4.17)



Şekil 4.17 Statik 0 sakıncalı durum

Eğer bir devrenin çıkışı, lojik olarak bir durumundayken giriş değişimlerine bağlı olarak yine birde kalması gerekirse, fakat çok kısa süreliğine lojik sıfır olursa bu duruma statik 1 sakıncalı durum denir.(Şekil 4.18)



Şekil 4.18 Statik 1 sakıncalı durum

- Dinamik sakıncalı durum:

Eğer çıkış sıfırdan bire (veya birden sıfıra) değişmesi öngörülür, fakat çıkış üç veya daha fazla değişirse bu duruma dinamik sakıncalı durum denir.(Şekil 4.19)

Şekil 4.19 Dinamik sakıncalı durum

- Temel sakıncalı durum:

Temel sakıncalı durum, aynı girişten kaynaklanan ve iki veya daha çok sayıdaki hat üzerinde oluşan farklı büyüklükteki gecikmelerden kaynaklanır. Temel sakıncalı durumlar statik sakıncalı durumlar gibi fazladan kapı ekleyerek

giderilemez. Ancak, yaratacakları sorun, yol üzerinde oluşacak gecikme miktarı ayarlanarak düzeltilebilir [8].

Senkron ardışıl devrelerde giriş işaretleri flip flopun setup ve hold zamanlarında sabit olması gerekir. Bu yüzden saat darbelerine bağlı olarak setup ve hold zamanlarının dışında anlık sakıncalı durumların meydana gelmesi bu tür devrelerde önemli değildir [9].

4.2.4. A5'in Doğru Sonuç Verdiğinin Test Edilmesi

A5 algoritması için program yazılıp benzetimi yapılmıştır, fakat doğru sonuç verip vermediğinin hesaplanması gerekir. Bu hesaplamayı elde yapmak çok zordur. Bu yüzden A5 çıkışını üretecek kod yazmak ve iki çıkışı karşılaştırmak gereği ortaya çıkmıştır. Dolayısıyla Matlab programı kullanılarak A5 algoritması için test vektörü oluşturulmuştur. Matlab'da yazılan kodun verdiği çıkış ile çalışma sonucu elde edilen çıkış karşılaştırılmış, aynı sonuçlar elde edilmiştir.

5. SONUÇ

Sahada programlanabilir kapı dizileri ile A5 algoritmasının gerçeklenmesi projesinde, kriptografi hakkında genel bir bilgi verilmiş ve dizi şifreleme üzerine yoğunlaşılmıştır. Bir dizi şifreleme çeşidi olan, cep telefonlarındaki iletişim güvenliğini sağlayan A5 algoritmasının tasarlanması ve Xilinx firmasının Virtex-E XCV1000e model FPGA üzerinde gerçeklenmesi yapılmıştır.

Günümüzde yeni algoritmalar tasarlandığında, bunun donanım üzerinde nasıl davranacağını görmenin, en az zaman alan ve maliyet açısından da zorlamayan yolu sahada programlanabilir kapı dizileri kullanmaktır. Bu projede ise yüz milyonlarca insanın yanında taşıdığı ve içinde A5 algoritmasının varlığından haberi olmayarak kullandığı cep telefonları hedef seçilmiştir.

Sonuç olarak bu çalışmada, hakkında az bilgiye sahip olunan kriptografi ve FPGA üzerine açıklamalarda bulunulmuş, sayısal donanım tasarım ve gerçekleme adımları açıklanmış ve bir güvenlik algoritması olan A5 tasarlanıp, gerçeklenmiştir. Tasarlanan A5 algoritmasının minimum periyodu 3.537ns, maksimum saat frekansı 282.725 MHz, kullandığı doğruluk tablosu sayısı 67 adettir. A5 için kullanılan doğruluk tablosu sayısı projede kullanılan FPGA modelinin toplam doğruluk tablosu sayısının %0.272'sine tekabül etmektedir. Bu durumda algoritmanın FPGA üzerinde az yer kapladığını ve maksimum saat frekansı 282.725MHz olduğu için çalışma hızının yeterince yüksek olduğunu söyleyebiliriz.

KAYNAKLAR

[1] **Apohan, M.,** 1993. Cryptography, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[2] **Savaş, E.,** 1994. Dizi şifreleme sistemleri ve doğrusal karmaşıklık, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[3] **Rechberger, C.,** 2004. Side channel analysis of stream ciphers, *Master's Thesis,* Graz University of Technology Institute of Applied Information Processing and Communications, Graz.

[4] **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.

[5] **Maxfield, C.,** 2004. The design warrior's guide to FPGAs. Elsevier, Amsterdam.

[6] **Topçu, İ. H.,** 2002. Sahada programlanabilir kapı dizileri kullanarak sayısal tasarım kartı gerçeklenmesi, *Yüksek Lisans Tezi*, İ.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.

[7] **DS022**, 2002. Xilinx VirtexE 1.8V Field Programmable Gate Arrays Data Sheet.

[8] **Mano, M.,** 2002. Digital design. Prentice Hall, Upper Saddle River, NJ.

[9] **Brown, S. and Vranesic Z.,** 2000. Fundamentals of Digital Logic with VHDL Design, McGraw-Hill, Boston.

ÖZGEÇMİŞ

1983 yılında doğan Furkan DAYI, Cağaloğlu Anadolu Lisesi'nden 2001 yılında mezun olduktan sonra lisans öğrenimini İTÜ Elektronik Mühendisliği'nde görmeye başladı. 2005 yılında, Darmstadt Teknik Üniversitesi'nde bulunarak lisans öğreniminin bir dönemini Almanya'da geçirdi. Birçok kez, İTÜ Onur ve Yüksek Onur listesine giren Furkan DAYI, 2001 senesinde İTÜ başarı bursuna, 2005 senesinde Nortel Networks başarı bursuna layık görülmüştür.