İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

Klonlanamaz Fonksiyonlar ve Yardımcı Bilgiler Kullanılarak Patent Hakları Korunması, Özgün Algoritma ve Donanımın Güvenliğinin Sağlanması

YÜKSEK LİSANS TEZİ

Burak GÖVEM

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

HAZİRAN 2013

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

Klonlanamaz Fonksiyonlar ve Yardımcı Bilgiler Kullanılarak Patent Hakları Korunması, Özgün Algoritma ve Donanımın Güvenliğinin Sağlanması

YÜKSEK LİSANS TEZİ

Burak GÖVEM
(504091267)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Programı

Tez Danışmanı: Doç. Dr. Müştak Erhan YALÇIN

HAZİRAN 2013

İTÜ, Fen Bilimleri Enstitüsü'nün 504091267 numaralı Yüksek Lisans Öğrencisi **Burak GÖVEM**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"Klonlanamaz Fonksiyonlar ve Yardımcı Bilgiler Kullanılarak Patent Hakları Korunması, Özgün Algoritma ve Donanımın Güvenliğinin Sağlanması"** başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Doç. Dr. Müştak Erhan YALÇIN** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :	**Doç. Dr. Müştak Erhan YALÇIN** İstanbul Teknik Üniversitesi	.....
	Y. Doç. Dr. Metin Yazgı İstanbul Teknik Üniversitesi	.....
	Y. Doç. Dr. Fethullah Karabiber Yıldız Teknik Üniversitesi	.....

Teslim Tarihi : **3 Mayıs 2013**
Savunma Tarihi : **6 Haziran 2013**

ÖNSÖZ

Bu çalışmayı gerçekleştirmemi sağlayan danışman hocam Doç. Dr. Müştak Erhan Yalçın'a teşekkür ederim.

Bu tez 00957.STZ.2011-2 kodlu Sanayi Tezleri Programı (SAN-TEZ) projesi tarafından desteklenmektedir. Projeye verdikleri destek için T.C. Bilim, Sanayi ve Teknoloji Bakanlığı'na ve Vestek Elektronik AR&GE A.Ş.'ye teşşekkür ederim.

Haziran 2013

Burak GÖVEM
Elektronik Mühendisi

İÇİNDEKİLER

Sayfa

KISALTMALAR

AAD : Additional Authentication Data (Ek Kimlik Verisi)
AES : Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)
ALU : Arithmetic Logic Unit (Aritmetik Mantık Ünitesi)
ASIC : Application Specific Integrated Circuit (Uygulamaya Özgü Tümleşik Devre)
BPI PROM : Byte Peripheral Interface Programmable Read Only Memory (Paralel Programlanabilir Salt Okunabilir Bellek)
CBC : Cipher Block Chaining (Şifreleme Bloğu Zincirleme)
CLB : Configurable Logic Block (Yapılandırılabilir Mantık Bloğu)
CPLD : Complex Programmable Logic Device (Karmaşık Programlanabilir Mantık Aygıtı)
CSoC : Configurable System on Chip (Yapılandırılabilir Yonga Üzeri Sistem)
DPR : Dynamic Partial Reconfiguration (Dinamik Kısmi Yinelenebilir Yapılandırılabilirlik)
ECB : Electronic Code Book (Elektronik Kod Defteri)
EDA : Electronic Design Automation (Elektronik Tasarım Otomasyonu)
FF : Flip-Flop (Tutucu)
FPGA : Field Programmable Gate Array (Alanda Programlanabilir Kapı Dizisi)
GAL : Generic Array Logic (Genel Dizi Mantığı)
GCM : Galois Counter Mode (Galois Sayıcı Kipi)
HDL : Hardware Description Language (Donanım Tanımlama Dili)
I/O : Input/Output (Giriş/Çıkış)
IC : Integrated Circuit (Tümleşik Devre)
ICAP : Internal Configuration Access Port (Yapılandırma Devresine Dahili Erişim Kapısı)
IDE : Integrated Development Environment (Tümleşik Geliştirme Ortamı)
ISE : Integrated Software Environment (Tümleşik Yazılım Ortamı)
IP : Intellectual Property (Fikri Mülkiyet)
IV : Initialization Vector (İlklendirme Vektörü)
JTAG : Joint Test Action Group
LUT : Look-Up Table (Başvuru Çizelgesi)
MB : Mega Byte
MHz : Mega Herz
PAL : Programmable Array Logic (Programlanabilir Dizi Mantığı)
PLD : Programmable Logic Device (Programlanabilir Mantık Dizisi)
PLL : Phase Locked Loop (Faz Kilitlemeli Çevrim)
PR : Partial Reconfiguration (Kısmi Yinelenebilir Yapılandırma)
PRM : Partial Reconfigurable Module (Kısmi Yinelenebilir Yapılandırılabilir Modül)
PRR : Partial Reconfigurable Region (Kısmi Yinelenebilir Yapılandırılabilir Bölge)

PUF **:** Physical Unclonable Function (Fiziksel Klonlanamayan Fonksiyon)
RO **:** Ring Oscillator (Halka Osilatör)
SPR **:** Static Partial Reconfiguration (Statik Kısmi Yinelenebilir Yapılandırılabilirlik)
UART **:** Universal Asynchronous Receiver Transmitter (Evrensel Asenkron Alıcı Verici)
VHDL **:** Very high speed integrated circuit Hardware Description Language
(Çok Yüksek Hızlı Tümleşik Devre Donanım Tanımlama Dili)

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

Klonlanamaz Fonksiyonlar ve Yardımcı Bilgiler Kullanılarak Patent Hakları Korunması, Özgün Algoritma ve Donanımın Güvenliğinin Sağlanması

ÖZET

Sahtecilik, günümüzde tekstilden eczacılık ve elektroniğe, gıda sektöründen marka isimlerine kadar endüstrinin hemen her alanında rastlanılan bir vaka haline gelmiştir. Bu durum şirketleri milyonlarca dolar zararlara uğratmakta ve birçok önemli markanın güvenilirliğini ve saygınlığını sarsmaktadır.

Elektronik sektöründe sahtecilik olgusu daha vahim boyutlara ulaşmıştır. Sahte elektronik parça ve yongalara artık hayati önem taşıyan askeri, uzay elektroniği ve tıp elektroniği sektörlerinde bile rastlanmaktadır. Sahtecilik olaylarının bu boyutlara erişmesi sahte elektronik yongalarının tespit edilmesini sağlayacak ekonomik yöntem araştırmalarının hız kazanmasına yol açmaktadır.

Elektronik yonga klonlama yöntemlerinden bazıları, istek fazlası üretim, kopyalama ve tersine mühendislik gibi yöntemlerdir. Günümüzde birçok elektronik tasarım ve üretim şirketleri tasarımlarını diğer üretim yapan şirketlere sipariş verebilmekte veya tasarımlarını IP olarak satabilmektedir. Bu türdeki pazar ve üretim gereksinimleri nedeniyle firmalar tasarımlarını paylaşmak durumunda kalmaktadır. Çalıntı tasarımlar bu şekilde ele geçirilebilmektedir.

Elektronik tasarımların çalınmasını önlemeye yönelik olarak akla gelebilecek ilk yöntemlerden birisi tasarımların şifrelenerek saklanması olabilir. Şifreleme yöntemi bu tür bir probleme çözüm olabilir ancak şifreleme sistemlerinin anahtar yönetimi etraflıca düşünülmeli ve gizli şifreleme anahtarlarının çalınması riski de ortadan kaldırılmalıdır.

Son yıllarda ortaya çıkan fiziksel klonlanamaz fonksiyonlar (PUF) teknolojisi şifreleme anahtarı yönetimi problemine çözüm sunabilmektedir. Üretildiği ortama özgü karakteristik özelliklere bağlı olan PUF verisinin kopyalanması önceden öngörülemeyen ortam parametrelerine bağlı olduğu için pratik olarak mümkün değildir. Elektronik yongalar üzerinde çalışan PUF devrelerinden elde edilen veriler de yongaların üretim süreçlerindeki engellenemeyen idealsizliklere bağlı olarak yongaya özgü klonlanamaz bilgiler olacaktır. PUF verisinin yonga üzerinde elde ediliyor olması şifreleme anahtarının taşınması sorununu da çözecektir.

PUF verisi yongaya özgü üretim parametrelerine bağlı olduğu gibi çalışılan ortam değişkenlerine de bağlıdır. Devrenin çalıştığı ortamın sıcaklık, gerilim ve radyasyon miktarı gibi fiziksel şartlarındaki değişimler PUF verisinin değişmesine yol açabilmektedir. Bu nedenle PUF verisinin değişmeden tekrar tekrar elde edilebilmesi için ham PUF verisi üzerinde hata düzeltme işlemleri yapılmalıdır.

Esnek tasarım imkanı sunmaları nedeniyle gün geçtikçe daha çok kullanım alanı bulan sahada programlanabilir kapı dizileri (FPGA) bu çalışmada da tercih edilmiştir. FPGA'ların son yıllarda sahip oldukları özelliklerden birkaçı, otomatik yeniden

yapılandırma ve kısmi yapılandırmadır (PR). Bu çalışmada PR özelliği, gerçeklenmek istenen güvenlik sisteminin otomatik olarak çalıştırılıp üretim süreçlerine uygun bir tasarım olması amacıyla kullanılmıştır.

Literatürde bulunan PUF çeşitlerinden, FPGA ortamı için uygun bulunan halka osilatör (RO) tabanlı PUF bu çalışmada şifreleme algoritmasının anahtar verisini üretmek için kullanılmıştır.

Şifreleme algoritması olarak, sıkça kullanılan bir standart haline gelmiş Gelişmiş Şifreleme Standardı (AES) tercih edilmiştir. Bu algoritmanın çalışma kipi olarak da paralelleştirmeye uygun olan ve kimlik doğrulama algoritması olarak da işlev gören Galois Sayaç Kipi (GCM) kullanılmıştır.

Protection of Patent Rights,
Maintenance of Original Algorithm and Hardware Security
Using Physical Unclonable Functions and Helper Data

SUMMARY

Counterfeiting is encountered among almost every branch of industry, from pharmacy to electronics, from food to brand names. The increasing trend in counterfeiting damages many company economically and reduce their reliability and status.

The current state in counterfeiting is more destructive nowadays. Counterfeit electronic components and chips can be found at even military and space electronics and medical electronics industry. These serious advance of counterfeiting accelerate the research in the field of detecting counterfeit products economically.

Some methods of counterfeiting electronic chips are overproduction, cloning and reverse engineering. Todays, many fabless electronic design companies make their products to other companies or many of them sell their designs as electronic IPs to other electronic design companies. In such cases, companies can be obliged to share their designs with other companies. Counterfeit designs emerge in such a way.

The basic solution against counterfeiting in electronics industry is ciphering electronic IP designs. However, key management is another problem that is to be handled and the risk of the key to be stolen must be eliminated.

Physical Unclonable Functions (PUF) technology that has been developing in recent years can solve the key storage problem of cryptographic systems. PUF data depends on the static environment parameters that is unavoidable and non-deterministic. As a result, PUF data cannot be cloned practically. PUF circuits on chips rely on the non-ideal nature of process technology (doping differences etc.). Such as, mathematically equivalent timing delay paths on the chip, actually are not equivalent, have a slight difference. These slight differences can be gathered and result in a measurable delay. These delay differences are expected to vary for identical chips and then each chip has its own characteristic PUF data. Extracting PUF data on chip also solves the key transmission problem. In the literature, several PUF circuit types are available, for example Anderson, butterfly, arbiter and ring oscillator (RO) PUFs are well-known. RO-PUF is used in this work. RO-PUF is suitable for FPGA implementations. RO-PUF consists of RO pairs. One bit PUF output is the sign of the difference of the frequency of these two identical ROs. RO pairs must have identical path delays. So the difference is controlled with the characteristics of the chip.

On the other hand, PUF circuits are affected by environmental conditions. For instance, voltage and temperature fluctuations, radiation can change PUF output result from measurement to measurement. This fact causes unreliable PUF outputs and this means cryptographic key cannot be extracted truly on every creation. The reliability problem have to be solved to get a consistent cryptographic key. Error correction codes can be used to stabilize fluctuating PUF data. These error correction codes are generally collect statistical data of PUF output, then raw PUF output is post-processed according

to the calculation results of statistical data. In this manner, an error correction code called labeling technique is used in this work. Labeling technique needs to collect statistical data about PUF outputs for different environmental conditions and for different chips. According to statistical data, each PUF circuit is labeled with a binary number and PUF output is evaluated after measurement according to label. Also, this label does not reveal PUF output.

FPGAs are ubiquitous for especially last decade thanks to their flexibility. In this work, FPGAs are used especially for their reconfigurability utilities. FPGAs are known as further behind the performance of ASIC designs. However depending on the developments in IC design and production technology FPGAs are closing the performance gaps between their ASIC rivals. FPGAs also have many new features in those days, partial reconfiguration is an example of these features. Partial reconfiguration (PR) utility is going to be a common standard for new generation FPGAs. Xilinx Virtex series and new 7 series FPGAs have this PR feature. PR enables exchanging hardware designs in the time domain. In dynamic partial reconfiguration (DPR), rest of the FPGA continues to work, when partial region is reconfiguring. As a result, area and power can be reduced dramatically with programming separately working designs partially.

PR is necessary for this work to automate authentication process. In every power-up of FPGA, before loading actual design, authentication check design deciphers ciphered bitstream of the actual design and if it passes authentication check then deciphered bitstream is loaded on the FPGA otherwise authentication fails and deciphered data is erased, it is not loaded on the FPGA.

In this work, Advanced Encryption Standard (AES) which is a common encryption standard is used as crypto algorithm. AES is used with Galois Counter Mode (GCM) of operation. GCM is a highly parallelable mode of operation which is very suitable for hardware implementations. GCM provides both message confidentiality and authenticity. FPGA bitstream is very sensitive to bit errors, even one bit error can destroy reconfiguration, and also FPGA can be damaged. Therefore, some kind of checksum algorithm have to be used to ensure the error-free bitstream. Authentication feature of GCM also fulfills the checksum task.

Non-volatile memory is used to store bitstreams of the FPGA designs. A BPI PROM on Xilinx ML 505 board is used as non-volatile memory. PROM contains a full bitstream with authentication check design and a partial encrypted bitstream contains ciphered actual design. Full bitstream also called golden design contains a PROM reader module to get partial ciphered bitstream into FPGA. Partial bitstream is deciphered and also authentication check is calculated in parallel.

Deciphered bitstream have to be stored in FPGA due to authentication check lasts till the end of deciphering process. At the end of the authentication check, deciphered bitstream is needed if it passes the authentication check for loading on the FPGA. FPGA has limited internal buffer. All partial bitstream cannot be stored on the FPGA. Therefore, partial bitstream is deciphered as constant size of blocks (2048 block for this application). Every block has its own authentication check and authentication tag stored on BPI PROM. If calculated tag in the FPGA matches the retrieved tag from PROM block passes authentication check and buffered bitstream is sent to reconfiguration controller module to load it through Internal Configuration Access Port (ICAP).

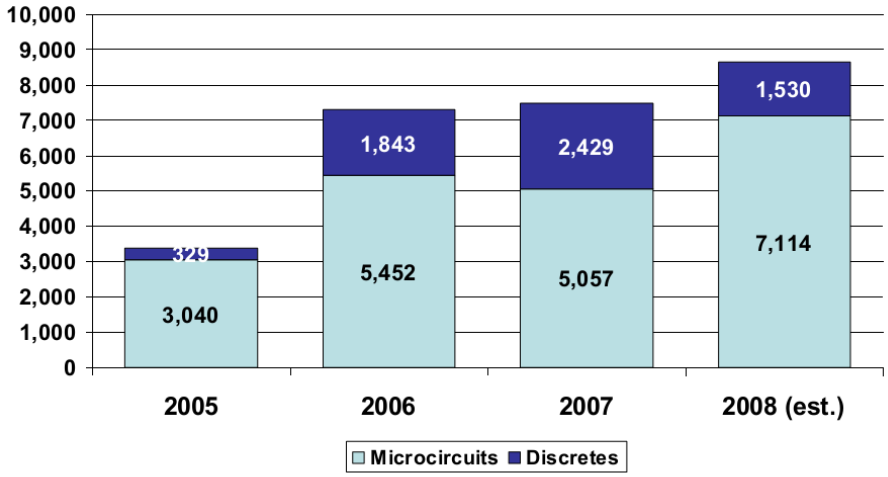
Reconfiguration process is handled with ICAP primitive on FPGA. Reconfiguration process is controlled with user logic called reconfiguration controller module. Since functionality of the PR is important for this work, simple partial modules are used to test the designed system. Switches are used as input and leds are used for output of the PRMs. Input signals are given to EXOR or AND logic gates depending on a select input signal at one PRM, other PRM uses EXNOR or OR logic operations depending on the select input. Piezo audio transducer is also used in the static design to check the operability of the static modules during PR process. During reconfiguration piezo transducer continues playing beep and after a short PR process time output leds are driven by second PRM.

1. GİRİŞ

Dünya üzerinde yapılan sahtecilik veya taklitçilik faaliyetleri her geçen gün daha da artmaktadır. Söz konusu bu çalışmalar belirli bir sektör ya da alan gözetmeksizin her yerde karşımıza çıkmaktadır. Tekstil, aksesuar, ecza ve elektronikten logo ve marka isimlerine kadar her şey kopyalanmakta ya da taklit edilmektedir. Oyuncaklardan uçaklara, kişisel bakım ürünlerinden kullanıcı elektroniğine yapılan tüm sahtecilik faaliyetlerine genel bir yakınsama yapıldığında ise ortaya çıkan rakam 500 Milyar doları aşmaktadır.

Uluslararası Ticaret Odası [7] raporuna göre kesin olmamakla birlikte dünya ticaretinin %5-7'si yasadışı sahtecilik faaliyetleri üzerinden yapılmakta olup her geçen yıl payını %10 civarında da artırmaktadır. Bundan en çok etkilenen endüstrilerden biri olan elektronik endüstrisinde ise tüketici elektroniği başta olmak üzere, oyun konsollarından her türlü yongalara (IC) hatta güç üniteleri ve pillerde dahi sahtecilikle sıklıkla karşılaşılmaktadır. En güncel örnek olarak Intel'in yeni işlemcisi olan i7-920 paketinin sahtelerinin piyasada satışa sunulduğu anlaşılmış ve söz konusu olay üzerine Intel'den de bunu doğrular bir açıklama gelmiştir. Pazar araştırma şirketi IHS iSuppli'ye göre 2009'dan 2011'e kadar sahte elektronik ürünlerde dört kat artış yaşanmıştır [8]. Amerika Ticaret Dairesi'nin konu ile ilgili raporları da elektronik ürün sahtekarlığının son yıllardaki artış trendini göstermektedir (Şekil 1.1).

Özellikle Ar-Ge çalışmalarının yüksek yatırım maliyetleri gerektirdiği elektronik sektöründe yapılan sahte ve taklit ürünler, sektörü ve firmaları büyük oranda zarara uğratmakta, pazardan alacakları payda doğrudan kayba sebep olmaktadır. Herhangi bir Ar-Ge yatırımı yapmadan, çok daha kısa bir markete sunma zamanına sahip olan sahte ürün ve sahte üreticilerin ise Ar-Ge maliyeti, ürün oluşturma-planlama gibi çalışmaları olmamakla birlikte yapmadıkları tasarım ve çalışmalar üzerinden gelir elde etmektedirler. Bunun yanında sahtecilik marka değerinde kayba ve saygınlığında düşmeye sebep olur.



Şekil 1.1: Elektronik ürün sahteciliğinin son yıllardaki değişimi [1].

Günümüzde yeniden kullanılabilirlik, daha kısa bir markete sunma, esneklik ve daha birçok arzu edilir özelliğini sayabileceğimiz FPGA teknolojisi elektronik endüstrisinde oldukça fazla ilgi çekmiştir. Yine aynı özelliklerin bir yan tesiri olarak bu araçların üzerinden tasarımların çalınması kullanıcı firmalara büyük bir tehdit oluşturmaktadır [9].

1.1 Elektronik Sahtecilik Yöntemleri

Elektronik tasarımlara yönelik başlıca tehditler aşağıdaki gibi sıralanabilir [10]:

1.1.1 Tersine mühendislik

Üçüncü kişiler tarafından elektronik bir sistem, çalışması analiz edilerek tasarıma ait özellikler çıkarılabilir ve sistemin tüm veya bazı özellikleri taklit edilebilir, elektriksel özelliklerin ölçümü ile yonganın serimi hakkında bilgi edinilebilir. Böylece üçüncü kişiler ürünün tasarım stratejilerinde değişiklik yaparak, tasarımın özelliklerini kendi tasarımlarında kullanarak orijinal üretici firmaları zarara uğratabilirler.

1.1.2 Kopyalama

Üçüncü şahısların elektronik bir sistemi analiz yapmaya gerek duymadan olduğu gibi kopyalamasıyla, bellenim (firmware), yazılımların kopyalanması ile orijinal üretici firmalar zarara uğratılabilmektedir. Kopya üretim için kullanılan malzemelerin kalitesi

de çoğunlukla düşük olacağından bu tür bir sahtekarlık firmaların imajına da zarar vermektedir. Orijinal ürünlerin kalitesiz kopyalarına havacılık ve askeri elektronik gibi yüksek dayanıklılık ve güvenlik gerektiren sektörlerde karşılaşılması elektronik ürün sahtekarlığının ciddiyetini artırmaktadır.

1.1.3 İstek fazlası üretim

Yonga tasarımı sektöründe sık rastlanan üretimi başka firmalara yaptırma gerekliliği nedeniyle yonga tasarım firmaları tasarımlarını üretici firmalarla paylaşmak zorunda kalmaktadır. Sipariş fazlası üretimler yoluyla orijinal tasarımın aynısı olan ürünler üretilse de tasarım firması zarara uğratılabilmektedir ve bu tür sahtekarlıkların saptanması daha zordur.

1.1.4 Kurcalama

Elektronik sistemlere izinsiz erişim yapılarak sistemin çalışmasına dair bilgi edinme çabaları olan bu işlem genel bir tersine mühendislik çabasının bir parçası da olabilir.

1.2 Önlemler

FPGA'lar yeniden programlanabilme özellikleri nedeniyle elektronik tasarım aşırılması sorunundan en çok etkilenen sistemlerden biridir. IP tasarımların çalınması ve yongaların kopyalanması sorunun ulaştığı boyutlar FPGA üretici firmaları da güvenlik konusunda çözümler üreten FPGA'lar üretmeye itmiştir. Xilinx firması da FPGA'larında güvenlik amacıyla çeşitli çözümler üretmektedir, bunlardan bazıları, bit katarı şifreleme, aygıt DNA'sı ve kalıcı programlanabilen FPGA'lardır.

1.2.1 Bit katarı şifreleme

Gizlenmesi gereken veriler için akla gelebilecek ilk yöntemlerden birisi şifrelemedir. Güvenlik sorunlarını dikkate alan FPGA üretici firmalar standartlaşmış şifreleme ve kimlik doğrulama algoritmalarını kullanarak şifrelenen bit katarını FPGA üzerindeki şifre çözme donanımı sayesinde FPGA'ya yükleme özelliğini yeni FPGA ailelerinde standart bir özellik haline getirmişlerdir. Bu yöntemin dezavantajı ise şifrelenmiş bit

katarını çözmek için gereken anahtar verisinin FPGA üzerinde kalıcı olarak saklanması amacıyla sürekli bir güç kaynağına ihtiyaç duymasıdır.

1.2.2 Aygıt DNA'sı

İlk olarak Spartan 3A FPGA'sına eklenen bu özellik, FPGA üzerinde kalıcı olarak bulunan ve kullanıcı tarafından da okunabilen fakat değiştirilemeyen 57 bitlik her FPGA yongasına özgü kimlik verisidir. Bu veri kullanıcı tarafından belirlenecek bir güvenlik algoritması kullanılarak sağlama (checksum) verisi olarak kullanılabilir. Aygıt DNA'sı yonga kopyalamayı önlemek için kullanılabilmesine rağmen IP tasarımların çalınması sorununa tek başına çözüm olamamaktadır.

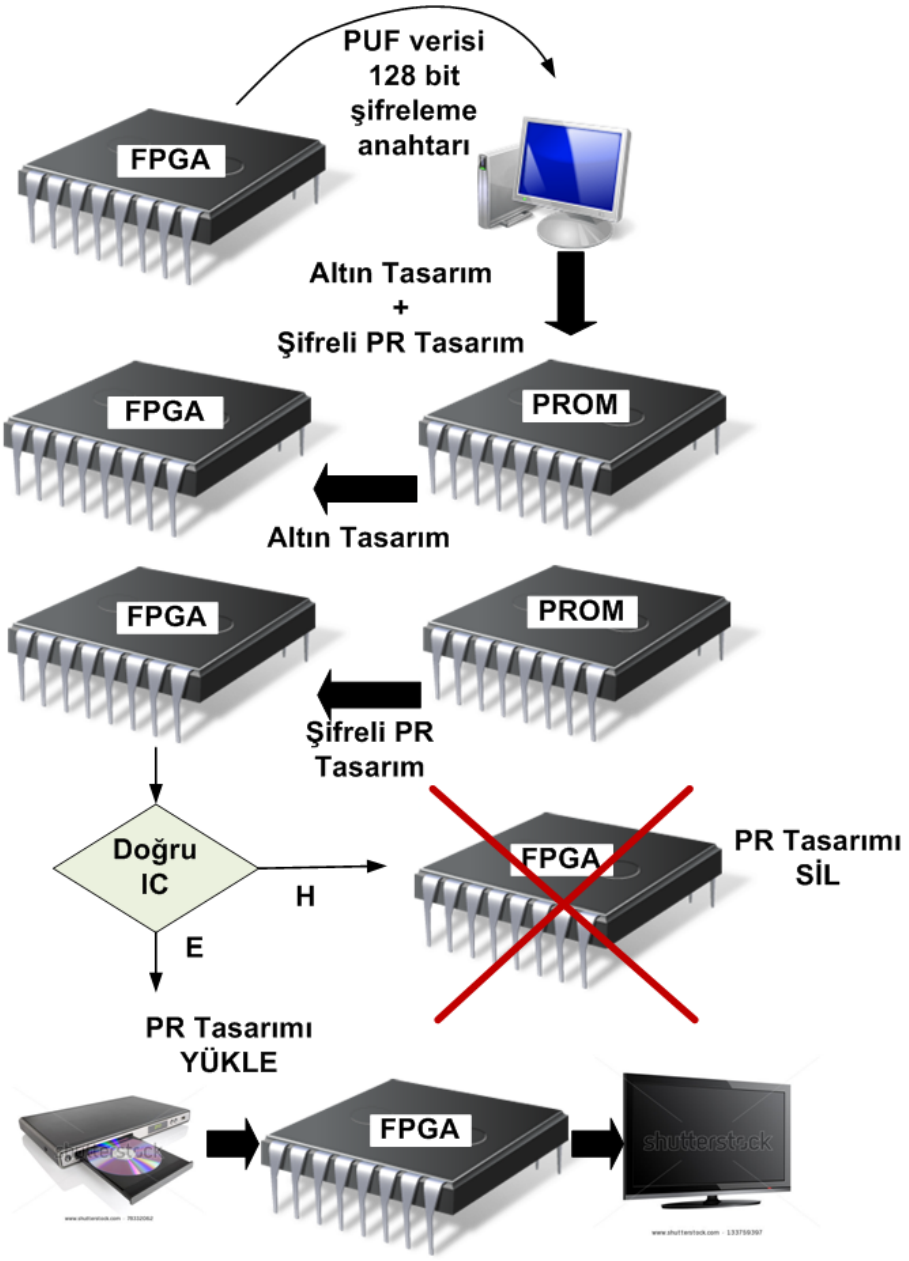
1.2.3 Kalıcı programlanabilen FPGA

Xilinx'in Spartan 3AN FPGA'larında, Lattice firmasının XP ve XP2 serisi FPGA'larında ve MachXO ve MachXO2 PLD'lerinde bulunan bu özellik kullanıcılara FPGA içinde uçucu olmayan bir depolama alanı (flash vb.) sunmaktadır. Böylece FPGA bit katarı verisi, FPGA içinde depolanarak bit katarının dışarısı ile bağlantısı kesilmiş olur [11], [12]. Bu tezin 2.2 bölümünde FPGA'larla ilgili detaylı bilgi verilmektedir.

1.3 Sistem Mimarisi

Burada anlatılacak sistemin tüm çalışmaları Xilinx firmasının ML505 kartı üzerinde yapılmıştır. ML505 kartı üzerinde kısmi yeniden yapılandırma (PR) özelliği olan bir Virtex 5 FPGA'sı bulunmaktadır. Bu tezin 2.3 bölümünde PR özelliği detaylı olarak anlatılmaktadır. Kart üzerinde ayrıca yapılandırma için gereken tasarımları kalıcı olarak saklamak amacıyla kullanılmış olan, birkaç bit katarı (bitstream) dosyası saklayabilecek 32 MB kapasiteye sahip bir BPI PROM bulunmaktadır. Tüm sistemin blok şeması Şekil 1.3'de verilmiştir ve sistemin çalışması Şekil 1.2'de gösterilmektedir.

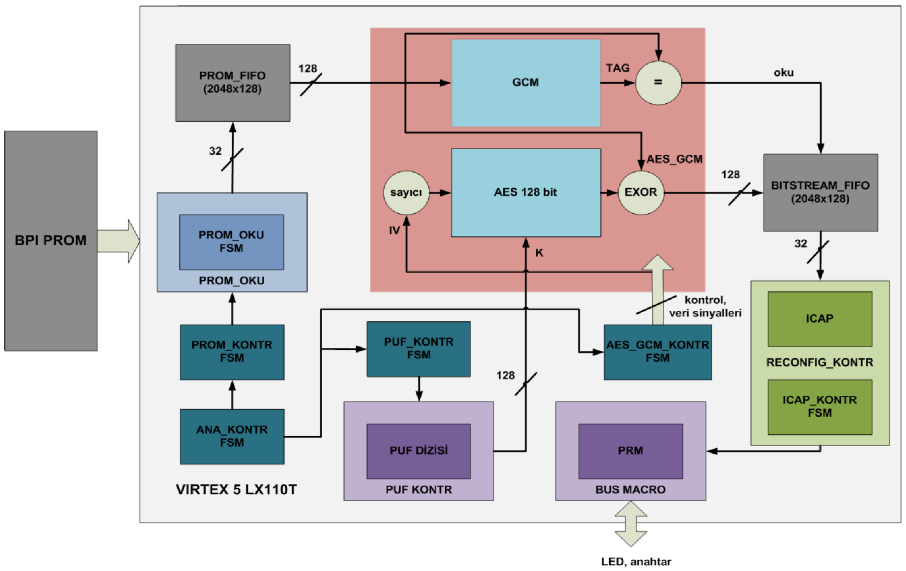
Sistemin çalışmaya başlamasıyla PROM üzerinde bulunan tam FPGA bit katarı yüklenir. Bu bit katarı; şifre çözme, anahtar üretimi, PROM okuyucu ve PR kontrolör devrelerini içerir. Bu ilk tasarım PROM üzerinde bulunan şifrelenmiş kısmi bit katarını okuyup, üretilen anahtar verisi ile şifre çözme ve kimlik doğrulama işlemlerini gerçekleştirir. Kısmi bit katarı kimlik doğrulama testini geçerse FPGA üzerindeki daha



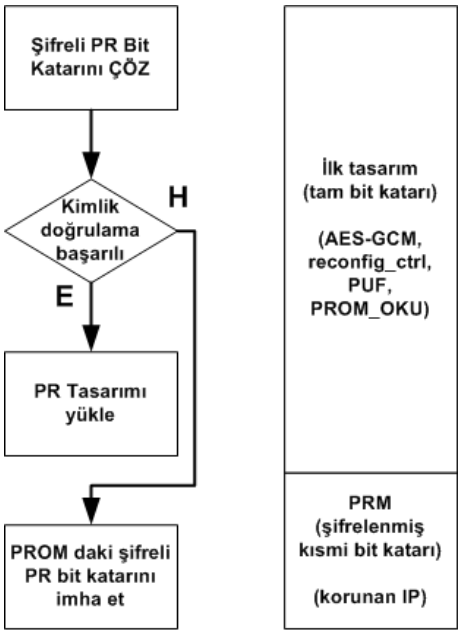
Şekil 1.2: Sistemin çalışması.

önceden belirlenmiş olan alana DPR (Dinamik Kısmi Yeniden Yapılandırma) işlemiyle yüklenir. DPR kullanımından 2.3.2 bölümünde ayrıntılı olarak bahsedilecektir. Kimliğin doğrulanamaması durumunda, PROM üzerindeki IP'nin içinde bulunduğu kısmi bit katarı imha edilerek korunmak istenen IP'nin çalınması önlenmiş olur (Şekil 1.4).

Yeniden yapılandırma kullanarak düşünce iyeliği (IP) güvenliği sağlayan bu çalışmada FPGA tasarımının güvenliği uluslararası standart bir şifreleme algoritması olan



Şekil 1.3: Tüm sistemin blok diyagramı.



Şekil 1.4: Sistemin çalışma akış diyagramı.

AES şifreleme algoritması ile sağlanmaktadır. AES ve gerçeklemesi detaylı olarak tezin 3. bölümünde açıklanacaktır. Kimlik denetleme algoritması olarak da kullanılan GCM çalışma kipi ile kullanılan AES-GCM sayesinde hem IP'nin kimlik doğrulaması yapılmaktadır hem de IP'nin bit katarının sınama toplamı (checksum) değerlendirilmektedir. FPGA bit katarının sadece bir bitinin bile farklı olması yeniden yapılandırma işlemine ve FPGA'ya zarar verebileceği için sınama toplamı kontrolü sistemin düzgün çalışması için gereklidir. AES-GCM algoritması paralel çalışmaya uygun bir algoritma olduğu için donanım uygulamaları için idealdir. AES-GCM

algoritmasının anahtar verisi fiziksel klonlanamaz fonksiyon (PUF) devresinden sağlanmaktadır. 4. bölümde PUF ve bu tezde kullanılan PUF gerçeklemesi anlatılacaktır. Böylece anahtar gizliliğini tehlikeye atabilecek bir iletişim yükünden kurtulunduğu gibi anahtar, kullanılan yongaya özel olarak elde edilmiş olmaktadır. Dolayısıyla her yonga için farklı bir anahtar olacağından IP'nin sadece doğru anahtarla şifrelendiği FPGA yongasında çalışması da sağlanmış olacaktır.

Kısmi bit katarını şifrelemek amacıyla gereken anahtar yani PUF verisi tek bir sefer FPGA üzerinde elde edilip dışarı alınmalıdır. Tek bir sefer yapılması gereken bu işlemin dışında PUF verisinin FPGA dışına alınmasına gerek kalmamakta böylece şifreleme algoritmasının anahtarının gizlenmesi sorununun önüne geçilmektedir.

2. FPGA VE YENİDEN YAPILANDIRMA

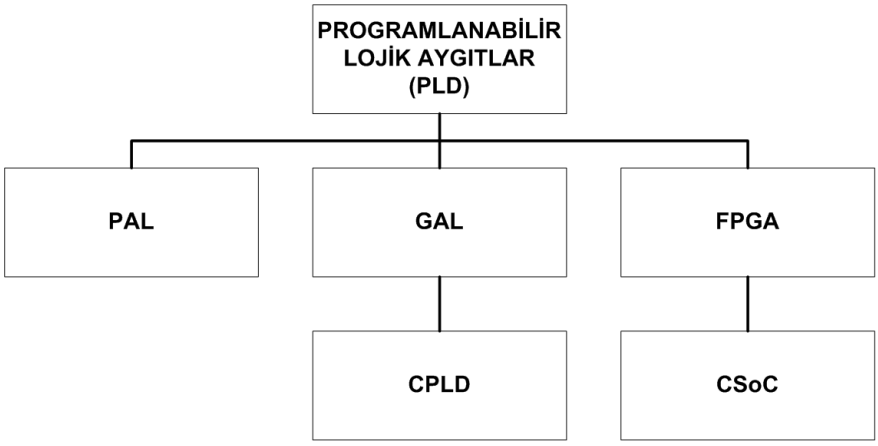
2.1 Genel Bilgi

Yeniden yapılandırma fikri ilk olarak Gerald Estrin' in 1960 tarihli "Organization of Computer Systems—The Fixed Plus Variable Structure Computer" adlı makalesinde ortaya atılmaktadır. Makalede ortaya atılan bilgisayar mimarisinde, genel amaçlı bir işlemci ve belli görevler için özelleşmiş ve optimize edilmiş olan programlanabilir donanım parçaları vardır. Genel amaçlı işlemci, bu donanım parçalarını gereğine göre devreye alır ve görevlerini tamamladıklarında diğer donanım parçalarını yükleyebilir. Temel motivasyon yazılımın esnekliği ile ASIC tasarımların hızını ve işlem gücünü birleştirmektir [2]. Estrin'in önerdiği yöntem zamanın elektronik devre teknolojisi ile gerçeklenemezdi ancak 1980 ve 1990'larda IC teknolojisinin karmaşık devreleri tek bir yongada bir araya getirebilmesiyle yeniden yapılandırılabilme konusunda birçok proje ortaya çıkmaya başladı. 1991'de üretilen ilk yeniden yapılandırılabilir bilgisayar olan Algotronix CHS2X4 ticari bir başarı olmasa da Algotronix firmasının teknoloji ve çalışanlarının Xilinx ile birleşmesiyle bu alanda bir atılım başlamıştır [13].

2.1.1 Programlanabilir mantık aygıtları (PLD)

İlk olarak 1970'lerin ortasında IBM, AMD ve Monolithic Memories, Inc (MMI) tarafından üretilen PLD'ler, mantıksal yapılarına göre PAL (Programlanabilir Dizi Mantığı) veya PLA (Programlanabilir Mantık Dizisi) olarak adlandırılan aygıtlardı. İlk PLD'ler senkron yapılar içermeyen kombinezonsal sayısal devrelerden oluşuyorlardı, fakat çıkışlarına tutucu eklenen tasarımlar da kısa bir süre sonra üretilmeye başlandı. PLA programlanabilir VE ve VEYA kapıları içerirken, PAL programlanabilir VE kapıları ve sabit VEYA kapıları içerir. PLD sektöründeki önemli bir gelişme GAL (Genel Dizi Mantığı) aygıtlarının üretilmesi olmuştur. Bu aygıtlar PAL'lardan farklı olarak silinip tekrar tekrar programlanabilmeleri özellikleri ile sektörde önemli bir gelişme sağlamışlardır. 1980'lerde yonga üretim teknolojisindeki gelişmelerle birçok

GAL ve PAL aygıtını içinde barındırabilen ve yüz binlerce sayısal mantık kapısının yerini tutabilecek CPLD'lerin üretimi mümkün olmuştur. FPGA'lar da 80'lerin ortasında ortaya çıkmaktadır. Günümüzde ise milyonlarca sayısal kapı devresini ve işlemci çekirdeklerini tek bir yongada üretmek mümkündür, bu tip hibrit yapılara CSoC denilmektedir. PLD aygıtlarının gelişimi ve sınıflandırılmaları Şekil 2.1'de görülmektedir [2].



Şekil 2.1: PLD'lerin sınıflandırılması [2].

2.2 Alanda Programlanabilir Kapı Dizileri (FPGA)

FPGA, programlanabilir ara bağlantılarla birbirine bağlı olan yapılandırılabilir mantık bloğu (CLB) dizilerinden oluşan yarıiletken yongalardır. ASIC devrelerin aksine üretildikten sonra üzerindeki devre değiştirilebilir. Bu özelliği sayesinde kullanıcıya sağladığı esneklik nedeniyle son yıllarda oldukça gelişmiş ve elektronik endüstrisinin birçok alanında uygulama alanı bulmuştur. Başlıca uygulama alanları şunlardır; uzay ve savunma elektroniği, ses işleme, otomotiv elektroniği, ASIC prototiplendirme, tüketici elektroniği, veri merkezleri, video ve görüntü işleme, güvenlik, yüksek başarımlı hesaplama, tıp elektroniği, haberleşme elektroniği. Genel olarak ASIC rakiplerine göre güç tüketimi, kaplanan alan ve hız açısından geride kaldığı bilinen FPGA'lar, son yıllardaki gelişmelerle bahsedilen kıstaslarda ASIC rakipleri ile rekabet edebilecek seviyelere gelmiştir [14].

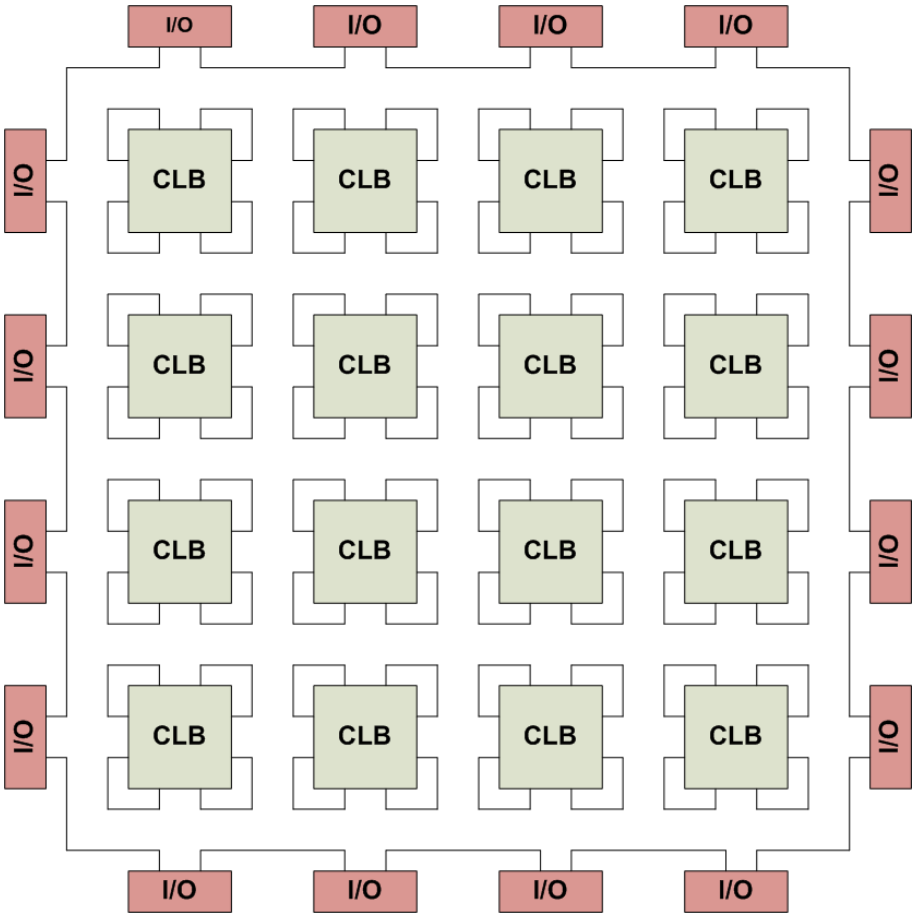
2.2.1 Üretim teknolojileri

FPGA endüstrisinde önemli üç programlama teknolojisi; SRAM hücre tabanlı, flash tabanlı ve anti-fuse tabanlı programlama teknolojileridir. SRAM hücre tabanlı programlama günümüzdeki hakim programlama teknolojisidir. Bunun sebebi, SRAM hücre tabanlı FPGA'ların elektriksel olarak kolayca programlanıp silinmelerinden dolayı birçok kez yeniden programlanabilir olmaları ayrıca standart CMOS teknolojileriyle üretilmeleridir. Standart CMOS teknolojileri ile üretime uygun olma, yüksek bütünleştirme ve hızlı gelişen üretim süreçleri ve düşük güç tüketimi ve hız açısından diğer teknolojilere üstünlük sağladığından SRAM hücre tabanlı FPGA'ların piyasada baskın FPGA'lar olmasını sağlamıştır. Bunun yanında SRAM hücre yapısı altı transistor gerektirerek diğer teknolojilere göre daha fazla alan kaplamaktadır ayrıca uçucu hafıza özelliğinde olması nedeniyle SRAM hücre tabanlı FPGA'lar için ayrıca bir bit katarı depolama aygıtı gerekmektedir (uçucu olmayan bellekler, flash, PROM, vs.) . Flash tabanlı teknoloji ise SRAM hücre yapısına göre daha az alan kaplamaktadır ve kalıcı programlama sağlamaktadır. Bunun yanında flash tabanlı yapılar SRAM hücre tabanlı yapılar gibi sonsuza dek programlanamazlar tekrar programlanma sayıları sınırlıdır ve standart CMOS teknolojisi ile üretilemezler. Anti-fuse tabanlı teknoloji ise az alan kaplar ve diğer iki teknolojiye göre daha düşük parazitik kapasite ve direnç büyüklüklerine sahiptir. Bu teknolojinin önemli dezavantajı ise standart CMOS teknolojisine uygun olmaması ve en önemlisi tekrar programlanamamasıdır [3].

2.2.2 Mimari

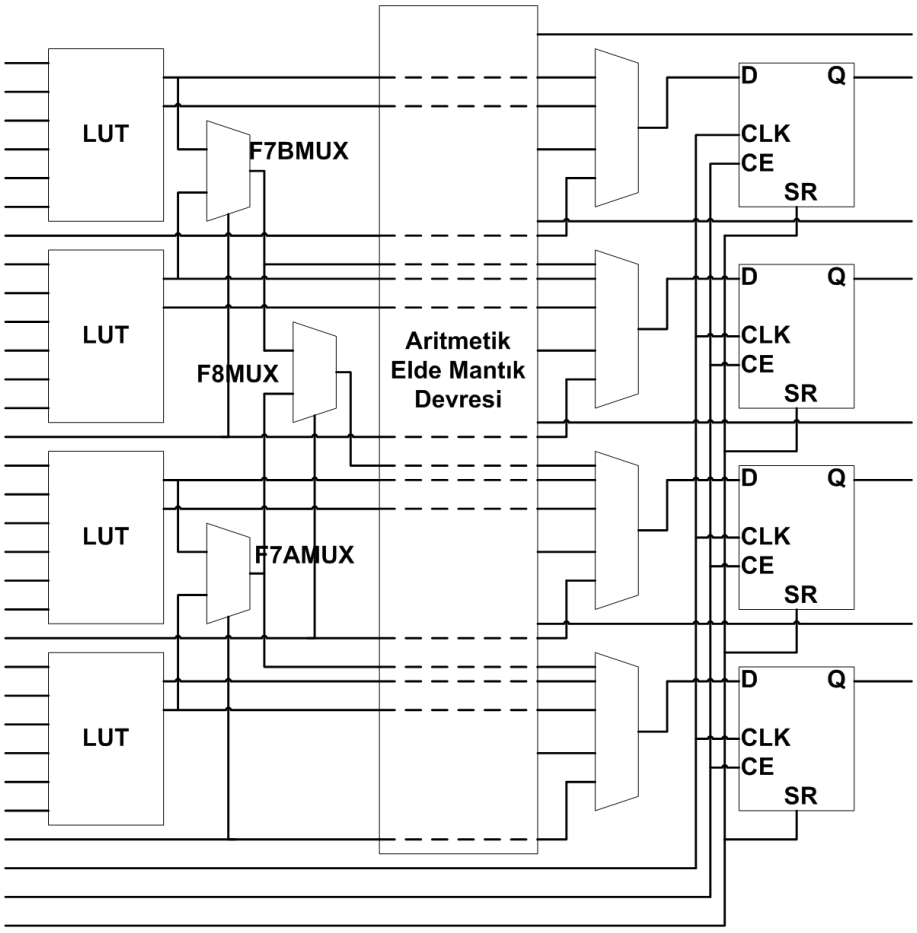
FPGA mimarileri üreticiden üreticiye hatta aynı üreticinin FPGA'ları arasında bile değişebilmektedir. Ancak genel olarak bir FPGA'nın CLB'ler, I/O bloklar ve bunları birbirine bağlayan yönlendirme kanallarından oluştuğu söylenebilir (Şekil 2.2). Bir CLB 2 girişli bir LUT kadar basit olabileceği gibi bir ALU kadar karmaşık yapıda da olabilir. Genel olarak bir CLB'de kombinezonsal fonksiyonları gerçekleyen LUT, aritmetik mantık devreleri, EXOR, çoğullayıcı devreler ve hafıza elemanları olan tutucu devreler bulunur. Bu çalışmada kullanılan Virtex 5 FPGA'sında bir CLB 2 dilimden (slice) oluşur. Bir dilim 4 adet 6-girişli LUT, 4 adet D tipi tutucu ve EXOR

elde mantık devreleri ve çoklayıcı devrelerinden oluşmaktadır (Şekil 2.3). Yonga üretim teknolojisindeki hızlı gelişmelere bağlı olarak günümüzdeki FPGA'larda daha üst seviye gömülü sistem aygıtları da tek bir yonga üzerinde gerçekleştirilebilmektedir. Tek bir FPGA yongasında blok RAM'ler (BRAM), çarpıcılar, bellek yönetim birimleri ve işlemciler bulunabilmektedir. FPGA'ların programlama teknolojileri de performansları üzerinde oldukça etkilidir.



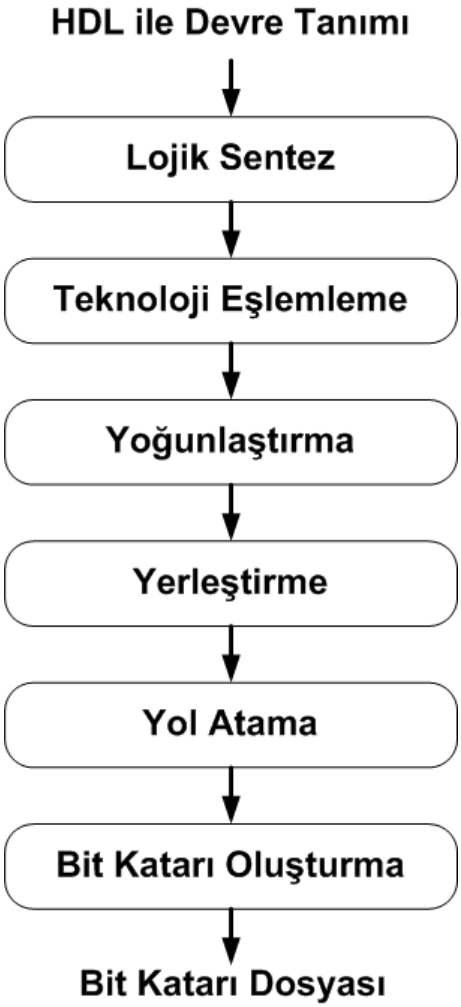
Şekil 2.2: FPGA mimarisi [3].

FPGA tasarımları için sıklıkla kullanılan iki HDL Verilog ve VHDL'dir. Tasarlanmak istenen devre bu dillerle tanımlandıktan sonra çeşitli sentez ve sentez sonrası işlemler (teknoloji eşlemleme, yoğunlaştırma, yol atama, vs.) için hazırlanmış olan EDA yazılımları ile bit katarı dosyası oluşturulur. Bu işlemlerin ilki olan sentez aşamasında, HDL ile tanımlanan uygulamanın teknolojiden bağımsız olarak mantıksal kapılar ve tutucular cinsinden ifadesi çıkarılır. Teknoloji eşlemleme, mantıksal kapılar ve tutuculardan oluşan devreyi FPGA mimarisindeki k-LUT (k girişli LUT) ve FF'lar cinsinden ifade edilen bir devreye dönüştürür. Bu işlem sırasında alan hız ve güç tüketimi ile ilgili kısıtlar da göz önünde tutulur. Yoğunlaştırma aşamasında k-LUT



Şekil 2.3: Virtex 5 dilim yapısı [4].

ve FF'lar bir araya toplanarak FPGA içindeki CLB benzeri yapılar oluşturulur. Yerleştirme işlemi ise yoğunlaştırma sonunda ortaya çıkan devredeki elemanların FPGA içindeki hangi elemanlara atanacağını belirler. Yol atama aşamasında, FPGA'daki sınırlı sayıdaki yerleştirme kanalları sadece bir bağlantıya atanacak şekilde devrenin bütün bağlantıları yapılır. Bu işlemler sırasında devrenin zamanlama analizi de yapılır. Zamanlama analizinin amacı hem devrenin maksimum hızını belirlemek hem de yol atama sırasında devrenin belirli hız kısıtlarına uymasını sağlamaktır. Tüm bu işlemlerden sonra hedef FPGA'ya göre gerçeklenen devre FPGA'daki SRAM hücrelerini programlayacak mantıksal '1' ve '0' dizileri şeklinde yazılarak bit katarı dosyası oluşturulur. FPGA bit katarının oluşturulması için gereken yazılım akış diyagramı Şekil 2.4'de görülmektedir.



Şekil 2.4: FPGA tasarımı yazılım akış diyagramı [3].

2.3 Kısmi Yeniden Yapılandırma (PR)

Tasarımların, çalışma zamanları kesişmeyen dolayısı ile aynı anda ihtiyaç duyulmayan devre parçaları donanım kaynaklarını zaman paylaşımlı olarak kullanabilirler. Böylece devrenin harcadığı alanda ve güç tüketiminde önemli ölçüde azalma sağlanabilir. Günümüzde gittikçe karmaşıklaşan ve daha çok işlem gücü gerektiren elektronik sistemlerde gerçeklenmek istenen algoritmalar zaman uzayında birbirinden bağımsız görevlere bölünebilir böylece ardışıl olarak sırayla yürütülen bu alt görevlerle tüm bir algoritma gerçeklenmiş olur. Birbirinden bağımsız olan alt görevler tek bir donanım kaynağında koşturularak (FPGA'da PRR) gömülü bir sistemin karmaşıklığı azaltıla bilinir. [15]'de verilen gerçek zamanlı biyometrik tanıma uygulamasının Virtex 4 FPGA'sında PR özelliği ile gerçeklemesi buna örnek olarak

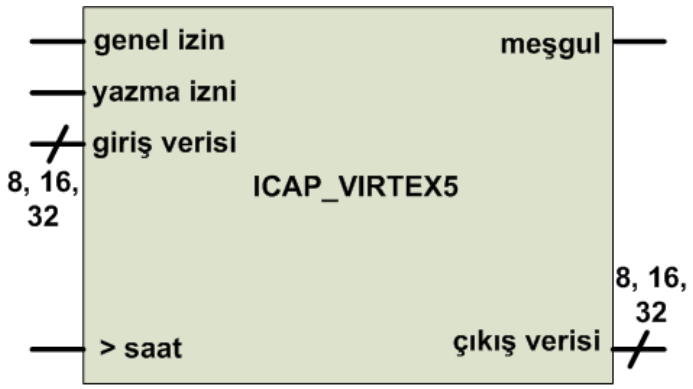
verilebilir. Ayrıca FPGA'nın yapılandırılmasının otomatik olarak yapılabileceği uygulamalar gerçekleştirilebilir. Yazılım güncellemesi yapılır gibi bir yongaya donanım güncellemesi yapılabilir. Kısmi bit dosyasının FPGA üzerine yüklenmesi tüm FPGA'nın silinip tekrar yapılandırılmasından daha kısa sürer ve depolanması gereken bit dosyasının boyutu daha küçük olur. Tasarıma sağladığı tüm bu katkılar nedeniyle PR özelliği gittikçe tüm FPGA'larda aranan bir özellik olmakta ve birçok FPGA tasarımında uygulama alanı bulmaktadır.

Xilinx'in Virtex serisi FPGA'larının genelinde ve yeni 7 serisi FPGA'larda bulunan kısmi yeniden yapılandırma (partial reconfiguration, PR) özelliği; FPGA'nın sadece belli bir bölgesindeki lojik elemanları, FPGA'nın diğer kısımlarının yapılandırmasını değiştirmeden yapılandırmaktır. İkiye ayrılır; Dinamik kısmi yeniden yapılandırma (DPR) ve statik kısmi yeniden yapılandırma (SPR). Dinamik olanında kısmi programlanabilen bölge yeniden programlanırken FPGA'nın kalan kısmı (statik tasarım) çalışmaya devam eder. Statik olanında ise kısmi bölgenin programlanması sırasında statik tasarım geçici olarak işlevsiz kalır [16].

Xilinx FPGA'larda tasarımlar arasında yapılmak istenen değişimin büyüklüğüne göre iki çeşit PR tasarım yaklaşımı vardır. Bunlar fark temelli ve bölüntü temelli PR tasarım yaklaşımlarıdır. Fark temelli PR tasarımlarda LUT fonksiyonları veya I/O yapılandırmaları gibi küçük ölçekli değişiklikler yapılır. Bu yaklaşımda ikinci bit katarı sadece ikinci tasarımın ilk tasarımdan farkını içerir. Bölüntü temelli PR, FPGA üzerinde belirli bir alanın (örneğin: dikdörtgen bir slice dizisi alanı) yapılandırmasının tamamen değişmesidir [17]. Fark temelli PR orta ölçekli FPGA'larda da (Spartan 3E, Spartan 6) bulunurken, bölüntü temelli PR sadece üst sınıf FPGA'larda (Virtex ve 7 serisi FPGA'lar) bulunan bir özelliktir. Bu çalışmada bölüntü temelli DPR çalışmasının Xilinx Virtex 5 LX110T FPGA'sında gerçeklemesi anlatılacaktır.

Xilinx FPGA'lara bit katarı yüklemek için çeşitli yöntemler mevcuttur, bunlar; seri yapılandırma, JTAG, SelectMap, SPI flash, BPI flash bellekler aracılığıyla yapılabilir. Seri olarak bit katarı yüklemek için FPGA üzerinde bulunan pinlerden veya yine seri olarak JTAG kapısı ile PC'den direk FPGA'ya bit katarı dosyası atılabilir. SelectMap ara yüzü kullanılarak paralel olarak çift yönlü 8, 16 veya 32 bit veri kapısından bit katarı yüklenebilir veya FPGA üzerindeki tasarım geri okunabilir (Read back özelliği). Bu çalışmada kullanılan FPGA programlama arayüzü ise ICAP'dir. ICAP, SelectMap

ile aynı şekilde bit katarı yükler ancak SelectMap'den farklı olarak ICAP FPGA'nın içinde bulunur ve FPGA üzerinde bulunan tasarıma (kullanıcı tarafından tasarlanıp FPGA'ya yüklenmiş tasarım) FPGA'nın yapılandırma devresine erişim imkanı sağlar. Böylece bu çalışmada amaçlanan kimlik doğrulama sonrası kısmi bit katarı dosyasının FPGA'ya yüklenmesi işlemi FPGA içindeki devre tarafından gerçekleştirilir. ICAP; saat, saat izin, yazma izin ve veri girişleri ve veri ve meşgul çıkış işaretleri olan bir devre olarak sürülebilmektedir. Giriş ve çıkış veri boyutları; 8, 16 veya 32 bit olarak ayarlanabilir. Saat ve izin sinyalleri uygun şekilde sürülerek giriş veri hattından bit katarı verilerek PR gerçekleştirilmektedir. Çıkış veri hattı başka uygulamalarda FPGA üzerindeki tasarımın geri okunması gibi farklı amaçlarla kullanılabileceği gibi gerçekleşen işlemin gidişatını belirten bir statüs sinyali olarak da işlev görmektedir. Virtex 5 FPGA'sı üzerinde bulunan ICAP biriminin giriş çıkış sinyalleri Şekil 2.5'de görülmektedir [18].



Şekil 2.5: ICAP virtex 5.

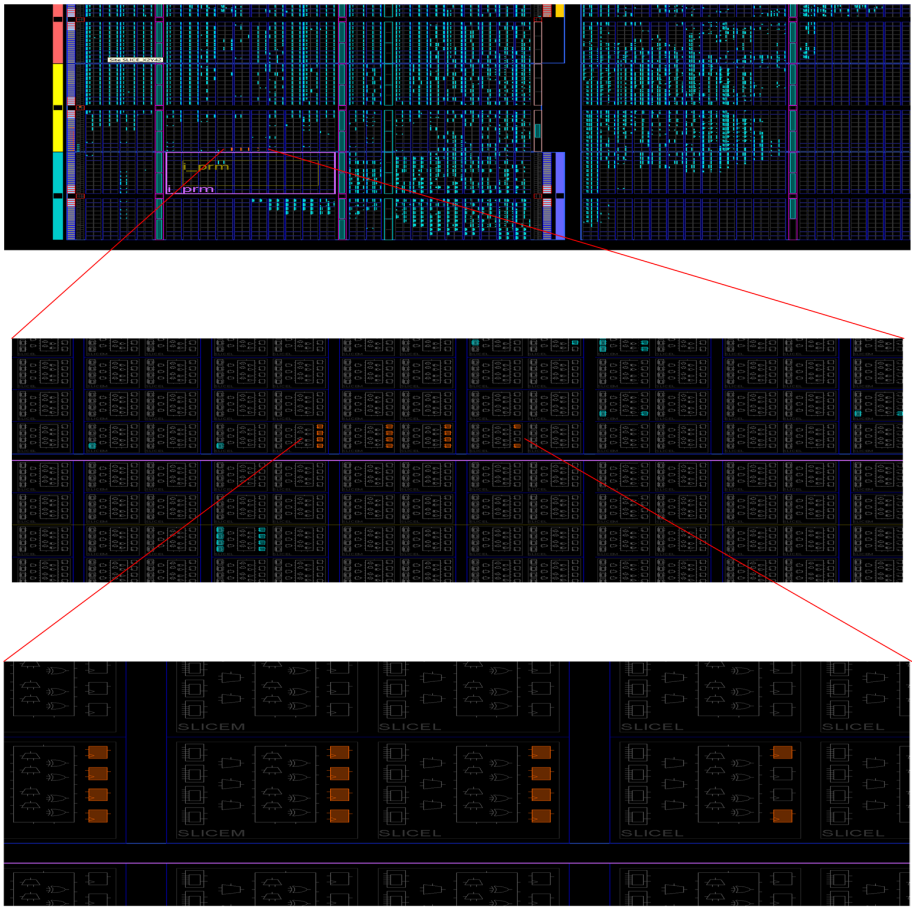
2.3.1 Gerekenler

Genel olarak Xilinx FPGA donanım tasarımı için yeterli olan Xilinx ISE IDE yazılımı PR tasarımlarda yetersiz kalır. HDL sentez işlemi ISE ile yapılabilir ancak eşlemleme (mapping) işleminden itibaren gereken gerçekleme (implementation) işlemleri PlanAhead yazılımında hazırlanabilir. PlanAhead'de PR tasarım yapmak için de ayrı bir lisans (PR lisansı) gerekmektedir.

2.3.2 DPR tasarım adımları

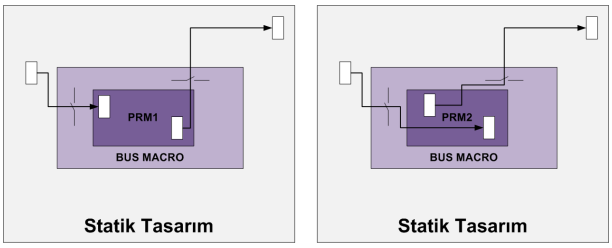
PR tasarımı genel olarak 3 kısma ayırabiliriz; statik modüller, kısmi yapılandırılabilen modüller (PRM) ve Bus Macro. Statik modüller, FPGA'daki tasarımın çalışma

ömrü boyunca çalışmaya devam edecek olan devredir, FPGA elektriği kesilinceye kadar veya genel olarak yeniden programlanıncaya kadar çalışmaya devam ederler. PRM, yeniden programlanabilen modüllerdir, tasarımın başında belirlenmiş olan alan içindeki bu modüller kendi alanlarına gelebilecek başka tasarımlarla FPGA çalışması sırasında yer değiştirebilirler. Bus macro, PR modüller ile statik modüller arasındaki iletişimi sağlayan devredir. Bus macro elemanlarının yeri FPGA içinde sabit olarak belirlenir. Sabitlemenin nedeni farklı PR tasarımların giriş çıkış yapılandırmalarının değişmesinin önlenmesidir. Böylece PR modüller değişse bile aynı hatlar üzerinden statik tasarım ile iletişim sağlanabilecektir (Şekil 2.7). Bus macro ayrıca PR işlemi esnasında PRR ile statik bölgenin bağlantısını keserek kullanılmalıdır, böylece PRM devrede PR sırasında giriş çıkış sinyallerinde oluşabilecek hataların (glitch, jitter, vs.) önüne geçilebilir. Bus macro devresi Şekil 2.6'da görüldüğü gibi izin girişli tutucular kullanılarak gerçeklenmiştir.



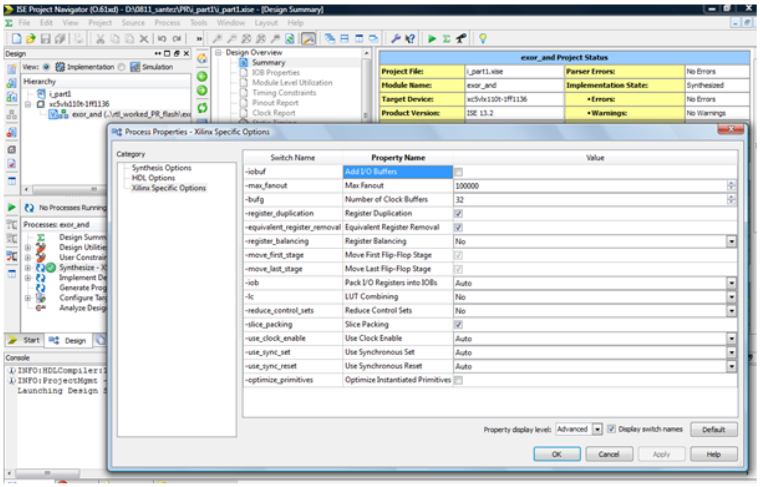
Şekil 2.6: Virtex 5 üzerinde bus macro ve PRM.

Her bir PR tasarım ayrı bir proje olarak ISE yazılımında sentezlenerek sentez dosyaları (*.ngc) oluşturulur. PR modüllerin portları direk olarak FPGA I/O'larına



Şekil 2.7: Bus macro üzerinden iletişim.

bağlanmayacağından sentez seçeneklerinden "Xilinx Specific Options" seçilerek buradaki "Add I/O Buffers" seçeneği kaldırılır (Şekil 2.8). Bu seçeneğin kaldırılmasıyla sentezlenecek tasarımın portları FPGA'nın I/O pinlerine bağlanmamış olur. PR modüller üst modülün içine yerleştirileceğinden giriş çıkışlarının FPGA I/O'larına bağlanmaması gerekir.



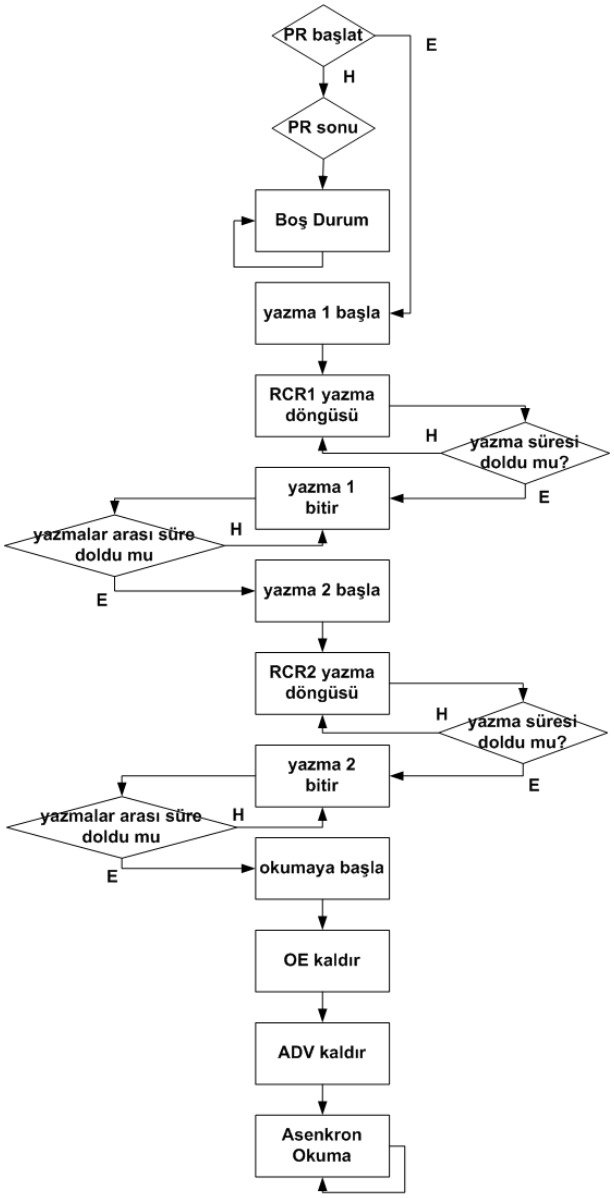
Şekil 2.8: Xilinx ISE sentez seçenekleri.

Tüm tasarım, PR modüllerin sentez dosyaları eklenmesi ile (Bus macro da sentez dosyası ile eklenebilir) ISE yazılımında sentezlenerek sentez dosyası oluşturulur. Bu aşamadan sonra yapılması gereken ve PlanAhead yazılımı ile gerçeklenen PR tasarım adımları Ek 1'de verilmiştir [19].

2.3.3 BPI PROM okuyucu devre

Kısmi bit katarı verisinin boyutu, PRR alanının büyüklüğü ile doğru orantılıdır. Tüm FPGA tasarımının bit katarı boyutu mega baytlar mertebesindedir. FPGA üzerindeki BRAM miktarı ise bu büyüklükte bir veriyi depolayamaz ve BRAM'lerin büyük kısmı gerçeklenen devrede kullanılıyor olabilir. Bu nedenle kısmi bit katarını FPGA dışındaki uçucu olmayan belleklerde saklamak gerekmektedir. Bu amaçla bit katarı dosyaları ML505 kartı üzerindeki 16 bit veri hattı bulunan BPI paralel PROM

belleğinde depolanmışlardır. BPI PROM kullanmanın dezavantajı FPGA içindeki devrenin çalışma hızını düşürmesidir. 33 MHz saat frekansı ile 16 bit veri gönderen belleğin işlem hacmi (518 Mbps), 100 MHz saat frekansı ile 32 bit (3200 Mbps) paralel veri işleyebilen ICAP modülünün işlem hacminin oldukça gerisinde kalmaktadır. Tasarlanan BPI PROM okuyucu devre sistemin başlangıcında PROM'u okuma kipine alacak ve diğer ayarlarını yapacak şekilde programlamaktadır ve PR işlemi başlayınca verilen adresten şifrelenmiş PR bit katarını okumaktadır. Okuyucu devrenin sonlu durum makinasının akış diyagramı Şekil 2.9'da görülmektedir.



Şekil 2.9: BPI PROM okuma durum makinası akış diyagramı.

3. ŞİFRELEME ALGORİTMASI

3.1 AES-GCM Algoritması

AES, 2001'de Birleşik Devletler Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından standartlaştırılmış bir şifreleme algoritmasıdır. Algoritma, Vincent Rijmen ve Joan Daemen tarafından geliştirilmiş olan Rijndael algoritmasında bazı değişiklikler yapılarak oluşturulmuştur. Rijndael algoritmasının 32 bitin katı olan anahtar ve veri uzunluklarına izin vermesine rağmen AES için veri uzunluğu 128 bit olarak sınırlandırılmıştır. Günümüzde birçok şifreleme paketinde kullanılan AES, Amerikan Ulusal Güvenlik Teşkilatı (NSA) tarafından çok gizli bilginin şifrelenmesi için önerilmektedir. [20]. Simetrik anahtar özelliğine sahip olması nedeniyle veriyi şifrelemek ve şifrelenmiş veriyi çözmek için kullanılan anahtar verisi aynıdır. Veri uzunluğu 128 bit olup, anahtar uzunluğu 128, 192 veya 256 bit olabilmektedir [21]. Tasarımımızda kullanılan AES algoritması 128 bit anahtar uzunluğu için çalışmaktadır.

Şifreleme algoritmaları blok olarak adlandırılan belirli uzunluktaki verileri şifreleyebilirler. Daha uzun blok dizilerini şifrelemek için şifreleme algoritması bir çalışma kipi ile kullanılır. Çalışma kipi, blokların ne şekilde şifreleneceğini belirler. Kullanılacak verinin yapısına ve gerçekleme ortamına göre (yazılım, donanım, vs.) çalışma kipleri güvenliği ve performansı oldukça etkileyebilirler. Bazı çalışma kipleri şunlardır:

3.1.1 Elektronik kod defteri (ECB)

ECB en basit çalışma kiplerindendir. Şifrelenecek veri bloklara ayrılarak ayrı ayrı şifrelenir. Bloklar arasında bir bağımlılık olmadan şifreleme yapılır. Şifrelenecek blokların rastgele seçilmesine veya bazı şifrelenmiş blokların kaybolmasının önemsenmediği uygulamalar için uygun bir yöntemdir. Ayrıca bloklar arası bağımsızlık nedeniyle paralelleştirmeye de uygundur. Bu yöntemde aynı blok için elde edilecek şifrelenmiş blok da hep aynı olacaktır. Bu özellikleri, veri içerisinde benzer örüntüler içeren uygulamalar için güvenlik zafiyeti yaratabilir. Bu durum elinde orijinal

metinden ve şifreli metinden benzer parçalar olan bir şifreli yazı analiz uzmanı için fazlaca bilgi verebilir. Şifrelenecek veri içerisinde tekrar eden yapılar olması güvenliği zayıflatacaktır. Bizim uygulamamızda da benzer bir durum söz konusudur. Bit katarı verisi uzun sıfırlar içerebilen bir metindir. ECB ile ilgili bir başka sorun da tekrarlama saldırılarına karşı zayıf olmasıdır. Şifrelenmiş metin kasıtlı olarak değiştirilerek alıcı taraf yanıltılabilir. Bu tür problemlere çözüm olarak zincirleme yöntemi geliştirilmiştir.

3.1.2 Şifreleme bloğu zincirleme (CBC)

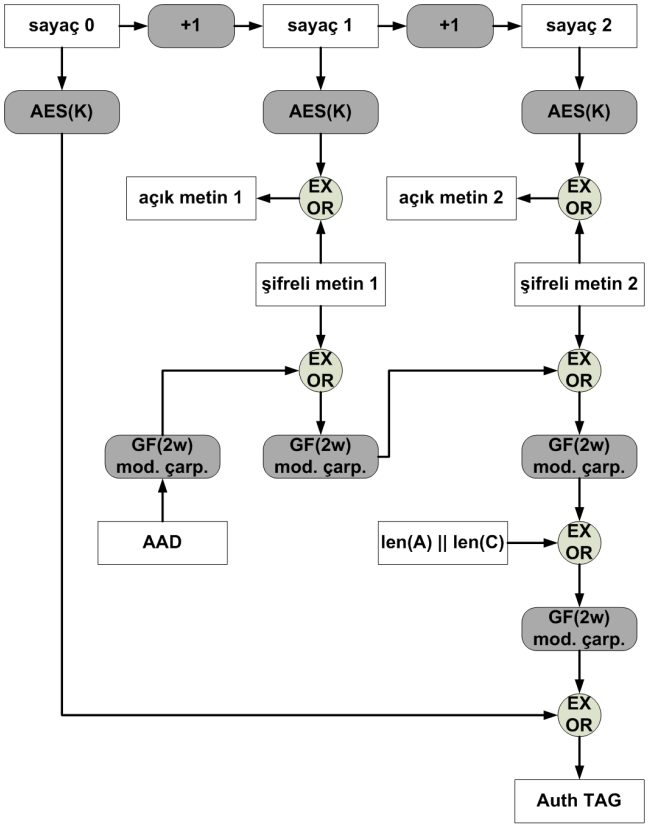
Zincirleme metodu ile şifrelenen bloklar arasında bir geri besleme sağlanmış olur. Her bloğun şifreli hali bir önceki bloğun şifreli haline bağlıdır. Benzer örüntüler içeren CBC kipinde şifrelenecek blok bir önceki şifrelenmiş blok verisi ile EXOR'lanarak şifrelenir. Böylece her blok kendinden önceki bloğa bağlanır ve tekrar eden blokların şifrelerinin aynı olması daha zorlaşır. Ancak şifreli blokların birinde olacak bir hata o bloğun yanlış çözülmesine neden olacağı gibi kendisinden sonraki blokları da etkileyecektir. Ayrıca her blok kendinden önceki bloğun şifrelenmesini beklemek zorunda olduğu için paralelleştirmeye uygun değildir. Paralelleştirme, bu çalışmada olduğu gibi donanım olarak gerçeklenecek çalışmalar için önemli bir özelliktir. İlk bloğun EXOR'lanacağı veri ilklendirme vektörü (IV) olarak adlandırılır ve rastgele bir değer atanması tekrarlama saldırılarına karşı dayanıklılık için gereklidir [22].

3.1.3 Galois sayaç kipi (GCM)

GCM, hem veri gizliliği hem de kimlik doğrulaması sağlayan bir algoritmadır. Başlangıç değeri IV olan sayaç verisi ile EXOR'lanan bloklar şifrelenerek şifreli metin oluşturulur. Böylece şifrelenecek metin içindeki tekrar eden yapılar için şifreli metin aynı olmayacaktır. Bloklar birbirine bağlı olmadığından GCM kipi paralelleştirmeye de uygundur. Bitstream verisi çok fazla sayıda tekrar eden bit dizileri içerebilen bir veri olduğundan şifreleme algoritmasının sayaç kipinde kullanmak güvenlik açısından gereklidir. GCM kimlik (authentication) algoritması olarak da kullanılmaktadır. Şifrelenmiş veriler ve isteğe bağlı olarak eklenebilecek kimlik doğrulama verisi (AAD) Galois Alanı üzerinde işlem yapan bir özet fonksiyonundan (hash) geçirilerek etiket verisi (tag) elde edilir. Etiket verisinin uzunluğu standartlara göre 128, 120, 112, 104

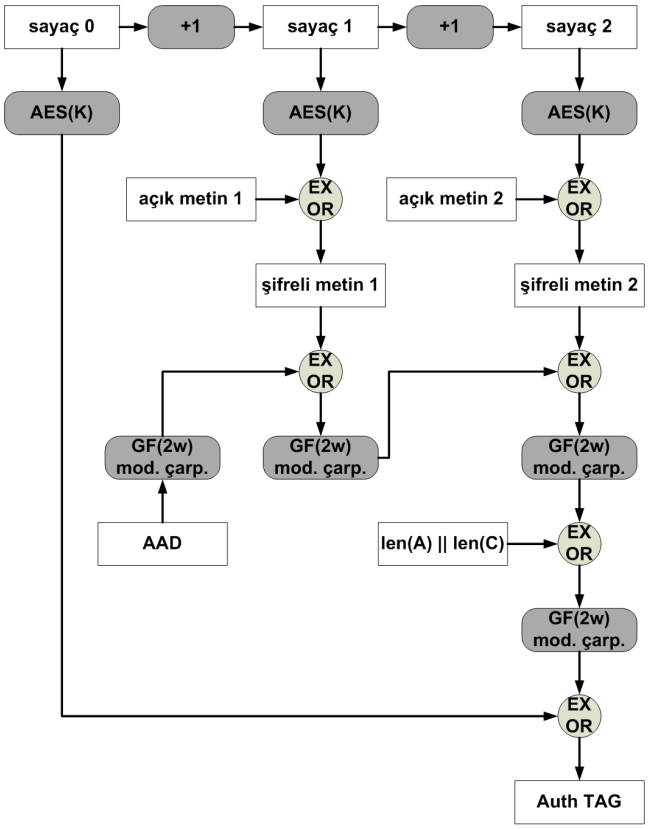
veya 96 bit uzunluklarında olabilir, bu çalışmada etiket verisi 128 bit uzunlukludur [23]. Böylece şifrelenen bütün bit katarı değerlerine bağlı bir özgün değer elde edilebilir ve çözülen bit katarı verisinin doğruluğundan emin olunabilir. Bit katarı verisinin 1 bitinin bile hatalı olması kısmi tasarımın FPGA üzerine yüklenememesine, hatta FPGA'nın zarar görmesine neden olabileceğinden şifrelenen veriden kimlik verisi çıkarmak uygulamanın kullanılabilirliği açısından gereklidir. Tüm bu nedenlerle uygulamamız açısından GCM, çalışma kipi olarak en uygun algoritmalardandır.

AES-GCM şifre çözme algoritmasının blok diyagramı Şekil 3.1'de görülmektedir. GCM kipinde şifrelenen veri her zaman sayaç değerleri olmaktadır. Metni şifrelerken şifrelenmiş sayaç değerleri açık metin ile EXOR'lanır. Şifreli metni çözerken ise şifrelenmiş sayaç değerleri şifreli metin ile EXOR'lanarak açık metin elde edilir. GCM kipinde şifreleme ve şifre çözme işlemlerinde kullanılan şifreleme algoritması aynıdır (AES şifreleme algoritması), değişen EXOR kapılarının giriş ve çıkışlarıdır. AES-GCM şifreleme algoritmasının blok diyagramı Şekil 3.2'de görülmektedir [24].



Şekil 3.1: AES-GCM şifre çözme algoritması blok diyagramı.

Kimlik doğrulama verisinin elde edildiği çarpma işlemleri Galois alanında GF(2^{128}) modüler çarpma işlemleridir. Şekil 3.1 ve Şekil 3.2'de görülen bu modüler çarpma



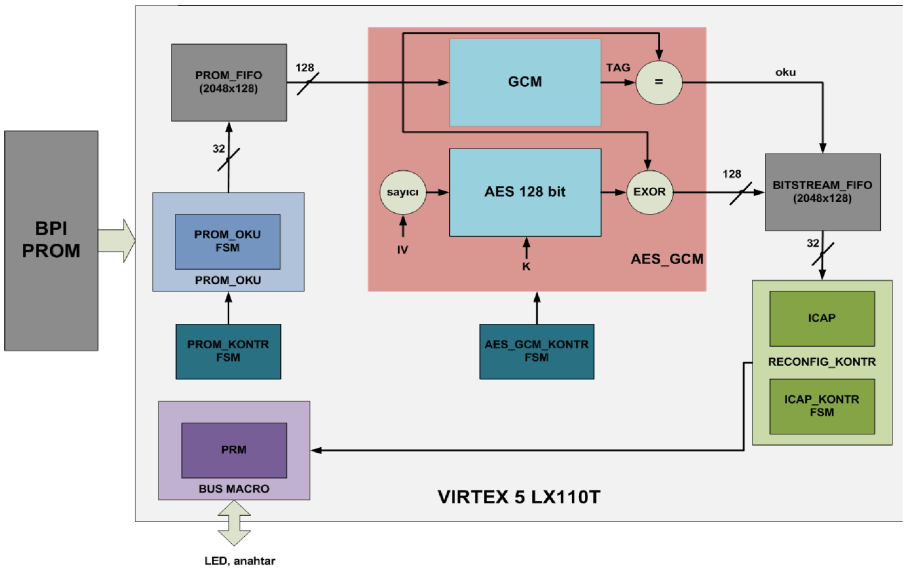
Şekil 3.2: AES-GCM şifreleme algoritması blok diyagramı.

blokları temel olarak anahtarın özet fonksiyon sonucu ile girişlerine gelen sayıları çarpıp 2^{128}'e göre kalanını almaktadır. Bekleneceği gibi gerçeklenmesi oldukça kaynak tüketecek olan bu işlem donanıma özelleştirilmiş olan algoritma sayesinde daha küçük parçaların çarpımı olarak gerçeklenerek kabul edilebilir kaynak tüketimi ile gerçeklenmektedir. Özet fonksiyonu, fonksiyon girişinin AES algoritması ile anahtar olarak 0 vektörü kullanılarak şifrelenmesidir. Anahtarın özet fonksiyonu da anahtar verisinin AES algoritmasına giriş olarak verildiği 128 bitlik 0 vektörünün anahtar girişine verildiği AES çıktısıdır. len(A/C), AAD verisinin (A) veya gizli metnin (C) bayt sayısının 64 bit olarak ifadesidir.

3.2 Donanım Gerçeklemesi

Bu çalışmada kullanılan donanım kodu kaynaklarda belirtilen Open Cores projesi üzerinde gerekli değişiklikler ve eklemeler yapılarak hazırlanmıştır [25]. AES-GCM devresi ve PR kontrolör devresi, PUF verisi yerine sabit anahtar kullanılarak kart üzerinde test edilmiştir. Bu çalışmanın blok diyagramı Şekil 3.3'de görülmektedir. Bit katarı verisi Xilinx'in Virtex 5 LX110T FPGA'sının bulunduğu ML505 kartı

üzerindeki BPI PROM üzerinde saklanmaktadır. PROM_OKU devresi PROM'dan şifrelenmiş kısmi bit katarını okuyup, AES_GCM modülüne göndermektedir. Burada bit katarı verisi deşifre edilir ve kimlik verisi hesaplanır. Hesaplanan kimlik verisi PROM üzerinden okunan, önceden hesaplanmış kimlik verisi ile aynı ise bu veri bloğu kimlik doğrulamasından geçmiştir ve RECONFIG_KONTR devresine gönderilerek ICAP üzerinden yüklenmektedir. Kimlik verisinin kontrolü için bütün verinin çözülmesinin beklenmesi gerektiğinden çözülen bit katarı verisini FPGA içinde bir yerde depolamak gerekir. Uygulamaya göre depolanması gereken veri çok büyük olabileceğinden kimlik verisi hesaplanan bit katarı verisinin büyüklüğü 2048 kelime ile (128 bit) sınırlandırılmıştır. Yani her 2048 kelimelik bloğun ayrı kimlik verisi vardır ve her 2048 blokluk veri kimlik doğrulama işleminden geçerse işlemlere devam edilmektedir. Son bit katarı veri bloğunun da çözülmesi ile kısmi yeniden yapılandırma işlemi tamamlanmış olur.

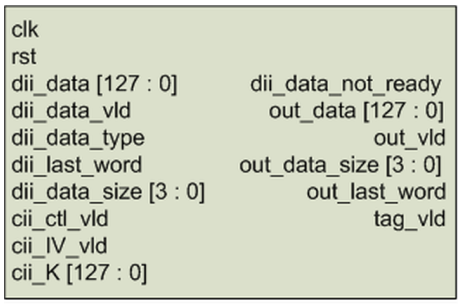


Şekil 3.3: AES-GCM ile PR blok diyagramı.

3.2.1 AES-GCM devresi

AES_GCM devresinin giriş, çıkışları Şekil 3.4'de görülmektedir. Başlangıç işareti "cii_ctl_vld" sinyalidir. Giriş verisi bit uzunluğu $2^{39} - 256$ değerine kadar ulaşabilir. 128 bit uzunluğundaki giriş "dii_data" ve giriş verisinin bayt olarak uzunluğu "dii_data_size" portuna girilebilir. "cii_K" anahtar girişidir ve bütün işlemler boyunca sağlanmalıdır. "dii_data_type" giriş verisinin şifrelenecek veri veya kimlik verisi olacağını belirtir, "dii_data_vld" giriş verisi örnekleme sinyali ile beraber

sağlanmalıdır. “cii_IV_vld” işareti, IV(Initialization Vector) örnekleme sinyalidir. IV, şifrelenen AES verisi olan sayacın başlangıç değeridir, 1 ile 2^{64} bit arasında herhangi bir uzunlukta olabilir fakat GCM algoritmasının etkin ve basit şekilde kullanılması açısından 96 bit uzunluğu önerilmektedir, [23]. IV her farklı anahtar değeri için farklı olmalıdır. Kimlik verisi (Additional Authentication Data, AAD) 2^{64} bit uzunluğuna kadar değişebilir, hiç kullanılmaya da bilinir. Kullanılan bütün bu veriler 8 bitin katları uzunluğunda olmalıdır. Son kelimeyi belirtmek için “dii_last_word” işareti kullanılır.



Şekil 3.4: AES-GCM devresinin girişleri ve çıkışları.

Devrenin çıkışında yer alan “dii_data_not_ready” işareti ise bir nevi meşgul işaretidir. “out_data_size” çıkış verisinin uzunluğunu bayt cinsinden ifade eder. “out_last_word” çıkıştaki verinin son kelime olduğunu gösterir. AAD ve şifrelenen veriler kullanılarak oluşturulan etiket (TAG) değeri “tag_vld” sinyali ile “out_data” portundan çıkışa verilir. Standartlara göre 128, 120, 112, 104 ve 96 bit olabilen etiket verisi, bu tasarımda 128 bit uzunluğundadır.

Tüm şifreli kısmi programlama işlemi, tetikleyici işareti aldıktan sonra PROM’ un programlanıp sıfırlanması ile başlar, aynı zamanda AES_GCM devresine de başla işareti (cii_ctl_vld) gönderilir ve IV değeri yazılır. IV değerinden sonra AAD kelimesinin yazılmasından sonra şifreli bit katarı çözülmeye başlanır. 2048 kelimelik bir bit katarı bloğunun çözülmesi ile hesaplanan TAG değeri PROM’dan okunan o bloğa ait TAG ile karşılaştırılarak sonraki blokları çözmeye devam edilir veya TAG’lerin uyuşmaması durumunda kısmi programlamaya son verilir.

GCM algoritmasında kimlik doğrulaması işlemlerinde gereken modüler çarpma işlemi [24]’da belirtilen direk gerçekleştirme yoluyla yapıldığında çarpılan sayıların bit uzunluğu kadar devrenin saat periyodu zaman almaktadır, bizim çalışmamız için 128 bitlik sayılar nedeniyle 128 saat periyodu zaman alır. 128 bitlik sayıların direk çarpılması ise lojik eleman tüketimi açısından pek mümkün olmayacak bir

yöntemdir. Bu nedenle bu tür modüler çarpmalar için donanım tasarımlarına uygun farklı yöntemler geliştirilmiştir. Devrede kullanılan bu tür bir yöntemle modüler çarpma devresinin hızı sekiz saat periyodu sürer ve kaynak kullanımı da kabul edilebilecek düzeydedir.

4. HALKA OSİLATÖR TABANLI KLONLANAMAZ FONKSİYON

4.1 Fiziksel Klonlanamaz Fonksiyon (PUF)

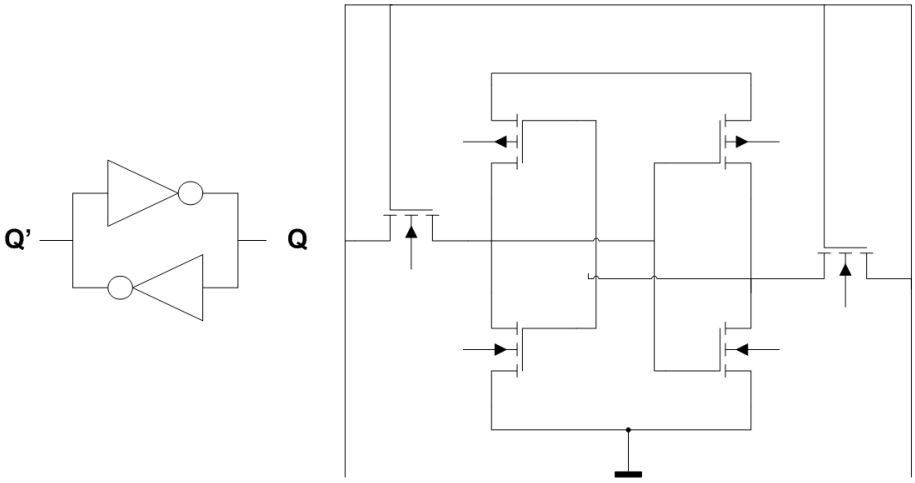
PUF, aygıta özgü bilgi veren, elde edilmesi basit olan fakat aygıt olmadan elde edilebilmesi pratik olarak imkansız olan fiziksel rastgele fonksiyonlardır [26]. Bu tanıma göre PUF'ın en önemli iki özelliği elde edilmesinin basit olması ancak aygıt olmadan elde edilmesinin çok zor olmasıdır.

Literatürde birçok PUF önerilmiştir, bunların hepsi elektronik olarak gerçeklenen PUF'lar değildir, [27–29] çalışmalarında belirtildiği gibi çok çeşitli yöntemlerle gerçeklenmiş olan hatta PUF tanımından önce önerilen yöntemler vardır (optik, akustik, kağıt, kompakt disk, radyo frekans, manyetik, vs.). Elektronik olarak gerçeklenen PUF'lar ise analog elektronik yapılarla, elektronik elemanların gecikmelerine göre ve hafıza elemanlarıyla gerçeklenenler olmak üzere üçe ayrılabilir. Burada da bahsedilecek olan Asıl PUF'lar ise sayısal devrelerdeki gecikmelere göre ve hafıza elemanlarıyla gerçeklenen PUF yapılarıdır ve genel olarak silikon PUF olarak adlandırılmaktadırlar [5, 30].

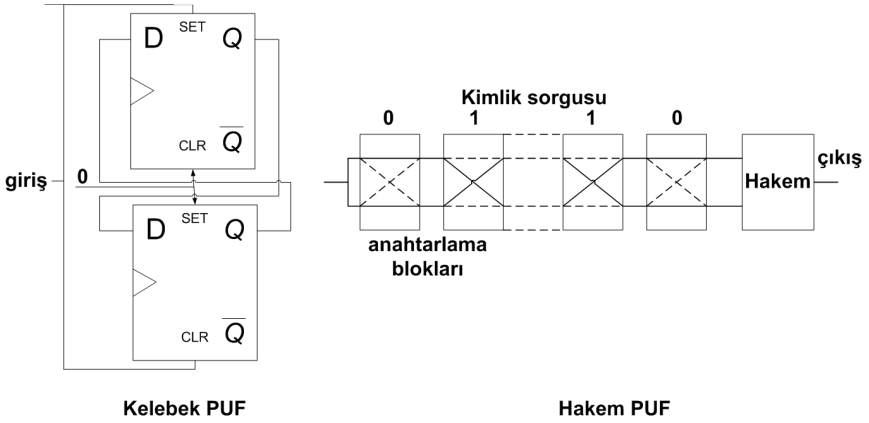
4.1.1 Silikon PUF

Önerilen silikon PUF yapılarından bazıları SRAM, hakem, kelebek ve halka osilatör tabanlı PUF (RO-PUF) yapılarıdır. Bunlardan SRAM PUF, bir SRAM bellek hücresinin elektrik verilince alacağı sayısal çıkış değeridir. Şekil 4.1'de görüldüğü gibi giriş ve çıkışları çapraz bağlı iki eş eviriciden oluşan SRAM hücresinde performans kaygıları nedeniyle iki evirici mümkün olduğunca eş yapılır. Bu devreye elektrik verildiğinde devrenin alacağı ilk değer tamamen rastlantısal bir davranış göstermektedir, bazıları sayısal 1, bazıları sayısal 0 ve bazıları da herhangi bir sayısal değer sayılamayacak değerler almaktadırlar [5]. FPGA'ların birçoğunun yapılandırılma sonrasında sıfırlanması (reset) nedeniyle FPGA gerçeklemeleri için uygun değildir (Şekil 4.1). Şekil 4.2'de görülen kelebek PUF, SRAM PUF ile

aynı mantıkla açıklanabilir. İki eş tutucunun giriş çıkışlarının çapraz bağlanmasıyla elde edilir ve verilecek giriş işaretinden sonra alacağı değer iki tutucu arasındaki farklılıklardan kaynaklanır ve eş tutucular için rastlantısal olması beklenir. Ancak bu PUF türünün de FPGA üzerinde gerçeklemesi yol atama ile ilgili problemler nedeniyle zor olup ayrıca diğer yapılara nispeten fazla alan kaplamaktadır (Şekil 4.2) [31]. Bir başka silikon PUF yapısı olan hakem PUF'ın çalışma ilkesi, sayısal devredeki simetrik iki yoldan geçen sinyallerin yarışa sokulması ve yarışın sonucunun hakem adı verilen bir devre tarafından belirlenmesidir. Simetrik yolların gecikmesinin eş olması sağlanırsa yarışın sonucunu öngörülemeyen ve rastlantısal olan, devreye özgü olması beklenen küçük gecikme farkları belirleyecektir (Şekil 4.2). Bu çalışmada FPGA üzerinde gerçeklemeye uygun olması açısından RO-PUF tercih edilmiştir. RO-PUF aşağıda daha detaylı olarak anlatılmıştır.



Şekil 4.1: SRAM PUF sayısal ifadesi ve devre şeması [5].

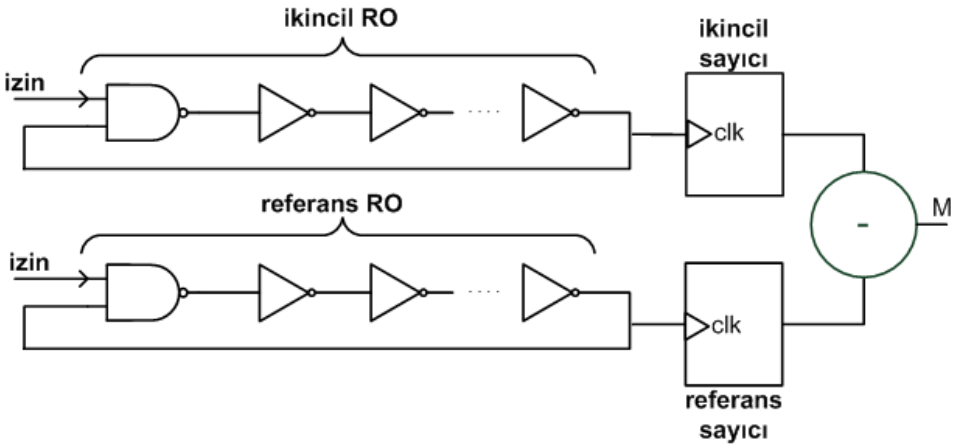


Şekil 4.2: Kelebek ve hakem PUF [5].

4.1.1.1 RO-PUF

RO-PUF yapısı birbirine eş olan iki halka osilatörün frekansları arasındaki farktan yararlanılarak oluşturulur. RO çiftlerinin matematiksel modellemelerindeki gecikmeleri aynı olmalıdır, böylece aralarındaki fark fiziksel yapılarındaki engellenemez ve öngörülmesi pratik olarak imkansız olan farklılıkları ile karakterize olacaktır.

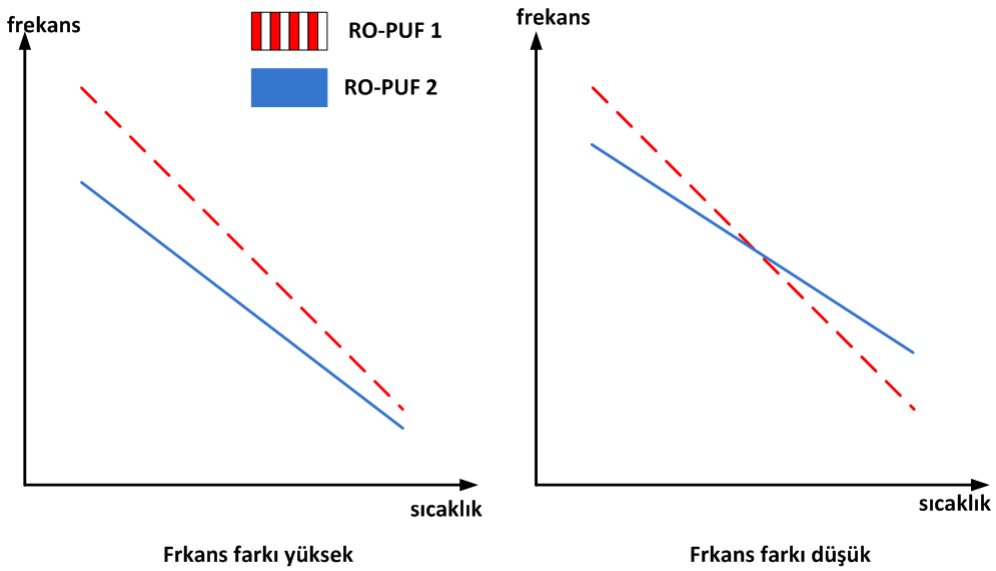
RO çıkışları saat işareti olarak kullanılarak sayıcılar saydırılmaktadır. Aynı anda saymaya başlayan bu sayıcı çiftlerinden birisi belirli bir değere ulaştığında diğer sayıcının değeri kontrol edilerek iki sayıcı karşılaştırılmaktadır (Şekil 4.3). Bu karşılaştırmanın sonucunun büyük veya küçük eşit olmasına göre 0 veya 1 değeri oluşturulur. Eğer RO-PUF çiftlerinin gecikmelerinin eşit olması sağlanırsa sayıcılar arasındaki farkı yaratacak olan etkenler üretim sürecinde kontrol edilemeyen idealsizliklerden ve ortam şartlarından (sıcaklık, radyasyon, gerilim dalgalanmaları, yaşlanma vb.) kaynaklanacaktır. Böylece PUF çıktısı öngörülemeyecektir ve çıkışı rastgele oluşan ve aygıtın kendisine özgü olan bir devre elde edilecektir. Bu şekilde 128 adet RO çifti kullanılarak 128 bitlik rastgele ve aygıtın kimliğini taşıyan bir değer elde edilebilir. Böyle bir rastgele sayı projede kullanılacak şifreleme algoritmasının anahtarı olarak güvenli bir şekilde kullanılabilecektir.



Şekil 4.3: RO-PUF frekans farklarının ölçülmesi [6].

PUF'ın özelliği gereği her IC için farklı olması gerekir, bu özelliğe yongalar arası değişim (inter-chip variation) adı verilir. Diğer bir özelliği de bir yonga için sabit olması, değişmemesidir. Bu özelliğinin adı ise yonga içi değişimidir (intra-chip variation). İdeal olarak yongalar arası değişim %50, yonga içi değişimi ise %0 olmalıdır.

Çalışma ortamının koşullarına göre devre içerisindeki gecikmeler değişebileceğinden PUF çıkışı da her zaman aynı sonucu vermeyebilir. Eğer RO-PUF çiftlerinin frekansları birbirine oldukça yakınsa fiziksel etkenlerle frekans büyüklüklerinin sıralaması Şekil 4.4'de görüldüğü gibi değişebilir [30]. Bu tür bir durumun önlenmesi amacıyla RO-PUF devresine ait istatistiki veriler alınarak analiz edilir ve hata düzeltme kodları kullanılarak gürültünün PUF çıkışını değiştirmesi sorununun önüne geçilebilir. Kaya'nın [6]'de bahsettiği gibi etiketleme yöntemiyle hata düzeltme yapılabilir. Bunun için PUF devresinin farklı koşullar altında ve farklı yongalar üzerinde çalışmasına dair istatistiki çıkış verileri tutulmalıdır. Elde edilen istatistiki verilerle hatalı çıkış verme olasılığı yüksek olan RO-PUF çiftleri belirlenerek etiketlenir ve bu etiket PUF çıkışına dair önemli miktarda bilgi içermez. PUF devresinin her çalışması sırasında bu etiket değerlerinden yararlanılarak PUF çıkışları tekrar değerlendirilir ve hata olasılığı yüksek görülen PUF çıkışları değiştirilir. Bahsedilen yöntemle RO-PUF verisinin sıcaklıkla değişiminin önemli miktarda önlendiği [6]'de gösterilmiştir.



Şekil 4.4: RO-PUF çıkışının sıcaklıkla değişimi.

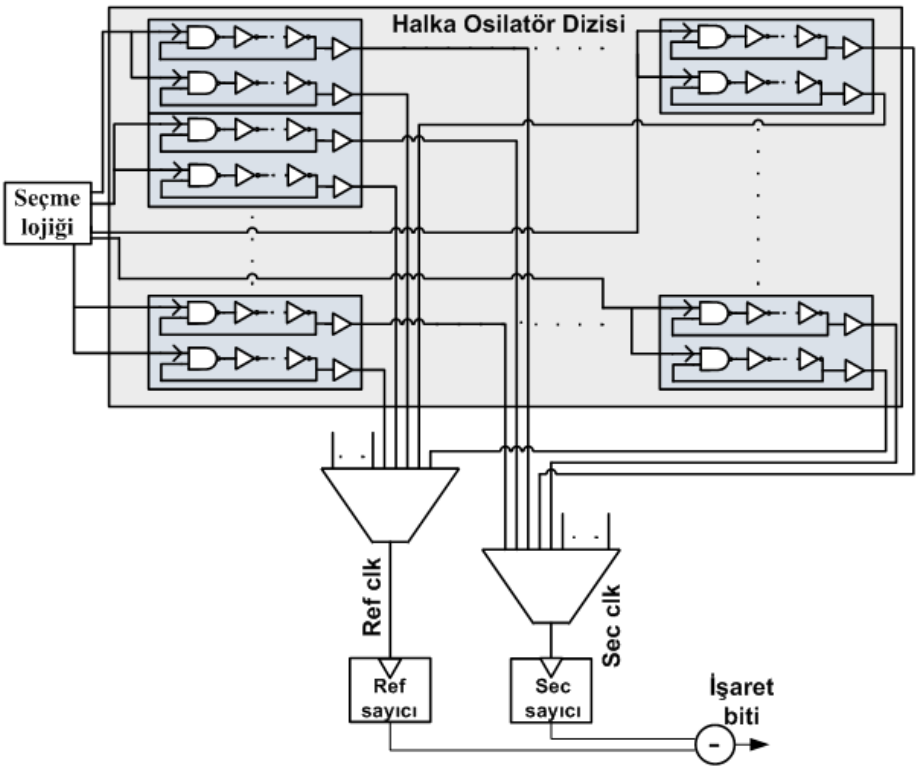
4.2 Donanım Gerçeklemesi

128 RO-PUF çifti birbiri ile karşılaştırılarak 128 bitlik FPGA'ya özgü veri elde etmek amaçlanarak bir devre tasarlandı. FPGA üzerindeki bir CLB içindeki yerel yol atamalarının gecikmeleri bilindiğinden bir RO'yu oluşturacak eviricilerin tek bir CLB içinde tutulması sağlandı. Aynı yapının başka bir CLB'de gerçeklenmesiyle de eş bir RO gerçeklenmektedir. Virtex 5 FPGA ailesinde bir karmaşık lojik bloğu (CLB) içerisinde 2 dilim (SLICE), bir dilim içerisinde de 4 LUT bulunur, yani bir CLB 8 LUT içerir. Tek bir RO-PUF devresi 7 evirici kapının art arda bağlanmasından oluşmaktadır. Her bir evirici için FPGA'da bir LUT kullanılmıştır. Bir LUT'un da RO çıkışında tampon (buffer) olarak kullanılmasıyla bir RO-PUF 8 LUT'dan oluşur ve bir CLB'deki bütün LUT'ları kullanır. 128 çift RO dizisi çoğullayıcılara bağlanıp teker teker seçilerek frekansları karşılaştırılmaktadır (Şekil 4.5).

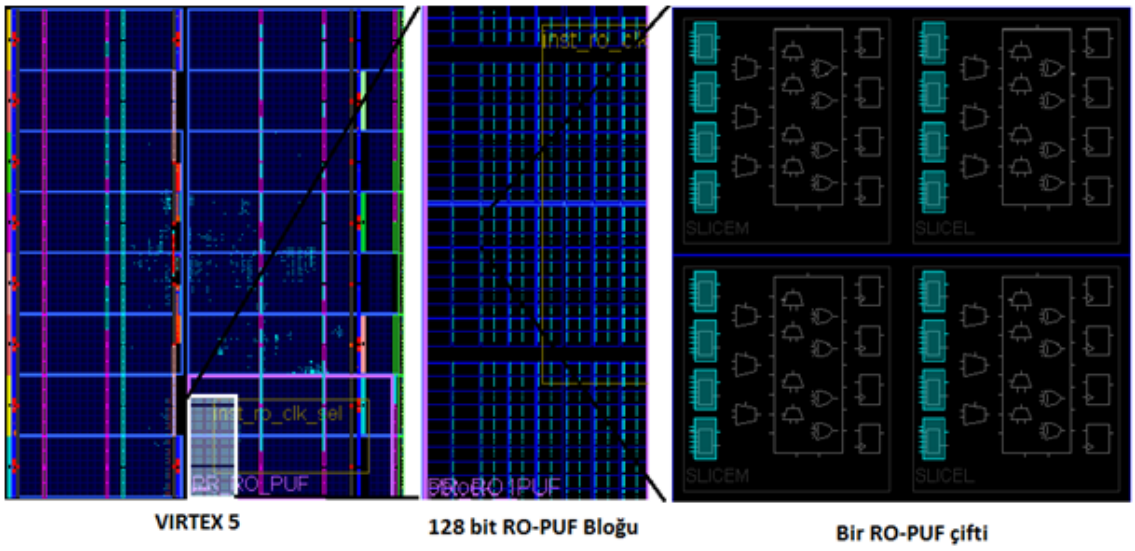
PUF oluşturan LUT'ların aynı FPGA dilimi içinde bulunması sağlanarak yol atama (routing) gecikmelerinin sabitlenmesi amaçlanmıştır. Böylece RO-PUF çiftleri komşu CLB'lere yerleştirilerek çiftlerin teorik gecikmeleri eşitlenmiştir (Şekil 4.6). Frekans farkı yaratacak gecikme FPGA'nın öngörülemeyen üretim karakteristiklerine ve ortam şartlarına bağlı hale gelmiştir. RO-PUF çiftlerinin FPGA üzerinde komşu CLB'lere bir blok halinde dizilmesiyle de ortam şartlarının bütün RO-PUF çiftlerini benzer ölçüde etkileyeceği söylenebilir (Şekil 4.6). Bu durum yonga içi değişimi özelliğini ideale yakın kılar.

128 bitlik RO-PUF çifti FPGA üzerinde DPR ile gerçeklenerek çalışması test edilmiştir. RO-PUF dizisinin bulunduğu alan PRR olarak belirlenmiştir. İlk yüklenen tasarımda PRM olan tüm bir 128 RO-PUF çifti ve statik tasarım parçaları olan frekans farkının ölçüldüğü sayaçların ve çoğullayıcıların olduğu devre parçası ve bilgisayarla haberleşme sağlanan UART kontrolör devresi bulunmaktadır. İkinci kısmi tasarımda ise 128 adet RO-PUF çiftinin olduğu ilk PRM'deki aynı devre yer almaktadır.

RO-PUF dizisinin DPR yöntemi ile incelenmesindeki amaç PUF devresinin kısmi olarak yüklenmesi durumunda davranışındaki değişimin gözlenmesidir. RO-PUF verisi bilgisayara RS-232 haberleşme protokolüyle MATLAB yazılımı üzerinden alınıp incelenmiştir. PR projede 2 RO-PUF bloğu kısmi yapılandırılabilir blok olarak



Şekil 4.5: RO-PUF devre yapısı [6].

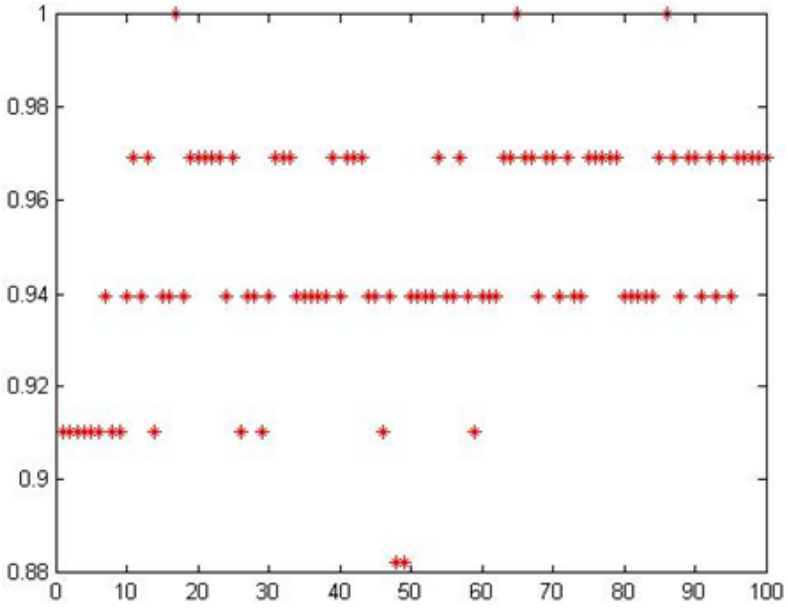


Şekil 4.6: FPGA serimi ve PUF yerleşimi.

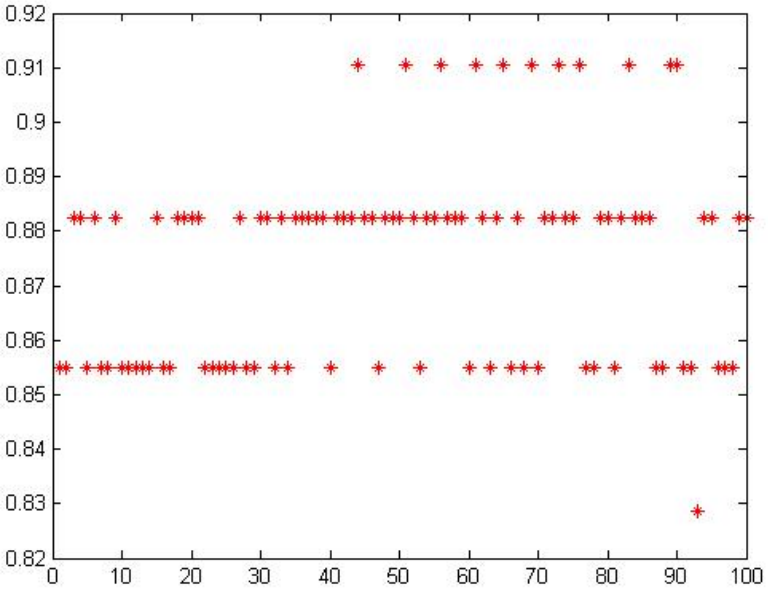
FPGA'ya yüklenmiştir. Her iki yapılandırmadaki RO-PUF verisi 100 defa okunarak verideki 0 ve 1 oranı MATLAB yazılımında çizdirilmiştir.

Şekil 4.7'de birinci RO-PUF dizisinin, Şekil 4.8'de ise DPR ile ilk tasarımdaki RO-PUF dizinin yerine yüklenen ancak ilk devre ile tamamen aynı olan ikinci RO-PUF dizisinin 1/0 oranı görülmektedir. Her iki şekilde de oranın olması gereken 1 değerine

oldukça yakın değerler aldığı görülüyor. Yapılan ölçümlerde her iki yapılandırmadaki RO-PUF dizilerinin davranışlarının birbirine benzer olduğu sonucu çıkarılmaktadır. Buradan DPR işleminin PUF çıkışı üzerinde beklenmeyen bir yan etkisi olmadığı söylenebilir. İkinci RO-PUF dizisinin verilerinde gözlenen değişimler normal çalışma sırasında görülmesi beklenen değişimlerden farklı olmamıştır.



Şekil 4.7: Birinci PUF'ın 1/0 oranı (100 deneme için).



Şekil 4.8: İkinci PUF'ın 1/0 oranı (100 deneme için).

5. SONUÇ VE ÖNERİLER

Tüm sistem Xilinx ISE 13.2 ve PlanAhead yazılımları kullanılarak sentezlenmiştir. Tasarım dili olarak VHDL ve Verilog kullanılmıştır. Tasarımın kaynak kullanımı Çizelge 5.1'de verilmiştir. Tasarımın benzetimleri için Modelsim 6.5d yazılımı, yonga üzerinde hata ayıklamak için de Xilinx ChipScope yazılımı kullanılmıştır.

Çizelge 5.1: Kaynak kullanımı.

	LUT	FF	36K BRAM	ICAP	PLL
	4800	3212	6	1	1
PUF_KONTR	2579	491	-	-	-
TOPLAM	7379	3703	6	1	1
MEVCUT	69120	69120	148	2	6
TOPLAM %	10,7	5,35	4	50	16

Maksimum 177 MHz frekansında çalışabilecek şekilde sentezlenen tasarımın PROM kontrolör devresi 33 MHz, diğer devreleri 100 MHz frekanslı saatlerle çalıştırılmıştır. PROM kontrolör devresinin hızı BPI PROM yongasına bağlı olarak belirlenmiştir. 100 MHz frekansı ise Virtex 5 üzerinde ICAP devresinin maksimum çalışma frekansı olarak verilmiştir [32]. ICAP devresi veri portu 8, 16 veya 32 olarak çalıştırılabilmektedir [18]. Tasarımda ICAP maksimum frekansında ve veri portu 32 bit genişliğinde çalıştırılarak maksimum verimlilikle çalıştırılmıştır.

Tasarımda kullanılan PLL kart üzerindeki 100 MHz saat işaretinden devre için gereken farklı saatleri üretmek için kullanılmıştır. Micron firmasına ait BPI PROM'un dokümanında çalışabileceği en yüksek frekans 40 MHz olarak verilmiştir. Ancak devre üzerinde yapılan denemelerde BPI PROM en yüksek 33 MHz frekansında çalıştırılabilmiştir. Sistemin geriye kalan devreleri 100 MHz saat frekansı ile çalıştırıldığından, sistemi hız olarak sınırlayan PROM kontrolör devresidir.

Kısmi yeniden yapılandırma özelliğini denemek için basit lojik işlemler yapan iki PRM kullanılmıştır. ML505 kartı üzerindeki anahtar butonları giriş sinyali olarak, LED'ler de çıkış sinyali olarak PRM tasarımlarına bağlanmıştır. İlk PRM girişten aldığı sinyale

göre EXOR veya AND sayısal işlemlerini yaparken ikinci PRM EXNOR veya OR sayısal işlemlerini yapmaktadır.

Kısmi yeniden yapılandırma sırasında statik tasarımın çalışmasını kontrol etmek amacıyla statik tasarıma basınç ses dönüştürücü bir devre eklenmiştir. Çalışması kart üzerindeki anahtara bağlanan dönüştürücü PR işlemi sırasında açık tutulmuş ve ikinci PRM'in yüklenmesi sırasında çalışmaya devam ettiği gözlenmiştir. Böylece DPR özelliğinin gerçeklendiği, kısmi tasarım yüklenirken statik tasarımın çalışmaya devam ettiği sonucu çıkarılmıştır. İkinci tasarım yüklendikten sonra da LED'lerin ikinci PRM devresinin işleminin sonucunu verdiği gözlenmiştir.

Şifreleme ve kısmi yeniden yapılandırma kontrolör devresinin kritik işaretleri de ChipScope yazılımı ile gözlenmiştir ve kimlik doğrulama işlemlerinin düzgün bir şekilde çalıştığı ve ikinci PRM tasarımın başarılı bir şekilde yüklendiği gözlenmiştir. PR işlemi sırasında ChipScope ile gözlem yapılabilmesi statik tasarımın PR işlemi sırasında çalışmasını başka bir şekilde göstermiştir.

Literatürde yapılan benzer bir çalışmada [33], PR kontrolör, AES ve GCM devrelerinin toplam kaynak kullanımı Virtex 5 LX50T üzerinde 3040 LUT, 2166 FF ve 13 BRAM olarak verilmiştir. Çizelge 5.1'de verilen kaynak kullanımı ile karşılaştırıldığında BRAM haricindeki kaynakların bizim tasarımımızda daha çok harcandığı görülüyor. Aynı çalışmada ICAP üzerinden PR işleminin işlemci ile ve donanımsal PR kontrolör devresiyle gerçeklemesi kıyaslanmaktadır. Sonuç olarak işlemcili çözümlerin PRM yüklenme zamanını binlerce kat yavaşlattığı gösterilmiştir. Bu nedenle işlemci ile PR kontrolör tasarımı gerçek zamanlı uygulanabilir PR tasarımlar açısından uygun değildir.

5.1 Çalışmanın Uygulama Alanı

Literatürdeki benzer çalışmaların PUF verisinin anahtar olarak kullanımı hariç kaynak kullanımı bu çalışmanın kaynak kullanımından bir miktar daha azdır [33]. Tasarlanan sistem üzerinde PRM olarak herhangi bir tasarım kullanılarak, PRM değiştirilerek kullanılabilir. PUF devresi veya hata düzeltme algoritması değiştirilmek istenirse sisteme kolayca tümleştirilerek farklı IP güvenliği uygulamaları için kullanılabilir. Sistemin tasarım aşamalarını otomatik olarak düzenleyip koşturabilecek bir yazılım

hazırlanarak otomatize edilebilecek olan bu çalışma ile farklı PRM devreleri kolaylıkla sisteme dahil edilerek kullanılabilir.

KAYNAKLAR

[1] **DOC**, DEFENSE INDUSTRIAL BASE ASSESSMENT: COUNTERFEIT ELECTRONICS, http://www.bis.doc.gov/defenseindustrialbaseprograms/osies/defmarketresearchrpts/final_counterfeit_electronics_report.pdf, alındığı tarih: 13.12.2012.

[2] **Rodríguez-Henríquez, F.**, **Pérez, A.D.**, **Saqib, N.A. ve Koç, C.K.** *Cryptographic Algorithms on Reconfigurable Hardware*.

[3] **Farooq, U.**, **Marrakchi, Z. ve Mehrez, H.** (2012). *Tree-based Heterogeneous FPGA Architectures*, Springer New York.

[4] FPGA Logic Cells Comparison, http://www.1-core.com/library/digital/fpga-logic-cells/, alındığı tarih: 02.05.2013.

[5] **Maes, R. ve Verbauwhede, I.** (2010). Physically Unclonable Functions: A Study on the State of the Art and Future Research Directions, *Towards Hardware-Intrinsic Security*.

[6] **Kaya, G.** (2012). Reliability Enhancement of Ring Oscillator Based Physically Unclonable Functions, *İstanbul Teknik Üniversitesi(İTÜ)*.

[7] **OECD**, (1998), The Economic Impact of Counterfeiting, http://www.oecd.org/dataoecd/11/11/2090589.pdf.

[8] **Barak, S.**, (2012), Counterfeit parts putting military at risk, http://www.eetimes.com/electronics-news/4236345/Counterfeit-parts-putting-military-at-risk-?cid=NL_EETimesDaily.

[9] SANTEZ Proje Başvuru Dokümanı Proje Kodu: 00957.STZ.2011-2.

[10] **McNeil, S.**, White Paper: Solving Today's Design Security Concerns, http://www.xilinx.com/support/documentation/white_papers/wp365_Solving_Security_Concerns.pdf, alındığı tarih: 08.05.2012.

[11] **Maxfield, C.**, Xilinx redefines the non-volatile FPGA landscape, http://www.eetimes.com/electronics-products/fpga-pld-products/4091876/Xilinx-redefines-the-non-volatile-FPGA-landscape, alındığı tarih: 18.02.2013.

[12] **Lattice Semiconductor**, http://www.latticesemi.com/, alındığı tarih: 18.04.2013.

[13] **Reconfigurable computing**, http://en.wikipedia.org/wiki/Reconfigurable_computing, alındığı tarih: 12.04.2013.

[14] **Field-programmable gate array**, http://en.wikipedia.org/wiki/Field-programmable_gate_array, alındığı tarih: 03.05.2013.

[15] **Fons, F.**, **Fons, M.**, **Cantó, E. ve López, M.** (2011). Real-time embedded systems powered by FPGA dynamic partial self-reconfiguration: a case study oriented to biometric recognition applications, *Journal of Real-Time Image Processing*.

[16] **Lie, W. ve Feng-yan, W.** (2009). Dynamic Partial Reconfiguration in FPGAs, *Third International Symposium on Intelligent Information Technology Application (IITA)*, ***2***, 445 – 448.

[17] **Eto, E.**, Difference Based Partial Reconfiguration Application Note, http://www.xilinx.com/support/documentation/application_notes/xapp290.pdf, alındığı tarih: 13.12.2012.

[18] Virtex-5 FPGA Configuration User Guide, http://www.xilinx.com/support/documentation/user_guides/ug191.pdf, alındığı tarih: 18.11.2011.

[19] Xilinx Partial Reconfiguration User Guide, www.xilinx.com/support/documentation/sw_manuals/xilinx12_3/ug702.pdf, alındığı tarih: 29.06.2006.

[20] Advanced Encryption Standard, http://en.wikipedia.org/wiki/Advanced_Encryption_Standard, alındığı tarih: 26.04.2013.

[21] **FIPS** (2001). Announcing the ADVANCED ENCRYPTION STANDARD (AES), *Federal Information Processing Standards (FIPS) Yayınları 197*.

[22] **Schneier, B.** (1996). *Applied Cryptography, Second Edition: Protocols, Algorthms and Source Code in C*, John Wiley & Sons, Inc.

[23] **Dworkin, M.** (2007). Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, *National Institute of Standards and Technology*.

[24] **McGrew, D.A. ve Viega, J.**, The Galois/Counter Mode of Operation (GCM), http://www.csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/gcm/gcm-revised-spec.pdf.

[25] **Ahmad, T.**, OpenCores Projesi:Galois Counter Mode Advanced Encryption Standard GCM-AES, http://opencores.org/project,gcm-aes.

[26] **Gassend, B.** (2003). Physical Random Functions, *Massachusetts Tekonoloji Enstitusu (MIT)*.

[27] **Tuyls, P.**, **Skoric, B.**, **Stallinga, S.**, **Akkermans, A. ve Ophey, W.** (2005). Information theoretical security analysis of physical unclonable functions, *Proceedings of Conference on Financial Cryptography and Data Security 2005, volume 3570 of Lecture Notes in Computer Science*.

[28] **Skoric, B.**, **Tuyls, P. ve Ophey, W.** (2005). Robust key extraction from physical unclonable functions, *Proceedings of the Applied Cryptography and Network Security Conference 2005, volume 3531 of Lecture Notes in Computer Science*.

[29] **Pappu, R.** (2001). Physical One-Way Functions. PhD tezi, *Massachusetts Tekonoloji Enstitüsü (MIT)*.

[30] **Suh, G.E. ve Devadas, S.** (2007). Physical unclonable functions for device authentication and secret key generation, *Design Automation Conference - DAC 2007*.

[31] **Verbauwhede, I.**, **Tuyls, P. ve Maes, R.** (2008). Intrinsic PUFs from flip-flops on reconfigurable devices, *3rd Benelux Workshop on Information and System Security (WISSec 2008)*.

[32] Virtex-5 FPGA Data Sheet: DC and Switching Characteristics, `http://www.xilinx.com/support/documentation/data_sheets/ds202.pdf`, alındığı tarih: 05.05.2010.

[33] **Hori, Y.**, **Satoh, A.**, **Sakane, H. ve Toda, K.** (2008). Bitstream encryption and authentication with AES-GCM in dynamically reconfigurable systems, *International Conference on Field Programmable Logic and Applications*.

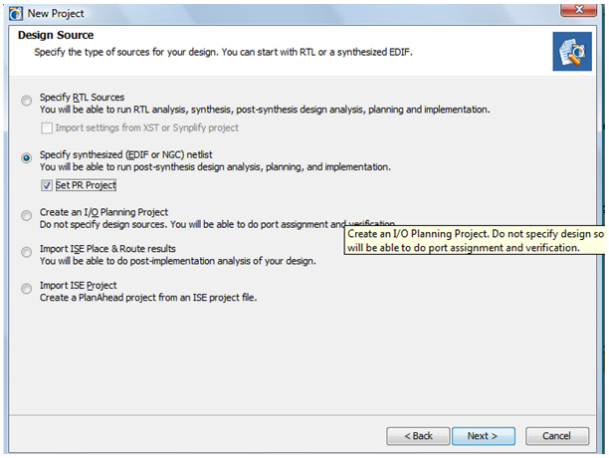
EKLER

EK A.1 : PlanAhead ile PR Tasarım

EK A.1

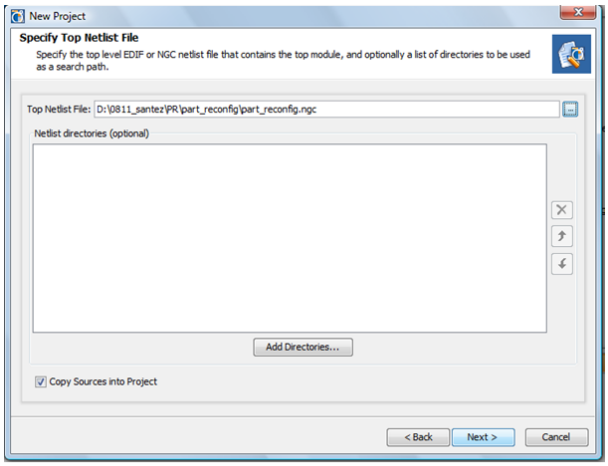
1.1 PlanAhead ile PR Tasarım

Xilinx ISE Design Suite ile beraber yüklenen PlanAhead yazılımında File -> New Project seçilerek açılan pencerede proje adı ve konumunu girdikten sonra Şekil A.1'deki pencerede "Specify synthesized (EDIF or NGC) netlist" seçeneği altındaki "Set PR Project" ile seçilir.



Şekil A.1: PR proje oluşturma

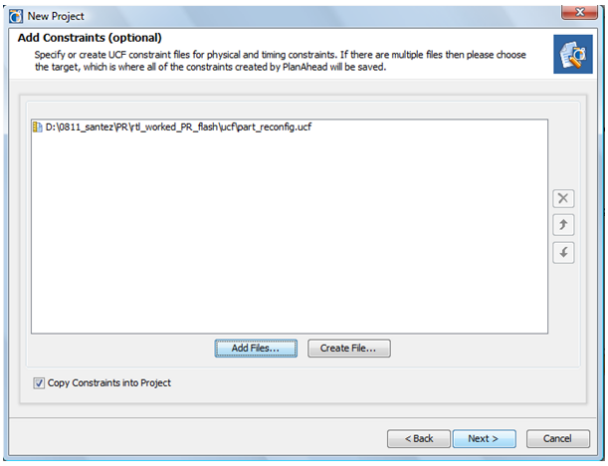
Sentezlenmiş üst modülün sentez dosyası (*.ngc) Şekil A.2 gibi eklenir. Tek bir dosya yerine sentez dosyalarının bulunduğu bir klasör de girilebilir.



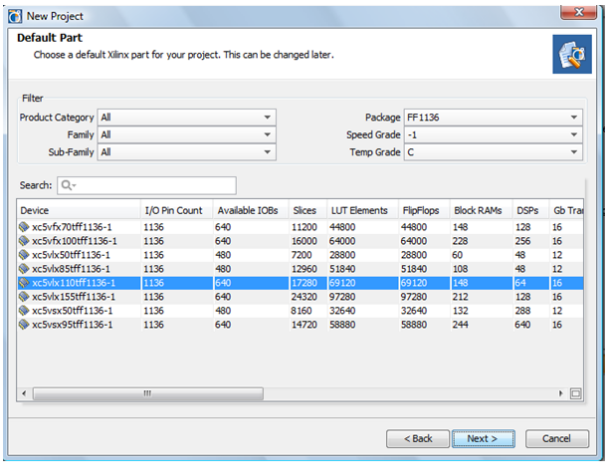
Şekil A.2: Üst devrenin sentez dosyasının seçilmesi

Bir sonraki pencerede tasarımın kısıt dosyası (*.ucf) Şekil A.3'deki gibi eklenir.

Bir sonraki pencerede Şekil A.4'deki gibi tasarımın gerçekleneceği FPGA seçilerek devam edilir.

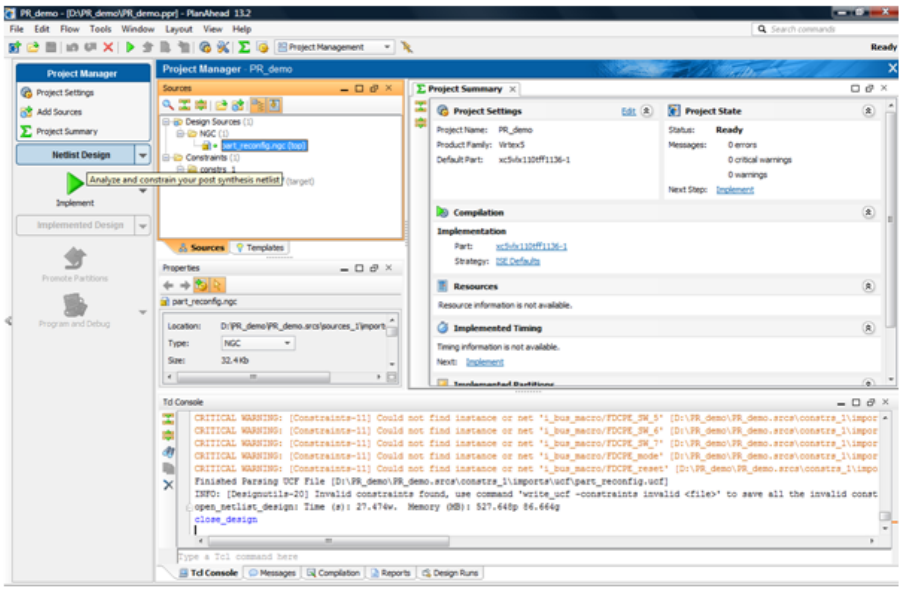


Şekil A.3: Kısıt dosyası ekleme



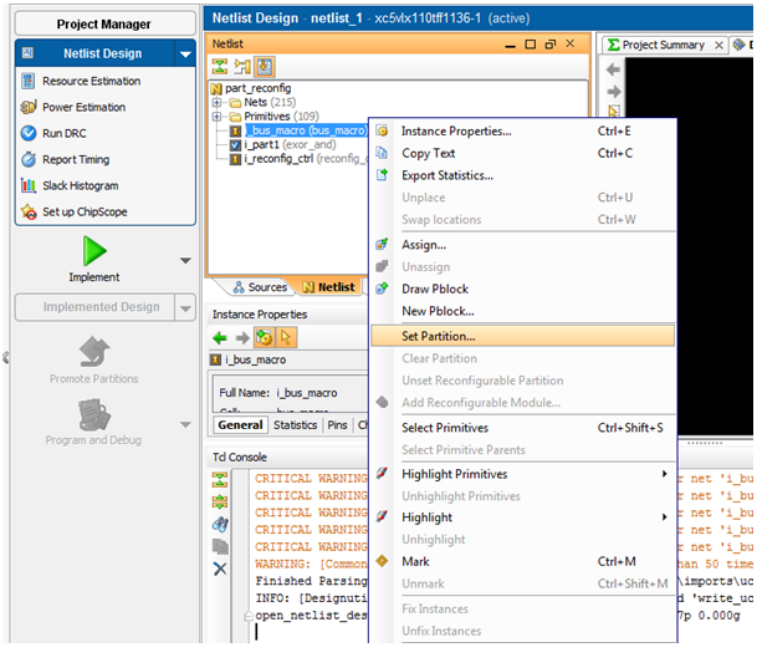
Şekil A.4: FPGA'nın seçimi ve proje oluşturmanın son adımı

Böylece PlanAhead ile bir PR tasarım açtık. Şekil A.5'deki gibi pencerenin solundaki "Netlist Design" tıklanarak tasarımın netlist i açılır.



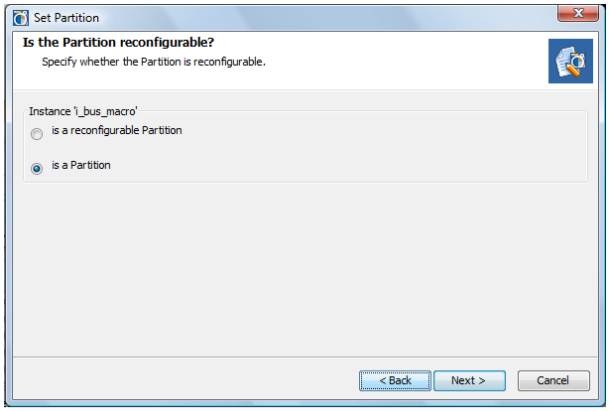
Şekil A.5: Netlist açma

Açılan netlist penceresinden “Black Box” ve PR modüller, sağ tık ile açılan “Set Partition” seçeneği ile sentez dosyaları eklenir. (Şekil A.6)



Şekil A.6: PRM ve "black box" alanları belirleme

Bus macro gibi statik modüller Şekil A.7‘deki gibi “is a Partition” seçeneği ile ayarlanır.



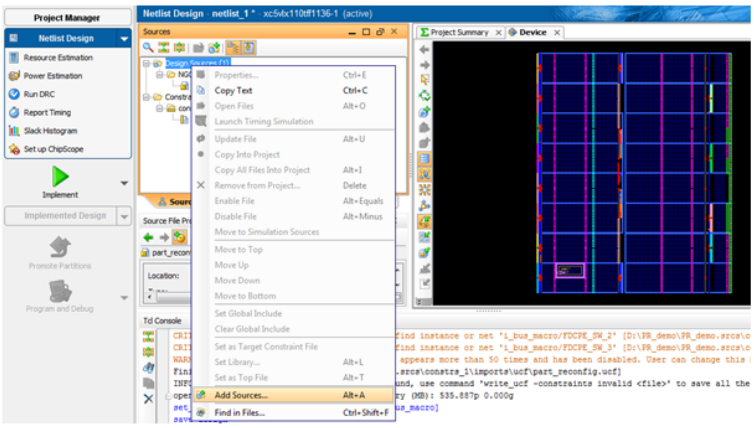
Şekil A.7: Bus macro ekleme

Partition olarak seçilen “Black Box” modüllerin sentez dosyaları “Sources” sekmesinden eklenir. (Şekil A.8)

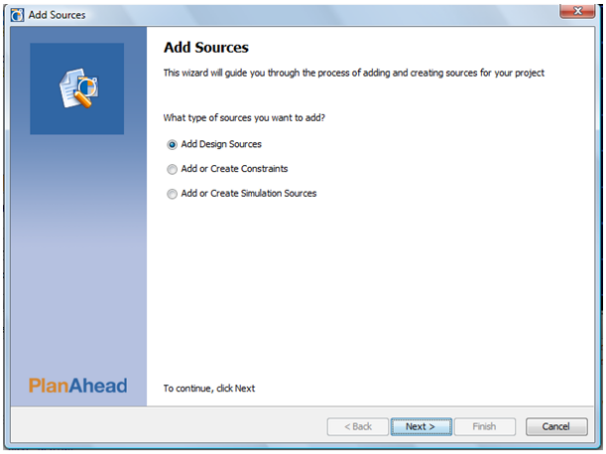
Tasarıma yeni bir modül eklenirken ilk olarak dosyanın cinsi seçilir. Tasarım, kısıt veya simülasyon dosyası seçeneklerinden biri ile devam edilir. (Bus macro için tasarım seçeneği) (Şekil A.9)

Bir sonraki pencerede sentez dosyası taranarak eklenir. (Şekil A.10)

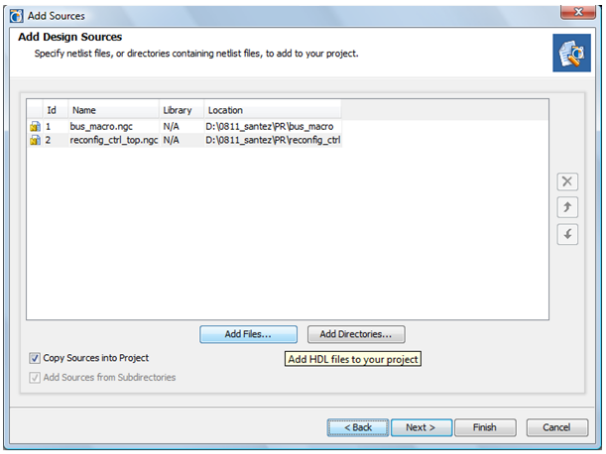
PR modüller “Set Partition” penceresinde “is a reconfigurable Partition” seçilerek eklenir. Sonraki adımlarda PR modülün adı, dosyanın yeri belirlenerek ve eğer kısıt dosyası varsa eklenerek PR modül ekleme işlemi bitirilir.(Şekil A.11)



Şekil A.8: Tasarıma dosya ekleme 1



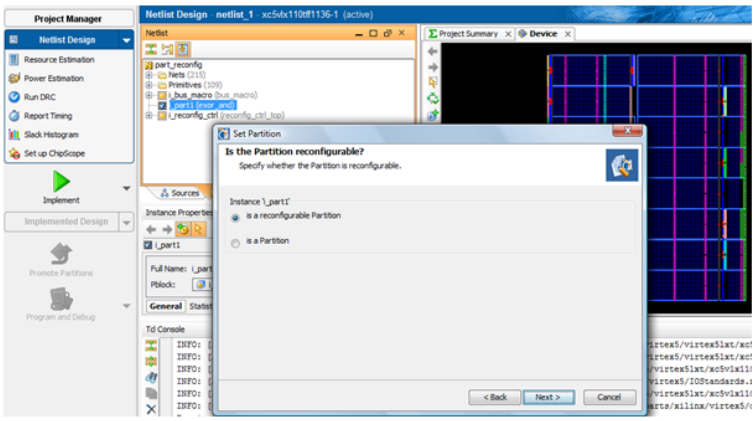
Şekil A.9: Tasarıma dosya ekleme 2



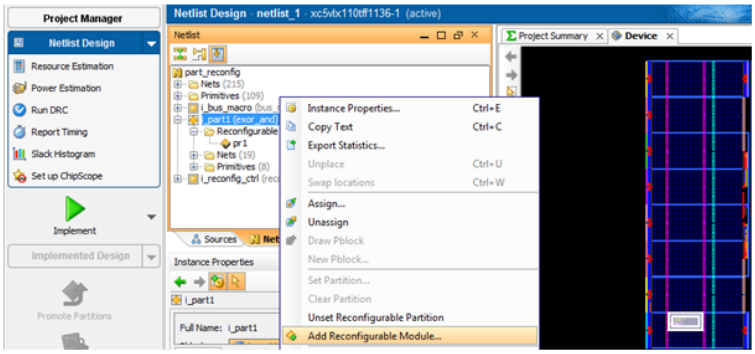
Şekil A.10: Bus macro ve PR sentez dosyalarının eklenmesi

Diğer PR modülleri eklemek için "Netlist" sekmesinde PR partition sağ tıklanarak "Add Reconfigurable Module" seçilir. (Şekil A.12)

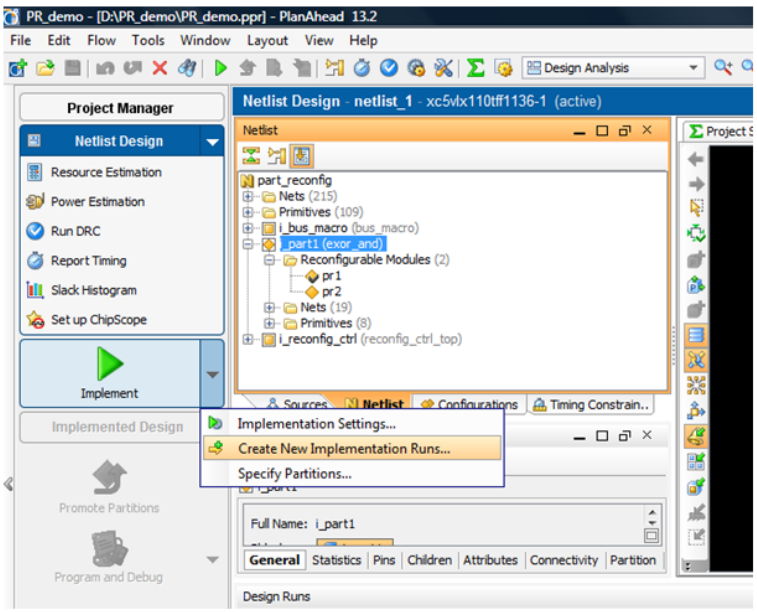
Her PR modül için ayrı gerçekleme yapılır. PR modüllerden biri altın tasarım "Golden Design" seçilerek statik modüller bu tasarımdan alınır. Yeni gerçekleme ortamını hazırlamak için pencerenin solundaki "Implement" butonundan "Create New Implementation Runs" tıklanır. (Şekil A.13)



Şekil A.11: PRR atama



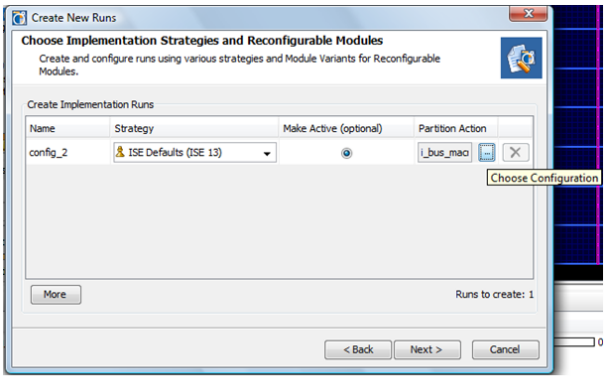
Şekil A.12: Diğer PRM sentez dosyalarının eklenmesi



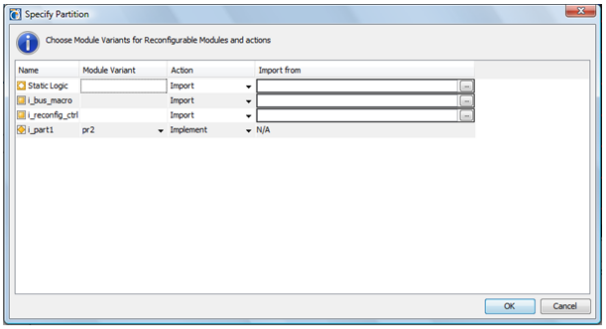
Şekil A.13: Yeni gerçekleme ortamı oluşturma

Yeni gerçeklemede olacak modülleri seçmek için “Partition Action” tıklanır. (Şekil A.14)

“Action” kolonunda statik modüller “Import” ile altın tasarımdan olduğu gibi alınır. Yeni PR modülü “pr2” “Implement” seçilir. (Şekil A.15)

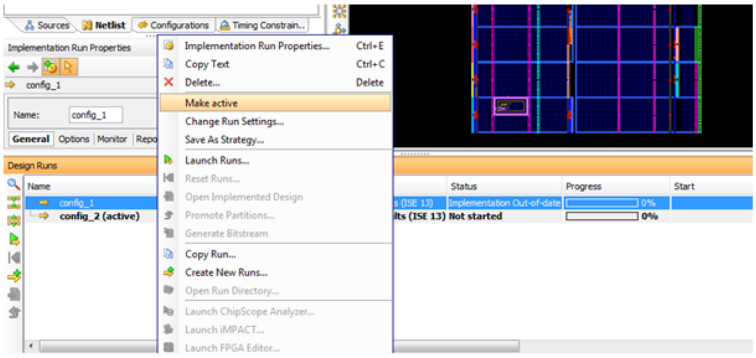


Şekil A.14: Gerçeklemedeki PRM'in seçimi 1



Şekil A.15: Gerçeklemedeki PRM'in seçimi 2

PlanAhead penceresinin altındaki "Design Runs" sekmesinde bütün gerçeklemeler ve durumları görülebilir. Gerçeklemeler sağ tıklanarak "Make Active" ile seçilebilir. (Şekil A.16) Bütün gerçeklemeler bu şekilde ayrı ayrı bit katarı oluşturma aşamasına kadar koşturulur.

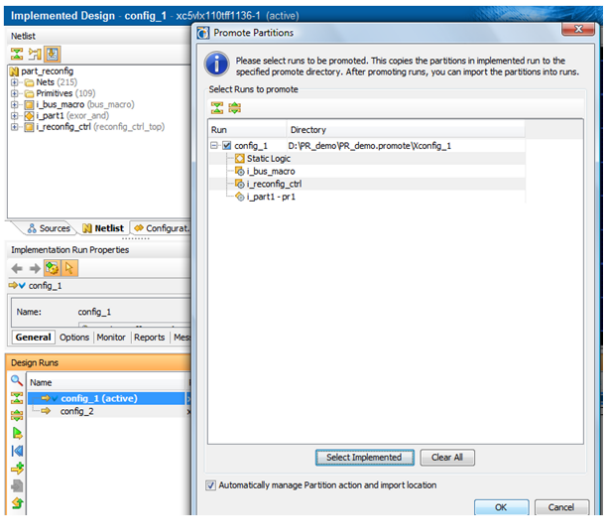


Şekil A.16: Gerçeklemeler arası geçiş

Altın tasarım seçilen gerçekleme "Design Runs " sekmesinde sağ tık ile "Promote Partitions" seçilerek statik modülleri diğer gerçeklemelere aktarılır. (Şekil A.17)

Her gerçekleme için iki bit katarı oluşturulur, biri PRM içeren FPGA tasarımının bulunduğu tam bit katarı diğeri ise PRM kısmının bulunduğu kısmi bit katarı. PR için ek olarak aşağıdaki "Generate Bitstream" seçenekleri girilir. (Şekil A.18)

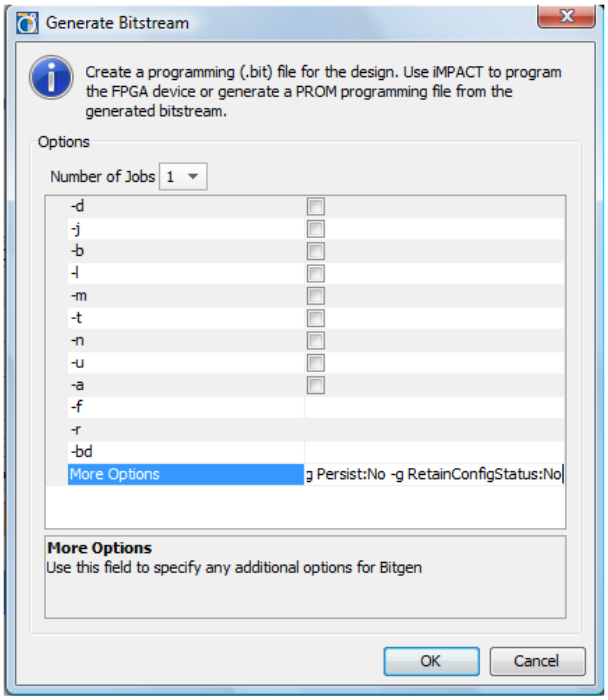
-g ActiveReconfig:Yes
-g Binary:Yes
-g ConfigFallback:Disable



Şekil A.17: Altın tasarım seçme

-g CRC:Enable
-g Persist:No
-g RetainConfigStatus:No

Bu seçeneklerden "Persist" seçeneği PR işleminin gerçekleştirileceği arayüz ile ilgilidir, FPGA'nın konfigürasyon pinlerinin kullanıcı için ayrılması veya SelectMap için tahsis edilmesini belirler. Eğer PR işlemi için SelectMap kullanılacaksa "-g Persist:Yes" olarak ayarlanmalıdır. Bu çalışmada bit katarı PROM'dan kullanıcı lojiği tarafından okunup ICAP üzerinden yükleneceği için "-g Persist:No" olarak seçilmiştir.



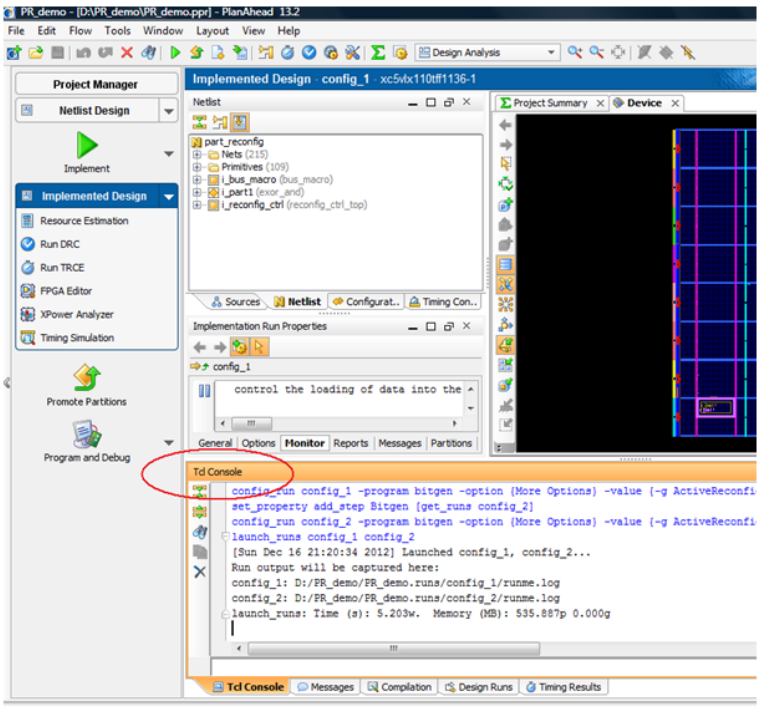
Şekil A.18: Bit katarı oluşturma

PlanAhead proje klasörünün altındaki .\<proje_adı>.runs klasöründeki “configX” (X: gerçekleme numarası) klasörlerinde her gerçekleme için bütün ve kısmi tasarımları içeren bit dosyaları (*.bit) bulunabilir.

1.1.1 Script oluşturma

Bütün bu PR tasarım aşamaları oldukça uğraştırıcı rutin işlemler gerektirir. Başlangıç için özellikle birçok kez deneme yapılabileceğinden sıkıcı ve yorucu olabilen bu işlemler aynı zamanda tasarım süresini de uzatır. PlanAhead yazılımının TCL komutları kullanılarak bütün bu işlemler script yazılarak kullanılabilecek bir BATCH dosyası çalıştırılarak otomatik olarak yapılabilir.

TCL komutları için PlanAhead penceresinde alt kısımdaki “TCL Console” sekmesindeki komutlar kopyalanarak bir *.tcl dosyasına yazılır. (Şekil A.19) Batch dosyasında bu *.tcl dosyası source edilerek PlanAhead komut satırından çağrılabilir. ISE nin sentezleme programı XST de komut satırından kullanılarak bütün bir tasarım tek bir seferde BATCH dosyası çalıştırılarak daha hızlıca tamamlanabilir.



Şekil A.19: Script konsolu

ÖZGEÇMİŞ

Ad Soyad: Burak Gövem

Doğum Yeri ve Tarihi: Karabük, 1987

Adres:

E-Posta:

Lisans: İTÜ Elektronik Mühendisliği, Şubat 2010

Mesleki Deneyim ve Ödüller:

Yayın ve Patent Listesi: