İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

FPGA ÜZERİNDE MAC/IP/UDP PROTOKOLÜNÜN GERÇEKLENMESİ

YÜKSEK LİSANS TEZİ

Servet AYOK

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Doç. Dr. Müştak Erhan YALÇIN

HAZİRAN 2013

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

FPGA ÜZERİNDE MAC/IP/UDP PROTOKOLÜNÜN GERÇEKLENMESİ

YÜKSEK LİSANS TEZİ

Servet AYOK
504071221

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Doç. Dr. Müştak Erhan YALÇIN

HAZİRAN 2013

İTÜ, Fen Bilimleri Enstitüsü'nün 504071221 numaralı Yüksek Lisans Öğrencisi **Servet AYOK**, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı **"FPGA ÜZERİNDE MAC/IP/UDP PROTOKOLÜNÜN GERÇEKLENMESİ"** başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Doç. Dr. Müştak Erhan YALÇIN (İTÜ)**

Jüri Üyeleri : **Yrd. Doç. Dr. Metin YAZGI (İTÜ)**

Yrd. Doç. Dr. Fethullah KARABİBER (YTÜ)

Teslim Tarihi : **19 Eylül 2013**
Savunma Tarihi : **24 Eylül 2013**

Eşime ve Aileme,

ÖNSÖZ

En başta, yüksek lisans öğrenimim boyunca bana verdiği destek ve gösterdiği anlayıştan dolayı değerli hocam, tez danışmanım Doç. Dr. Müştak Erhan Yalçın'a teşekkür ederim.

Bugünlere gelmemde en büyük emeği olan canım aileme ve zor günlerimde hep yanımda olan sevgili eşime teşekkürlerimi sunarım.

Ayrıca, çalışmalarım sırasında desteklerini esirgemeyen değerli arkadaşlarım Hilmi Öztürk, Cem Ayyıldız ve Fatih Sadıç'a teşekkürlerimi borç bilirim.

Önsözün sonunda, yüksek lisans tezime fırsat sağlayan tüm Turkcell Teknoloji ekibine teşekkür ederim.

Eylül 2013

Servet Ayok
Elektronik Mühendisi

İÇİNDEKİLER

Sayfa

KISALTMALAR

OSI	**:** Open Systems Interconnection
ISO	**:** International Organization for Standardization
CRC	**:** Cyclic Redundancy Check
MAC	**:** Medium Access Control
ARP	**:** Address Resolution Protocol
DEC	**:** Digital Equipment Corporation
CSMA/CD	**:** Carrier Sense Multiple Access with Collision Detection
IP	**:** Internet Protocol
UDP	**:** User Datagram Protocol
TCP	**:** Transmission Control Protocol
PCS	**:** Physical Coding Sublayer
PMA	**:** Physical Medium Attachment
PMD	**:** Physical Medium Dependent
MII	**:** Media Independent Interface
GMII	**:** Gigabit Media Independent Interface
RGMII	**:** Reduced Gigabit Media Independent Interface
SGMII	**:** Serial Gigabit Media Independent Interface
IFG	**:** Inter Frame Gap
IPv4	**:** Internet Protocol Version 4
ICMP	**:** Internet Control Message Protocol
TTL	**:** Transistor Transistor Logic
MOS	**:** Metal Oxide Semiconductor
ASIC	**:** Application Specific Integrated Circuit
PLD	**:** Programmable Logic Devices
SRAM	**:** Static Random Access Memory
FGMOS	**:** Floating Gate MOS
FPGA	**:** Field Programmable Gate Array
CPLD	**:** Complex Programmable Logic Devices
CLB	**:** Configurable Logic Block
LUT	**:** Look Up Table
SOF	**:** Start Of Frame
EOF	**:** End Of Frame
ASIC	**:** Application Specific Integrated Circuit

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

FPGA ÜZERİNDE MAC/IP/UDP PROTOKOLÜNÜN GERÇEKLENMESİ

ÖZET

Bilgisayarlar çok güçlü ve yetenekli cihazlar olmalarına rağmen, birbirleri ile haberleşemediklerinde yetenekleri önemli ölçüde sınırlandırılmış olur. Veri ve kaynak paylaşımı amacı ile bilgisayarlar arasında bir ağ kurulduğunda, kullanıcıya daha etkin ve güçlü bir sistem sağlamaktadırlar. Ağ bir bina içindeki bilgisayarlar arasında kurulabileceği gibi, kıtalar arası birçok bilgisayar arasında da kurulabilmektedir.

Bilgisayar ağlarının ilk yıllarında firmalar bilgisayarları haberleştirebilmek için kendi donanın ve yazılımlarını satıyorlardı. Bu sistemler kendi içinde uyumlu çalışmasına rağmen farklı ağ sistemleri donanım ve yazılım farklılıklarından dolayı düzgün haberleşemiyordu. Bu durum ağ tabanlı yazılım geliştirmek isteyen kişiler için de zorluk yaratıyordu. Bu amaçla 1970'lerin sonlarında ve 1980'lerin başlarında ISO'nun TC 97 (Technical Committee 97), Enformasyon İşleme'si tarafından Açık Sistem Bağlantısı OSI modeli (Open System Interconnect) geliştirilmeye başlanmış, son OSI modeli 1984'te çıkartılmıştır.

Uygulamaya özel tümdevre tasarlamanın maliyet, süre ve zorluk gibi dezavantajları olduğundan, tasarımcılar yeniden konfigüre edilebilir yapılara ihtiyaç duymuşlardır. Önceleri PLA, PAL gibi basit PLD'ler kullanılsa da kapasite, hız ve yetenek ihtiyaçları arttıkça, CPLD ve sonra FPGA gibi karmaşık yapılar geliştirilmiştir. CPLD'ler ayrık halde bulunan PLD'lerin performansını iyileştirmek, kapladığı alanı azaltmak ve güvenilirliği arttırmak amacı ile bir araya getirilmesi ile oluşturulmuştur.

Daha sonra CPLD'lerden daha gelişmiş yapıya sahip olan FPGA'lar geliştirilmiştir. Tasarımcılar için esnek bir platform sunan FPGA'lar günümüzde çok popüler hale gelmiştir. FPGA'ların en önemli uygulama alanları arasında savunma ve uzay sanayi, sayısal işaret ve görüntü işleme, telekomünikasyon, tıbbi görüntüleme ve otomotiv sanayi sayılabilir.

Bu tez çalışmasında, FPGA'lerin ethernet arayüzünü destekleyen cihazlarla haberleşebilmesi amacı ile ethernet MAC, IP ve UDP katmanlarının VHDL dili ile gerçeklenmesi yapılmıştır. Bu amaçla, Avnet firmasının tasarladığı, Xilinx firmasının Spartan 6 ailesinden XC6SLX9 tipi FPGA içeren LX9 Microboard kullanılmıştır.

Öncelikle MAC, IP ve UDP katmanları için, hem gönderim hem alım işlemini sağlayan VHDL modülleri yazılmıştır. Ücretsiz olan Xilinx ISE 14.1 Webpack sürümü üzerinde yazılan VHDL kodlarının, Xilinx ISIM simülatör yazılımı ile davranışsal benzetimi yapılmıştır. Son olarak Xilinx ISE 14.1 ile proje gerçeklenmiş, Xilinx Impact kullanılarak FPGA'ya gömülmüştür. Ayrıca gerçek devre ile test edebilmek amacıyla Uart modülü entegre edilmiştir. Böylece test için bilgisayardan gönderilen ethernet paketleri Uart üzerinden alınabilmiş, Uart üzerinden gönderilen veriler de bilgisayarda ethernet üzerinden alınabilmiştir.

Sistemi test edebilmek amacı ile Ethernet arayüzünden ve Uart arayüzünden 20ms aralıklarla istenilen sayıda paket atıp, gelen paketler ile karşılaştıran bir C# uygulaması yazılmıştır. Yapılan testlerde 10000 paket atılmış ve hiç paket kaybı yaşanmamıştır.

Örnek bir uygulama olması açısında, ev güvenlik uygulaması geliştirilmiştir. Bu amaçla bir adet hareket sensörü ve bir adet darbe sensörü FPGA kartına bir ara kart ile bağlanmış, sensörlerden gelen alarmları kontrol edip bilgisayara uygun UDP paketleri atan modül yazılmıştır. FPGA kartı ile haberleşip sensörlerin durumlarını gösteren bir C# uygulaması yazılmıştır.

Tez çalışmasında öncelikle, OSI katman yapısı ve ağ topolojileri hakkında bilgi verilmiştir. Daha sonra sırası ile ethernet MAC, IP ve UDP katmanları hakkında bilgi verilmiş ve paket yapıları detaylı bir şekilde anlatılmıştır. Arkasından yeniden programlanabilir yapılar ve FPGA hakkında detaylı bilgi verilmiştir. Son olarak bütün VHDL modülleri tek tek açıklanmış, basitleştirilmiş durum diyagramları verilmiştir.

IMPLEMENTATION OF MAC/IP/UDP PROTOCOL ON FPGA

SUMMARY

Although the computers are powerful and talented devices, their capabilities are limited without communication with each other. A more powerful system is provided to the user when a network is established between computers to share data and resource.

In the first years of computer network, companies were developing and selling their custom hardware and software. Although those systems were able to work well, they were not compatible with network systems from different companies. For this purpose, Technical Committee 97 from ISO started to develop Open System Interconnect and released the latest version in 1984.

In order to divide the communication into layers, some rules has to be followed. First, every layers must have a well-defined function. While defining functions for layers, international communication standards must be considered. Layers borders must be defined such that communication between layers must be minimum. Number of layers must be optimum such that a function must not contain two layers and system must not be cumbersome.

By dividing the communication into layers, designers may specialize on different layers. Thus different designers may work on different layers of the system at the same time. This can also simplify error detection and solution.

7 layers of OSI are physical layer, data link layer, network layer, transport layer, session layer, presentation layer and application layer. Ethernet in on data link layer, IP is on network layer and UDP is on transport layer. Physical layer of Ethernet communication is provided by a chip called Ethernet PHY.

Ethernet was developed and tested by Dr. Robert Metcalfe at Palo Alto research center in 1970. He could make a communication between Alto computers and laser printers with 2.94 Mbps using a coaxial cable. Name of ethernet was originated from ether which is believed to fill the universe. Ethernet was used in Xerox until 1970 and was released in 1980 officially. Xerox, DEC and Intel published DIX Standart that has 10Mbps speed over coaxial cable.

While developing Ethernet, three problems had to be solved such as how the data is sent over cable, how the sender and receiver are detected and which node uses the cable for half duplex lines. Data is sent in small packets so a computers does not occupy the line for huge data transfer and retransmission can be done for each packet in case of an error. Each computer has a 48-bit unique MAC address and an Ethernet packet contains both sender MAC address and receiver MAC address. Also carrier sense multiple access with collision detection method was developed to detect the node, which will use the line.

Internet protocol is on the third layer of the OSI architecture. Purpose of the IP layer is to provide a connectionless communication between two computers through

several IP networks by using only IP addresses. IP is a connectionless protocol, it does not guarantee delivery of the packets. Thus it is possible that IP packets do not arrive, arrive twice, corrupt or arrive in different orders. TCP, which is on the forth layer of OSI, guarantees the delivery of the packet.

Every computer on internet has a unique 32-bit IP address for IPv4. Since 4294967296 IP addresses for IPv4 are not sufficient nowadays, IPv6 has been developed which has 128-bit IP address.

When a computer wishes to send data over IP, it needs to know the IP and MAC addresses of the destination computer. In a case such that IP address is known but MAC address is not, computer sends an ARP request, and the receiver computer replies the ARP request with its own MAC address inside. Therefore, sender computer learns the MAC address and stores it to ARP table.

UDP is on the forth layer of OSI architecture. Purpose of UDP layer is to provide a connectionless communication between two software threads on a computer. UDP is a connectionless protocol, it does not guarantee delivery of the packets. UDP contains error detection but does not have error correction mechanism like TCP. The reasons why UDP is used instead of TCP are small header size, fast processing of packet and being appropriate for time critical application like voice over IP.

Although there are analog implementations, today computers, communication systems, signal and image processing systems are implemented as digital. First digital electronics applications were developed after the invention of magnetically controllable relays but digital computation age started after the invention of vacuum tubes.

First computer ENIAC, which used vacuum tubes, was designed in 1945 to calculate the trajectory of field gun of the American army. Since power consumption was too high, there were thermal cooling problems. It was 25m long, 3m tall and had 18000 tubes. After the invention of bipolar transistors at Bell laboratories in 1947, transistors replaced the vacuum tubes and digital and analog designs had a breakthrough.

Due to the cost, time and difficulty disadvantages of ASIC design, designers needed reconfigurable structures. Reconfigurable devices include reconfigurable blocks and configurable interconnects. In the first years of reconfigurable devices, simple structures like PLA and PAL were used, however, as the need for capacity, speed and capabilities increased, complex structures like CPLD and FPGA were developed. CPLDs are formed by combining discrete PLDs to improve the performance, reduce the total area and improve the reliability.

Later FPGAs, which have more complex structure, were developed. FPGAs are very popular today due to serving a flexible platform for designers. Codes written with VHDL or Verilog languages are converted to digital circuit by special software. FPGAs use different technologies such as anti-fuse, eeprom and sram. Anti-fuse technology allows programming only once. Eeprom technology allows programming multiple times but programming is slow and due to eeprom structure, it is difficult to manufacture. Sram structure is fast and easy to manufacture but it requires external non-volatile memory.

Basic building blocks of FPGA are configurable logic blocks, configurable interconnects and input output blocks. FPGAs may also include DSP blocks, digital clock managers, memory, Ethernet MAC, PCI controller and embedded

microprocessors. Most important application fields of FPGAs are defense and space industry, digital signal and image processing, telecommunication, medical imaging and automotive industry.

In this thesis, ethernet MAC, IP and UDP modules are implemented in VHDL for FPGAs to communicate with devices supporting ethernet interface. For this purpose LX9 Microboard, which is designed by Avnet containing Spartan 6 family XC6SLX9 FPGA, is used.

Firstly, VHDL modules of MAC, IP and UDP are implemented for both transmit and receive. VHDL modules, which are written on Xilinx ISE 14.1 Webpack free version, are simulated on Xilinx ISIM Simulator tool. Finally project is implemented on Xilinx ISE 14.1 and downloaded to FPGA by Xilinx Impact. An Uart module is integrated to project for test purposes. Ethernet packets sent from computer can be received from uart port on computer, uart packets sent from computer can be received from ethernet port on computer. Docklight terminal software is used to send and receive the uart packets. Colasoft Packet Builder software is used to generate UDP packets. Wireshark software is used to observe the UDP packets received by the computer.

In order to test the system, a C# application is developed which sends Ethernet and uart packets every 20ms. Application receives and compares the packets then increments the counter of each packet type. For test, 10000 packets were sent and packet loss was not seen.

Home security application is developed to show the possible application areas of the core. A printed circuit board was designed which interfaces a PIR sensor and a vibration sensor with FPGA board. A VHDL module that sends packets to computer in case of an alarm replaced the uart module. A C# application was developed which communicates with the FPGA board and shows the status and time of the individual sensors.

Finally, resource the design uses and timing values are given. Core is compared with other works in the literature in terms of resource and timing performance. Then future works that can be done are given.

In the thesis, firstly information about OSI model and network topologies are supplied. Later, information about ethernet MAC, IP and UDP is supplied and detailed packet structures are deeply analyzed. Information about reconfigurable structures and FPGA are supplied. Each VHDL module is explained individually and simplified state diagrams are given. Functional and performance test procedures are explained. An example application is shown and comparison with other works is done.

1. GİRİŞ

Bilgisayarlar matematiksel ve mantıksal işlem yapabilme açısından tek başlarına çok güçlü cihazlardır. Bir ağ ile birbirlerine bağlandıklarında, her bir bilgisayarın sağladığı fonksiyonlar ve farklı özellikler diğer bilgisayarlar tarafından da kullanılacağı için, bilgisayarlar daha da güçlü hale gelmektedirler. Ağ yapısının oluşturulma sebebi temel olarak kaynak ve veri paylaşımı ile bilgisayarların daha etkin ve daha güçlü bir şekilde çalışmalarına olanak sağlamaktır. Ağ yapısı, aralarında veri paylaşımı yapan birkaç bilgisayardan oluşabileceği gibi geniş coğrafi bölgelerde de oluşturulabilmektedir.

Bugünün iş dünyasında, verimli ve güvenilir veri erişimi rekabetçi bir avantaj sağlamaktadır. Veri depolamada bilgisayarlar dijital olarak dosya ve kağıtların yerini almışlardır. İş arkadaşları binlerce kilometre ötede bile olsa anında verileri paylaşabilmekte, aynı yerde yüzlerce kişi eşzamanlı olarak veriler arasında araştırma yapabilmektedir, aynı veri tabanına erişerek farklı işlemler yürütebilmektedir.

Kişisel bilgisayarların 80'li yılların başlarından itibaren yaygınlaşması ile bilgisayarlar arasında hızlı ve güvenli veri aktarımına olan ihtiyaç artmış, çözüm olarak ise Ethernet geliştirilmiştir.

Günümüzde Token Ring, FDDI, ARCNET gibi birçok ağ teknolojisinden söz edilse de, teknolojisi ve üretim haklarıyla herkese açık olması, kullandığı teknolojinin üretimi kolay ve ucuz olması, güvenilir ve kullanıcıların ihtiyaçlarını karşılayabilmesi gibi özelliklerinden dolayı Ethernet dünya üzerindeki en yaygın ağ teknolojisi olma özelliğini elinde bulundurmaktadır. En yaygın ağ teknolojisi olması Ethernet'in üreticiler için büyük bir pazar haline gelmesine ve sürekli geliştirilmesine olanak sağlamaktadır.

1.1 Tezin Amacı

Bu çalışmada, ethernet bağlantısını sağlayan cihazlar ile FPGA'nın haberleşmesini sağlayan bir yapı oluşturmak hedeflenmiştir. Bu amaçla, OSI mimarisindeki ilk 4

katman olan fiziksel katman, veri bağlantı katmanı, ağ katmanı ve taşıma katmanını gerçekleyen FPGA uygulaması tasarlanmıştır.

2. OSI KATMAN YAPISI VE AĞ TEMELLERİ

Bilgisayar ağlarının ilk günlerinde, firmalar bilgisayarları haberleştirmek için kendi donanım ve yazılımlarını geliştiriyorlar ve satıyorlardı. Bu ağ sistemleri kendi içinde uyumlu çalışıyor, fakat farklı ağ sistemleri kendi aralarında donanım ve yazılım farklılıklarından dolayı haberleşemiyorlardı. Bu durum ağ tabanlı yazılım geliştirmek isteyen kişiler için de zorluk yaratıyordu. Bu amaçla 1970'lerin sonlarında ve 1980'lerin başlarında ISO'nun TC 97 (Technical Committee 97), Enformasyon İşleme'si tarafından Açık Sistem Bağlantısı OSI modeli (Open System Interconnect) geliştirilmeye başlanmış, son OSI modeli 1984'te çıkartılmıştır.

OSI, bilgisayarda çalışan bir uygulamanın, diğer bir bilgisayarda çalışan başka bir uygulama ile nasıl bir mimari üzerinden haberleşeceğini tanımlamaktadır[1]. OSI modeli 7 katmandan oluşmaktadır ve bu katmanların oluşmasında uygulanan prensipler:

- Her farklı bir seviyedeki ayrım için bir katman oluşturulmalıdır.
- Her katmanın iyi tanımlanmış bir fonksiyonu olmalıdır.
- Her katmanın fonksiyonu oluşturulurken, uluslararası standartlaşmış protokoller göz önünde bulundurulmalıdır.
- Katman sınırları, katmanlar arasındaki haberleşme minimum olacak şekilde belirlenmelidir.
- Katman sayısı, bir fonksiyon iki katmanı kapsamayacak kadar fazla, sistemi hantallaştırmayacak kadar az olmalıdır.

Haberleşmenin katmanlara ayrılması, karmaşıklığı azaltarak tasarımcıların belli katmanların işlevlerinde uzmanlaşmasını sağlamaktadır[2]. Böylelikle farklı uzmanlıktaki kişiler aynı anda sistemin farklı katmanları üzerinde çalışabilmektedirler. Bir katmandaki değişiklik diğer katmanı etkilemeyeceğinden sorun tespiti ve çözümü kolaylaşmaktadır.

2.1 OSI Katman Yapısı

OSI katmanının 7 bileşeni aşağıda anlatılmaktadır.

2.1.1 Fiziksel katman

Verinin fiziksel ortamda iletilebilmesi için gerekli kuralları belirlemektedir. Bakır, fiber optik veya radyo frekansı gibi iletim ortamları, gerilim seviyeleri, empedans değeri, kablo özellikleri, konnektör özellikleri, iletişim frekansı ve modülasyon tekniği gibi özellikler fiziksel katmanda tanımlanmaktadır. Hub'lar bu katmanda çalışmaktadır. Hub'lar herhangi bir portundan gelen paketi diğer bütün portlarına göndermekte, paketi bütün bilgisayarlar almakta, adreslenen bilgisayar paketi işlemekte ve diğerleri paketi yok saymaktadır. Paketin içeriğini bilmemektedir. Hub'lar gelen paketleri her porta gönderdiğinden gereksiz ağ trafiği yaratmakta ve bundan dolayı pek kullanılmamaktadır.

2.1.2 Veri bağlantı katmanı

Bu katman, veri dizilerine başlık bilgileri ekleyerek paket oluşturma ve paket açma, alınan paketlerde hata oluşup oluşmadığının kontrolü, hata oluştu işe paketin tekrar gönderilmesi gibi işlevlerden sorumludur. Gönderen taraf pakete gönderen adresi, alıcı adresi ve hata kontrolü için CRC (Cyclic Redundancy Check) bilgisini koymaktadır. Alıcı taraf ise alıcı adresi kendine ait ise paketi almakta, hata denetimini yapmakta ve paket hatalı ise tekrar gönderilmesini sağlamaktadır. Anahtarlar (Switch) bu katmanda çalışmaktadır. Anahtarlar herhangi bir portundan gelen paketin alıcı adresine baktıktan sonra, ilgili adrese sahip bilgisayarın bağlandığı porta paketi aktarmaktadır. Portlarına bağlı bilgisayarların adreslerini belleğinde tutarak, yönlendirme işlemini yapmaktadır.

2.1.3 Ağ katmanı

Bir bilgisayar, bulunduğu yerel ağdan başka bir ağda bulunan bilgisayara veri göndermek istediğinde, ağ katmanı başlıkları eklenerek gönderilir. Veri paketlerinin bir uçtan diğer uca ağdaki çeşitli düğümlerden geçirilip, yönlendirilerek alıcısına ulaşmasını sağlayan bir işleve sahiptir. Yönlendiriciler (Router) bu katmanda çalışmaktadır. İki farklı yerel ağ kendi içlerinde hub veya anahtar ile

haberleşebilirken, farklı ağlardaki iki bilgisayarın haberleşebilmesi için yönlendirici gerekmektedir.

2.1.4 Taşıma katmanı

Bu katman, verinin uçtan uca hatasız gönderilmesinden sorumludur. Bir üst katmandan gelen verileri, daha küçük paketlere bölerek göndermektedir. Her bir pakete numara vererek, paketlerin alıcı tarafında doğru sıralanmasını sağlamaktadır. Verinin karşı tarafa gönderilip gönderilmediği bu katmanda kontrol edilmektedir.

2.1.5 Oturum katmanı

Bilgisayarlar arasındaki gerekli oturumun kurulması, yönetilmesi ve sonlandırılması işlemlerini içermektedir. Birden fazla bilgisayarla iletişim halinde olunması durumunda doğru bilgisayarla haberleşmeyi sağlamaktadır.

2.1.6 Sunum katmanı

Sunum katmanının en önemli görevi yollanan verinin karşı bilgisayar tarafından anlaşılabilir halde olmasını sağlamaktır. Böylece faklı programların birbirlerinin verisini kullanabilmesi mümkün olabilmektedir. Şifreleme/şifre çözme, sıkıştırma/açma gibi işlemler bu katmanda gerçekleştirilmektedir.

2.1.7 Uygulama katmanı

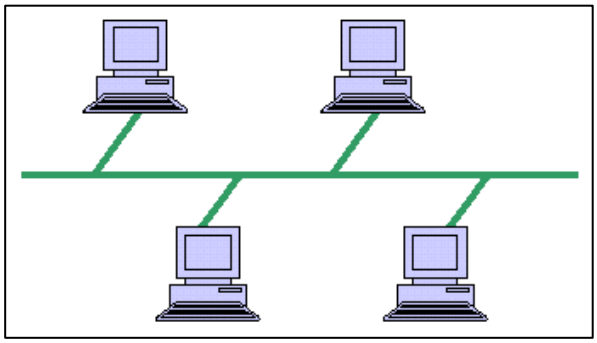
Kullanıcıya en yakın katman olup, programların ağı kullanabilmesi için araçlar sunmaktadır. Ağ araçlarına erişmek isteyen programlayıcı, alt tarafta nelerin olduğundan haberi olmadan uygulamasını geliştirebilmektedir.

2.2 Ağ Topolojileri

Fiziksel ağ topolojisi bilgisayarların, bilgisayarları bağlayan kabloların ve farklı ağ aygıtlarının bağlantı şekillerini belirtmektedir[3]. İki ağın topolojisi aynı olsa bile birimler arasındaki mesafe, fiziksel bağlantı, iletim hızları farklılık gösterebilmektedir. Ağ topolojileri temelde yol, halka, yıldız, ağaç ve örgü olmak üzere 5 ana gruba ayrılmaktadır.

2.2.1 Yol topolojisi

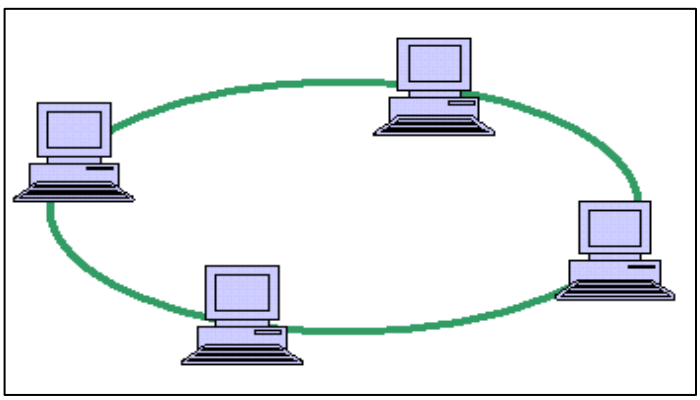
Bir ana hat ve buna bağlanmış uç birimlerden oluşmaktadır. Ana kabloya bağlı bir bilgisayar diğer bilgisayara paket göndermek istediğinde, paket bütün bilgisayarlar tarafından alınmakta, sadece adreslenen bilgisayar paketi işleme almaktadır. Bu topolojinin kurulumu ve kablolaması düşük maliyetlidir. Ağa katılabilecek bilgisayar sayısı sınırlı olup ana kabloda arıza meydana geldiğinde bütün ağ çalışmaz hale gelmektedir. Şekil 2.1'de gösterilmiştir.



Şekil 2.1 : Yol Topolojisi.

2.2.2 Halka topolojisi

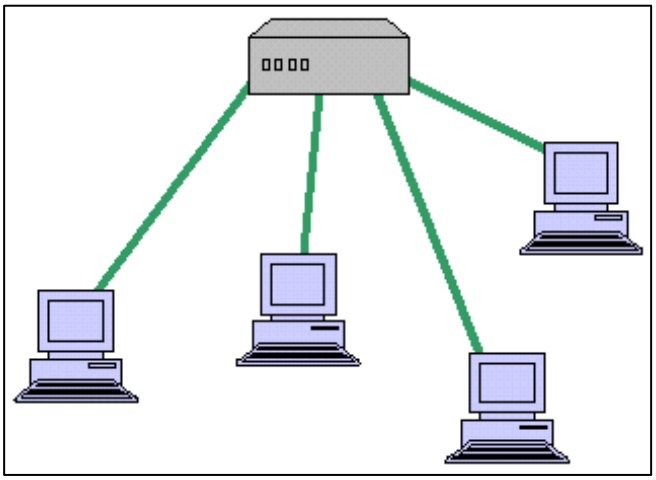
Halka şeklinde birbirine bağlanmış bilgisayarlardan oluşmaktadır. Her bir bilgisayarın 2 adet komşusu bulunmaktadır. Bir bilgisayar paket göndermek istediğinde, paketi sağındaki veya solundaki bilgisayara iletmektedir. Paketi alan bilgisayar paketin içeriğine bakıp, kendisine gönderildiyse paketi işleme koymakta, gönderilmediyse yanındaki bilgisayara iletmektedir. Bu şekilde paket sonunda adreslenen bilgisayara ulaşmaktadır. Bir bilgisayarda arız meydana geldiğinde ağ etkilenmektedir. Şekil 2.2'de gösterilmiştir.



Şekil 2.2 : Halka Topolojisi.

2.2.3 Yıldız topolojisi

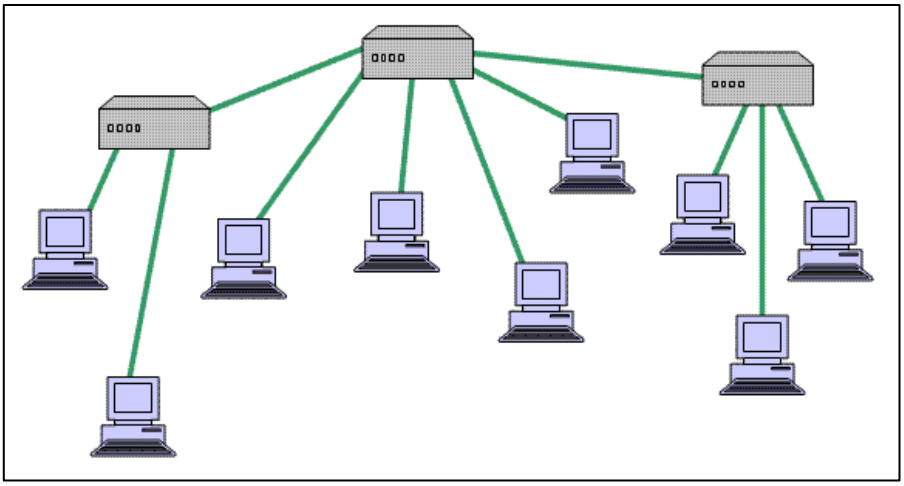
Yıldız topolojide, ortada bilgisayarların bağlı bulunduğu hub, switch veya router gibi bir cihaz bulunmaktadır. Bir bilgisayar paket göndermek istediğinde, paketi ortada bulunan cihaza göndermekte, cihaz da paketi ilgili cihaza göndermektedir. Bu topolojinin kurulumu ve kablolaması ilk ikisine göre yüksek maliyetlidir. Bir bilgisayarda arıza meydana geldiğinde ağın geri kalanı çalışmaya devam edebilmektedir. Şekil 2.3'de gösterilmiştir.



Şekil 2.3 : Yıldız Topolojisi.

2.2.4 Ağaç topolojisi

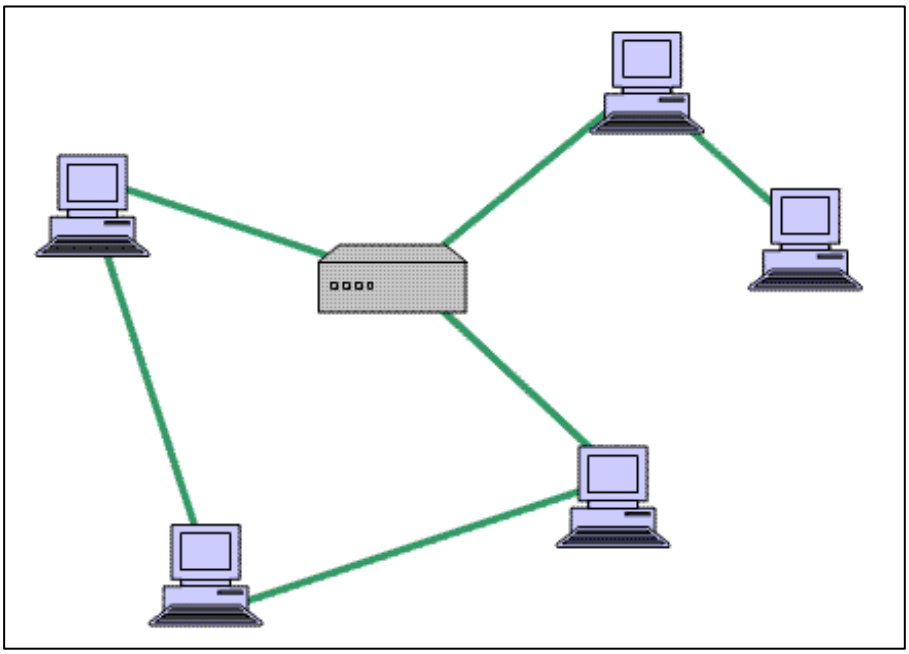
Yol ve yıldız topoloji birleştirilerek oluşmuştur. Birden fazla yıldız ağ yapısı bir yol üzerinde birleştirilmiştir. Segmentlerden birinde sorun çıktığında diğer segment çalışmaya devam edebilmektedir. Şekil 2.4'de gösterilmiştir.



Şekil 2.4 : Ağaç Topolojisi.

2.2.5 Örgü topolojisi

Örgü topolojide bir bilgisayar ağdaki bütün bilgisayarlara direkt veya dolaydı yoldan bağlıdır. Her bir bilgisayar hem kendi mesajını hem de başka bilgisayarların mesajlarını iletmekten sorumludur. Her bir mesaj farklı yollardan alıcısına ulaşabilmektedir. Kurulumu pahalı olmakla birlikte, genelde kablosuz ağlarda kullanılmaktadır. Şekil 2.5'de gösterilmiştir.



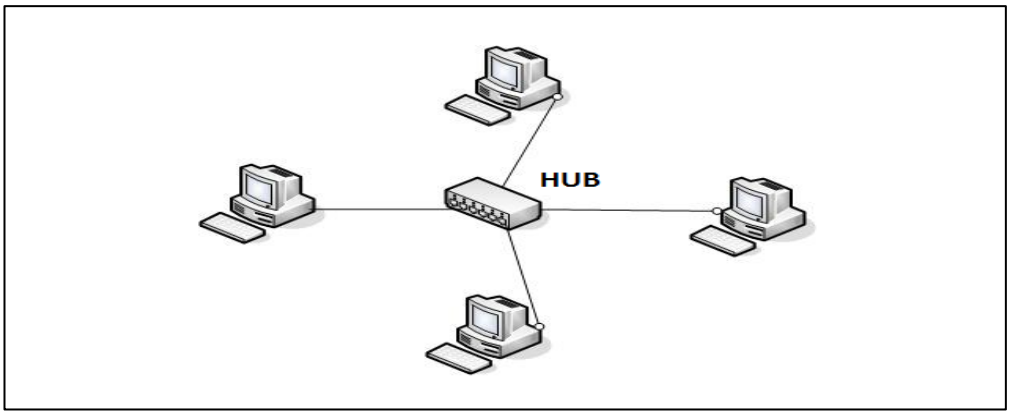
Şekil 2.5 : Örgü Topolojisi.

2.3 Ağ Cihazları

Farklı ağ yapılarında kullanılan cihazlar aşağıda anlatılmaktadır.

2.3.1 Hub

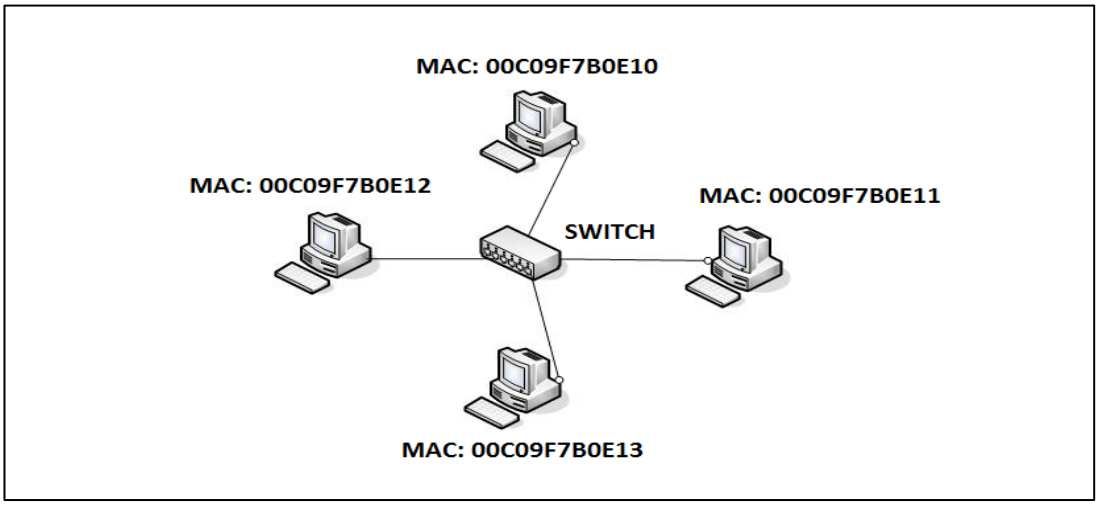
Şekil 2.6'da gösterildiği gibi HUB'lar yıldız topolojide bilgisayarları birbirine bağlamak için kullanılmaktadır. HUB'lar akıllı cihazlar değildir, hangi bilgisayarların kendisine bağlı olduğunu bilmez. Bir bilgisayar ağdaki başka bilgisayara paket göndermek istediğinde paketi HUB'a gönderir, HUB bu paketi kendisine bağlı bütün bilgisayarlara gönderir, ilgili bilgisayar paketi işler, diğerleri de paketi çöpe atar. Bu işlem ağdaki trafiği artırdığı için genelde basit ağlar kurulacağı zaman kullanılmaktadırlar. OSI mimarisinin 1. katmanı olan fiziksel katmanda çalışmaktadır.



Şekil 2.6 : HUB Yapısı.

2.3.2 Switch

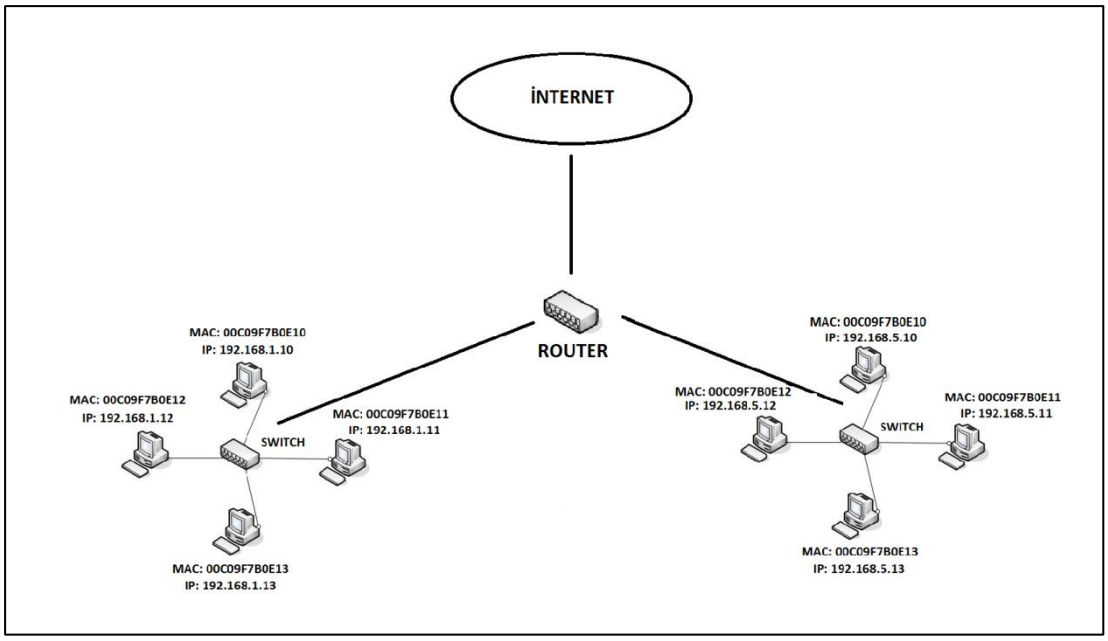
Switch yıldız topolojide bilgisayarları birbirine bağlamak için kullanılmaktadır. Switchler HUB'ların tersine akıllı cihazlardır. Kendine bağlı bulunun bütün cihazların MAC adreslerini ve hangi portuna bağlı olduğu bilgisini hafızasında tutabilmektedir. Bir bilgisayar ağdaki başka bilgisayara paket göndermek istediğinde paketi switche gönderir, switch bu paketin alıcı MAC adresine bakar, tablosundaki MAC adresleri ile karşılaştırır, herhangi bir portundaki bilgisayarın MAC adresi ile eşleştirirse paketi o portuna iletir, eşleştiremezse paketin geldiği port hariç diğer bütün portlara gönderir. Bir bilgisayar switche ilk kez bağlandığında, ARP paketi atar ve switche kendini tanıtır. Switch bilgisayarın MAC adresini ve port numarasını belleğinde kaydeder. İlerde o bilgisayara bir paket geldiğinde tablodan MAC adresi port numarası eşleştirmesi yapar ve paketi yönlendirir. OSI mimarisinin 2. katmanı olan veri bağlantı katmanında çalışır.



Şekil 2.7 : Switch Yapısı.

2.3.3 Router

Ağ içinde iki adet bilgisayar HUB veya switch üzerinden MAC adreslerini kullanarak haberleşebilmektedir. Fakat iki farklı ağdaki iki bilgisayar birbiriyle haberleşememektedir. Haberleşmenin gerçekleşmesi için Router(Yönlendirici)'a ihtiyaç vardır. OSI mimarisinin 3. katmanı olan ağ katmanında çalışmaktadır. Bir bilgisayar farklı bir ağdaki bilgisayara paket göndermek istediğinde, pakete alıcı bilgisayarın IP adresini koyması gerekmektedir. Paketi alan router, alıcı IP adresine bakar, tablosundaki IP adresleri ile karşılaştırıp ilgili ağa yönlendirir. Bu ilişki şekil 2.8'de gösterilmiştir.

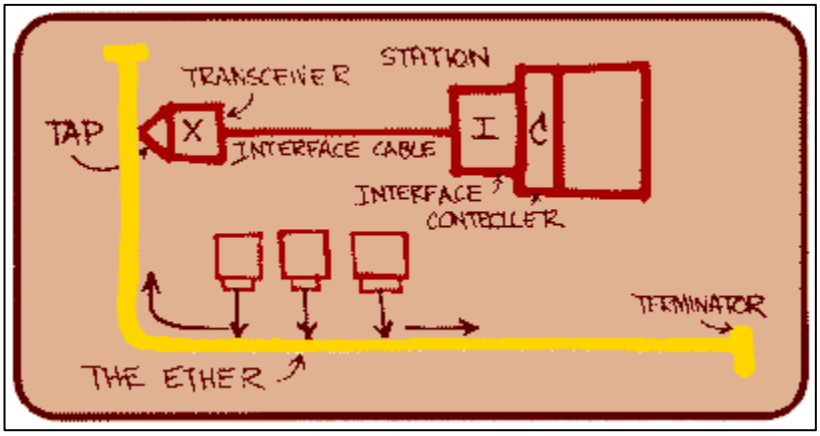


Şekil 2.8 : Router Yapısı.

3. ETHERNET

3.1 Ethernet'in Tarihi

Ethernet Xerox firmasının Palo Alto araştırma merkezinde 1970'li yıllarda Dr. Robert M. Metcalfe tarafından geliştirilmiştir ve denemeleri yapılmıştır[4]. Metcalfe, dünyanın ilk workstation bilgisayarlarından biri olan Xerox Alto bilgisayarlar ile "geleceğin ofisi" projesi üzerinde çalışmaktaydı. 1972 yılının sonlarında, Metcalfe ve Xerox'ta çalışan diğer tasarımcılar Xerox Alto bilgisayarlarını birbirine bağlamak için deneysel olarak Ethernet'i geliştirmişler ve böylece Alto bilgisayarlar diğer sunucular ve lazer yazıcılar birbiriyle haberleşebilmiştir. İlk Ethernet'in çalışma hızı Alto bilgisayarlar ile uyumlu olması için Alto bilgisayarların çalışma hızı ile aynı tutulmuş ve tek parça koaksiyel kablo kullanılan ağ 2.94 MegaBit/Saniye hızında çalışmıştır.



Şekil 3.1 : İlk Ethernet Çizimi[4].

Şekil 3.1'deki diyagram Dr. Robert M. Metcalfe tarafından 1976 yılının haziran ayında National Computer Conferance'da Ethernet'in doğuşu sırasında çizilmiştir. Ethernet'in doğuşundan beri bu diyagramdaki temellere dayanan kullanım süregelmiştir.

Metcalfe, önce Alto Aloha Network olan sistemin ismini 1973 yılında "Ethernet" olarak değiştirmiş ve böylece sistemin sadece Alto bilgisayarlarda değil tüm

bilgisayarlarda çalışabileceğini vurgulamak istemiştir. Ethernet kelimesi bir zamanlar tüm uzayı doldurduğuna ve elektromanyetik sinyallerin aktarımını sağladığına inanılan "ether" den gelmektedir. Metcalfe'nin sisteminde de veri bitleri tüm sistemlere ulaştığı için sonuçta "Ethernet" doğmuştur.

1979 yılına kadar sadece Xerox içinde kullanılan Ethernet'in resmi duyurusu 1980 yılında yapılmış, Xerox, DEC (Digital Equipment Corporation) ve Intel firmaları ile beraber, sonradan "DIX Standart" olarak anılan Ethernet standardını yayınlamıştır. DIX standardı koaksiyel kablo üzerinden 10 Mbps hızında çalışan Ethernet'i tanımlamıştır. Böylece Ethernet, firma içi deneysel bir çalışmadan herkese açık gerçek bir ürün haline gelmiştir.

DIX standardından sonra Ethernet, Institute of Electrical and Electronics Engineers (IEEE)'in 802 kodlu komisyonu tarafından geliştirilmeye devam etmiştir. IEEE 1985 yılında "IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications" şeklinde bir isimle yeni ethernet standardını yayınlamıştır. İzleyen dönemde IEEE standardı International Organization for Standardization (ISO) tarafından yürütülmeye devam etmiştir. ISO günümüzde bilgisayar ağları ile ilgili tüm standartları yürüten kuruluştur.

1985 yılından itibaren üretilen tüm ürünler IEEE 802.3 standardına göre üretilmiştir. Aslında bu ürünleri "IEEE 802.3 CSMA/CD" standardını kullanan ürünler olarak tanımlamak daha doğru olmakla birlikte, dünya çapında hala genel olarak "Ethernet" kelimesi tüm bu ürünler ve dahil oldukları teknolojiyi tanımlamak için kullanılmaktadır.

Ethernet tek bir ağ teknolojisi olmaktan çok, aynı yol topolojisini, paket yapısını ve ağ erişimi metodunu kullanan ağ teknolojileri ailesini tanımlamaktadır.

3.2 Ethernet'in Çalışma Şekli

Ethernet geliştirilirken üç ana probleme çözüm aranmıştır:

- Kablo üzerinden verinin gönderilme şekli
- Gönderen ve alıcı bilgisayarların tespit edilme şekli
- Belli bir anda kabloyu kullanacak cihazın karar verilme şekli

3.2.1 Paketler

Tüm bilgisayar ağları, ağ üzerinden aktarılacak veriyi sabit boyuttaki küçük paketler halinde iletmektedirler. Bu yöntemin iki önemli faydası bulunmaktadır. Birincisi, büyük bir dosya transferi yapan bir bilgisayar tarafından ağ uzun bir süre meşgul durumda tutulmamış olmaktadır. Bir sistemde veri paketler halinde yollanırken, her paket gönderilmeden önce kablonun kullanımda olup olmadığı kontrol edilmekte, eğer kullanımda değilse paket gönderilmekte, kullanımdaysa rasgele bir süre beklenmekte ve tekrar denenmektedir. Belli bir denemenin üzerinde hat hala meşgulse paket iptal edilmektedir. Paket karşıya ulaştığında, kablo tekrar ağdaki tüm bilgisayarlar için boş duruma gelmiş olmaktadır. Az önceki bilgisayar ikinci paketi yollamadan önce tekrar kabloyu kontrol etmek zorundadır. Bu arada diğer bir sistem kendi paketini yollayabilmektedir. Paketler küçük yapıda olduğu için saniyeler içinde yüzlercesi değişik bilgisayarlar tarafından yollanıp alınabilmektedir. Böylece ağı kullanan kullanıcılar için bir süreklilik sağlanmış olmaktadır. Veri bir bütün olarak gönderilmesi halinde, büyük boyutlu veri transferlerinde hattı tek bir kullanıcı meşgul etmekte ve diğer kullanıcılar uzun bir süre beklemek zorunda kalmaktadır.

Verileri paketler halinde göndermenin diğer avantajı ise hatasız veri transferini gerçekleştirmek amaçlıdır. Paketlere bölmeden gönderilen bir sistemde, verinin bir biti dahi bozulmaya uğrasa, bütün veri tekrar gönderilmek zorunda kalmaktadır. Oysa veri paketlere bölünüp yollandığında, sadece bozuk giden paketin tekrar yollanması yeterli olmaktadır.

Ethernet veri paketinin yapısı sabittir. Her paket şu dört bilgiyi içermektedir:

- Alıcının MAC adresi
- Gönderenin MAC adresi
- Veri
- CRC kodu

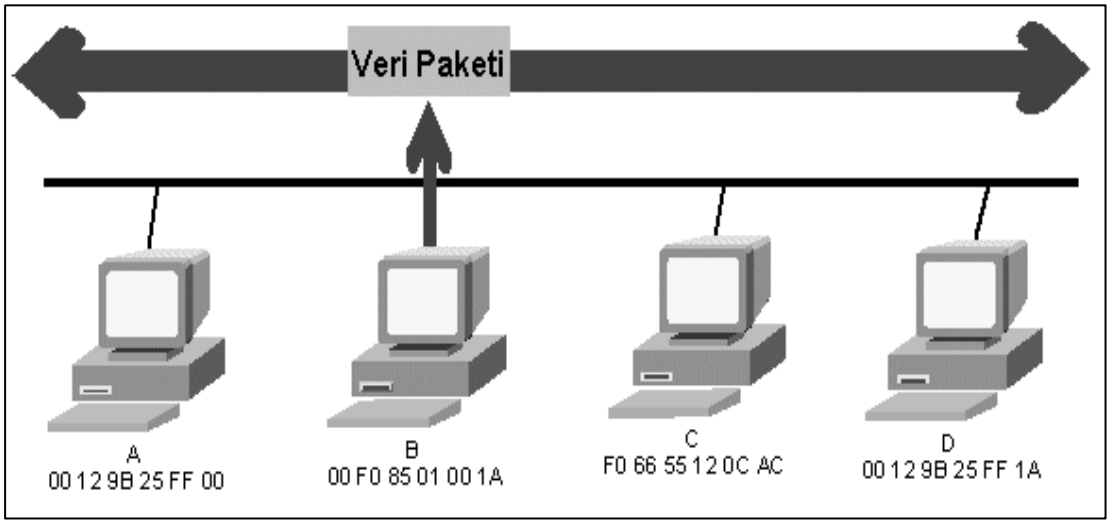
3.2.2 MAC adresi

Ethernet ağına dahil veya Ethernet ara yüzüne sahip her cihaz "node" olarak adlandırılmaktadır. Ethernet kartına sahip her bilgisayar bir node oluşturmaktadır, ancak Ethernet girişi olan başka cihazlar da olabileceği için (router'lar mesela) genel olarak ‘node’ terimi kullanılmaktadır.

Ethernet ağında sistemler birbirinden sahip oldukları MAC adresi ile ayırt edilmektedirler. Her node veya basitçe her Ethernet kartı dünyada eşi olmayan bir adrese sahiptir. Bu adres 48 bitlik bir sayıdır.

3.2.2.1 MAC adresinin kullanımı

MAC adresleri sayesinde sistemler, ağ üzerinden aldıkları veri paketinin kendilerine gönderilip gönderilmediğini tespit edebilmektedirler. Ethernet ağında, bir bilgisayar bir veri paketi yolladığında, bu paket ağdaki tüm sistemlere ulaşmaktadır. Her bilgisayar, paketin ilk bölümü olan alıcı MAC adresini okumakta ve kendi MAC adresiyle kontrol etmektedir. Eğer gelen paket ilgili bilgisayara ait ise işlenmekte değil ise işleme konulmamaktadır.



Şekil 3.2 : MAC Adresi Kullanımı.

Ethernet'in bu özelliği ciddi bir güvenlik açığına yol açabilir. Packet Sniffer olarak adlandırılan programlar, bilgisayara gelen veri paketlerini MAC adresi ne olursa olsun alıp kullanmaya izin vermektedir. Bu tip programlar iyi niyetle kullanıldığında problemlerin çözümüne yarayabileceği gibi, yerel ağdaki bir kullanıcının ağdaki paket trafiğini gözlemlemesine olanak sağlamaktadır.

3.2.2.2 Multicast ve broadcast MAC adresleri

Bir grup sistemin aynı veriyi alması isteniyorsa, bu gruba dahil olması istenen sistemlerde ethernet arayüzü, belli bir multicast adresine yollanmış veriyi kendi MAC adresine gelen bir veriyi alır gibi alması için ayarlanabilmektedir. Başka bir deyişle multicast adresler belli bir grup cihazın aynı veriyi almasını sağlamaktadır.

Broadcast MAC adresi ise 48 biti de 1 olan özel bir adrestir. Bu adrese yollanmış bir veri paketini alan her ağ kartı bu paketi kabul etmekte ve işleme koymaktadır. Ağdaki bütün bilgisayarlara gönderilmesi gereken paketlerin alıcı adreslerine Broadcast MAC adresi yazılmakta ve ağdaki tüm cihazlar bu paketi işleme koymaktadır.

3.2.3 CRC hata denetimi

Veriler iletim ortamında gönderilirken, bazı etkenlerden dolayı bozulmalara uğrayabilmektedir. Cyclic Redundancy Check veri paketlerinin ortamdaki bozulmaları durumunda bu bozulmanın tespitini sağlamaktadır.

Gönderen taraf, veri paketine konacak veriyi matematiksel bir işlemden geçirip 32 bitlik CRC kodunu oluşturmaktadır. Alıcı, Veri ve CRC kodunu aldıktan sonra aynı matematiksel işlemi veriye uygulamakta ve elde ettiği CRC kodunu gelen CRC kodu ile karşılaştırmaktadır. Eşitlik durumunda paketi işleme koymakta, eşitsizlik durumunda ise paketi iptal edip gönderici tarafa paketi tekrar göndermesi için talepte bulunmaktadır.

Bu noktaya kadar en başta karşılaşılan iki problem yani verinin nasıl gönderileceği (paketler halinde) ve ağa dahil sistemlerin nasıl birbirinden nasıl ayırt edileceği (MAC adresi ile) açıklanmıştır. Aşağıda ise kabloyu kimin kullanacağına karar veren yöntem anlatılmaktadır.

3.2.4 CSMA / CD (Carrier Sence Multiple Acces with Collision Detection)

CSMA/CD'ye göre, ethernet kartı veri gönderimine başlamadan önce kablonun kullanımda olup olmadığını kontrol etmektedir. O anda kablonun başka bir kullanıcı tarafından kullanıldığını tespit etme işlemi Carrier Sense olarak adlandırılmaktadır.

Kablo boşta olduğunda her Ethernet ara yüzüne sahip cihaz eşit hakka sahiptir ve veri aktarımına başlayabilmektedir ve Multiple Access olarak adlandırılmaktadır. Bir Ethernet ağında bilgisayar üzerinde çalışan işletim sistemi veya kullanıcısı önemli değildir. Bir DOS işletim sistemi ile çalışan bilgisayar ile Window 2000 Server ile çalışan bir bilgisayar kabloya erişim açısından eşit hakka sahiptir.

Bazı durumlarda iki sistem kablonun boş olduğunu tespit ederek aynı anda veri aktarımına başlayabilmektedir. Bu durumda iki tarafın yolladığı veri çakışmaktadır (Collision) ve Ethernet kartları çakışmayı hemen tespit edebilme yeteneğine sahiptir (Collision Detection).

CSMA/CD metodu esas olarak switch bulunmayan ve tek kablo üzerinde birden fazla Ethernet cihazının bulunduğu sistemler için geliştirilmiştir. Günümüzde switcler kullanıldığı için full-duplex bağlantılar için çakışma söz konusu değildir. Fakat half-duplex hatlarda, bir cihaz veri gönderirken veri alma durumu oluşursa çakışma olur. Bu nedenle half-duplex hatlarda CSMA / CD metodu kullanılmaktadır.

3.2.5 Çakışma durumları

Eğer birden fazla ethernet kartı aynı anda veri iletimine geçerlerse çakışma oluşmaktadır. Sistemler kendi yolladıklarıyla kablodan geleni karşılaştırarak bunu hemen tespit edebilmektedirler. Çakışmaların oluşması Ethernet'in doğasında olan bir şeydir ve her ethernet ağında çakışma olması kaçınılmazdır. Çakışma durumunda her iki taraf da özel bir algoritma ile belirlenen rastgele bir süre boyunca beklemeye başlamaktadırlar.

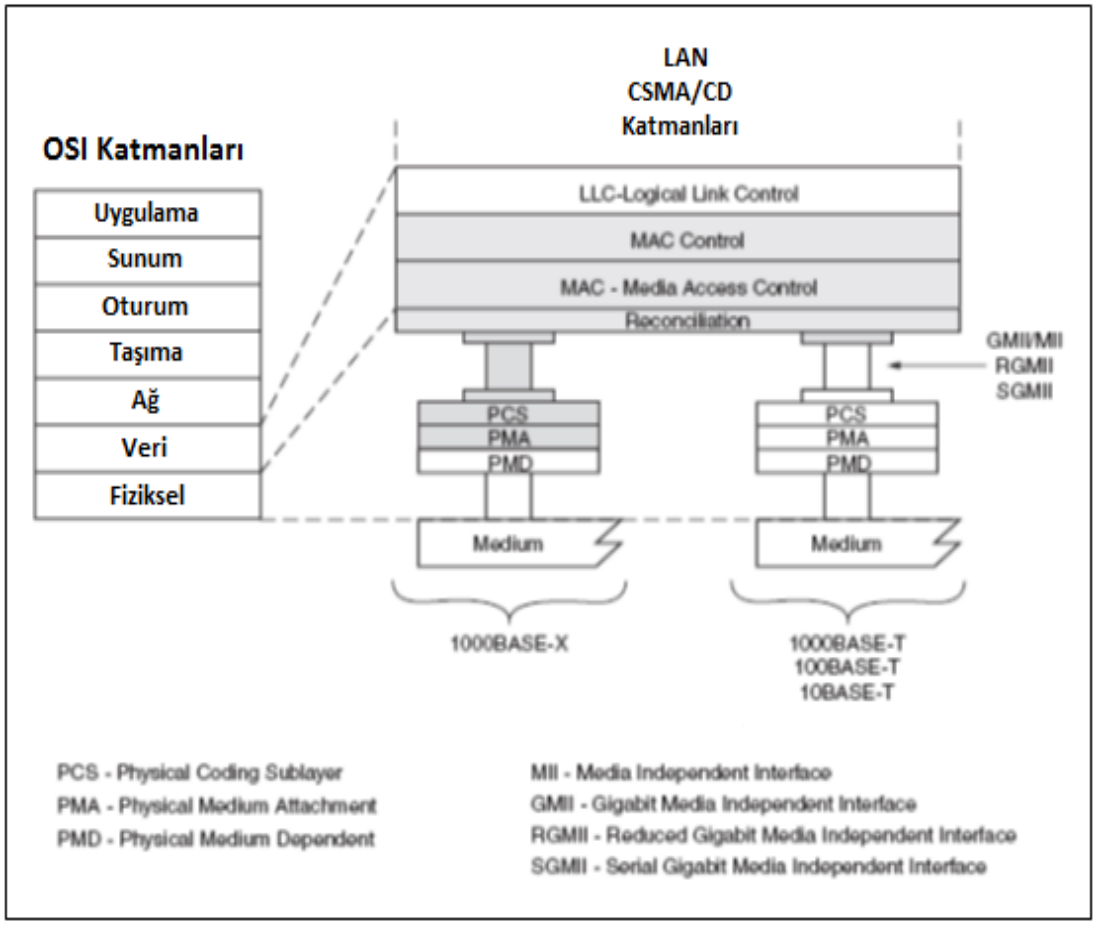
Eğer ağ çok yoğun kullanılıyorsa, aynı veri paketi gönderilirken birden fazla çakışma olabilmektedir. Bu durumda sistemler rastgele belirlenen bekleme süresini uzatmaya başlamaktadır. Ethernetin bu yapısı ağdaki trafik yoğunluğu arttıkça kendisini duruma uydurmasını sağlamaktadır. Eğer Ethernet paketi 16 denemede gönderilemiyorsa, paket iptal edilmektedir. Bu ancak çok uzun bir süre çok aşırı yoğunluk yaşanması durumunda veya kabloda meydana gelen bir arıza nedeniyle olabilir.

Bu noktada ethernetin diğer ağ teknolojilerinde de olduğu gibi veri aktarımını %100 garanti etmediğini görülmektedir. Bu açık üst katman protokollerinin sağladığı veri kontrolü ile telafi edilmektedir. Bir paket yolda kaybolursa veya 16 denemede de yollanamayıp iptal edilirse, alıcı taraftaki üst katman protokol (TCP/IP kullanılıyorsa; TCP) gönderen taraftaki TCP'ye gelen veride bir eksiklik olduğunu bildirecek ve tekrar yollanmasını isteyecektir.

Ethernetin kullandığı CSMA/CD tekniğinin basit yapısı ethernet ağ kartlarının ve diğer ekipmanların rakip teknolojilere (Token Ring) göre daha ucuza üretilebilmesini sağlamaktadır.

3.2.6 Ethernet alt katman mimarisi

Şekil 3.3'de OSI ağ modeli ile Ethernet MAC katmanının ilişkisi IEEE 802.3 spesifikasyonunda belirtildiği üzere gösterilmiştir. Gri renkli kısımlar Ethernet MAC'in görev yaptığı bölgelerdir. Şekilde ayrıca fiziksel arayüzde gösterilmiştir.



Şekil 3.3 : Ethernet Alt Katman Mimarisi[7].

3.2.6.1 MAC ve MAC kontrol alt katmanı

MAC, Ethernet paketlerinin oluşturulmasından ve hata tespit işlemlerinden sorumludur. MAC bağlandığı fiziksel katmandan bağımsızdır ve herhangi bir fiziksel katmana bağlanabilmektedir. MAC Control alt katmanı da MAC'in gerçek zamanlı akış kontrolünden sorumludur.

3.2.6.2 Fiziksel alt katmanlar

Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) ve Physical Medium Dependent (PMD) alt katmanları, Ethernet'in fiziksel katmanını oluşturmaktadır[5]. PCS, MAC katmanından çıkan verinin kodlanması (örn. 4B/5B), çoğullanması ve senkronizasyonundan sorumludur. PMA, Gönderilen verinin paralel/seri dönüşümünün yapılması ve veriden saat işaretinin elde edilmesinden sorumludur. PMD, Verinin gönderilen ortama uygun olarak elektriksel işarete dönüştürülmesinden sorumludur. Base-T ve Base-X olmak üzere iki adet ana fiziksel standart tanımlanmıştır.

Base-T fiziksel katmanı MAC ile bakır ortam arasındaki bağlantıyı sağlamaktadır. Her MAC cihazında bu fonksiyon sağlanmayabilir. Bu nedenle harici BASE-T PHY cihazları bulunmaktadır. Bu cihazlar MAC'e MII, GMII, RGMII veya SGMII ara yüzünü kullanarak bağlanabilmektedir.

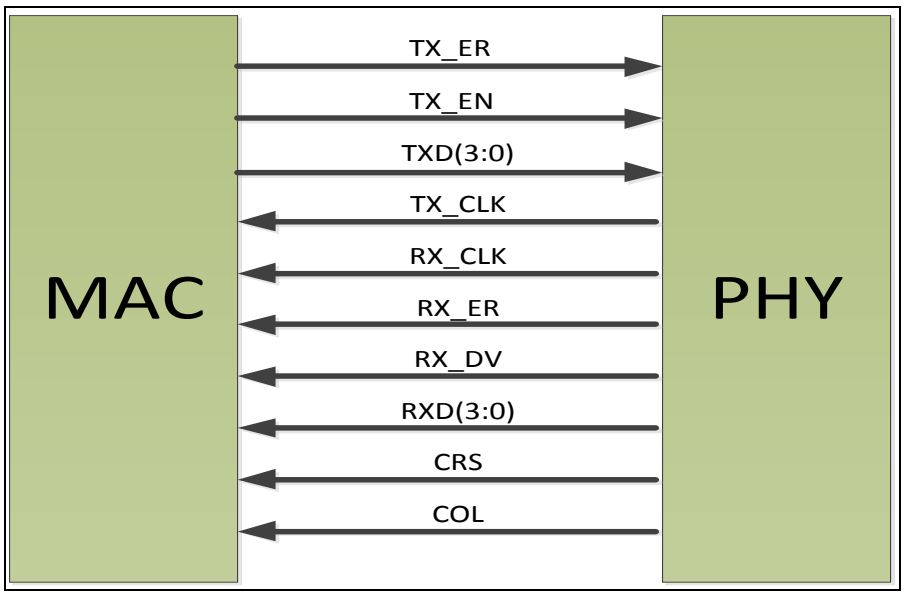
Base-X fiziksel katmanı MAC ile fiber optik ortam arasındaki bağlantıyı sağlamaktadır. Harici BASE-X PHY cihazları bulunmaktadır.

3.2.7 Fiziksel katman kontrolörü (PHY)

PHY, MAC katmanını bakır, fiber optik, RF gibi iletim ortamlarına bağlamakla görevlidir. Ethernetin fiziksel katmanını oluşturmaktadır[6]. Piyasada değişik hızları ve değişik ara yüzleri destekleyen modelleri bulunmaktadır. PHY'lar Auto-Negotiation özelliği içerebilmektedir. Auto-Negotiation özelliği, iki PHY'ın haberleşmeden önce olabilecek en iyi konfigürasyonla (Hız, duplex, akış kontrolü vs.) birbirine bağlanmasını sağlamaktadır. 10Mbps ve 100Mbps cihazlar için opsiyonel olmakla birlikte, 1Gbps cihazlar için zorunludur. PHY'lar ayrıca otomatik çaprazlama özelliği içerebilmektedir. Bu özellik ile aynı tip cihazlar birbirine bağlanmak istediğinde kabloyu çaprazlamaya gerek kalmamıştır. MAC ile PHY arasındaki haberleşme aşağıdaki ara yüz tipleri ile yapılabilmektedir[7].

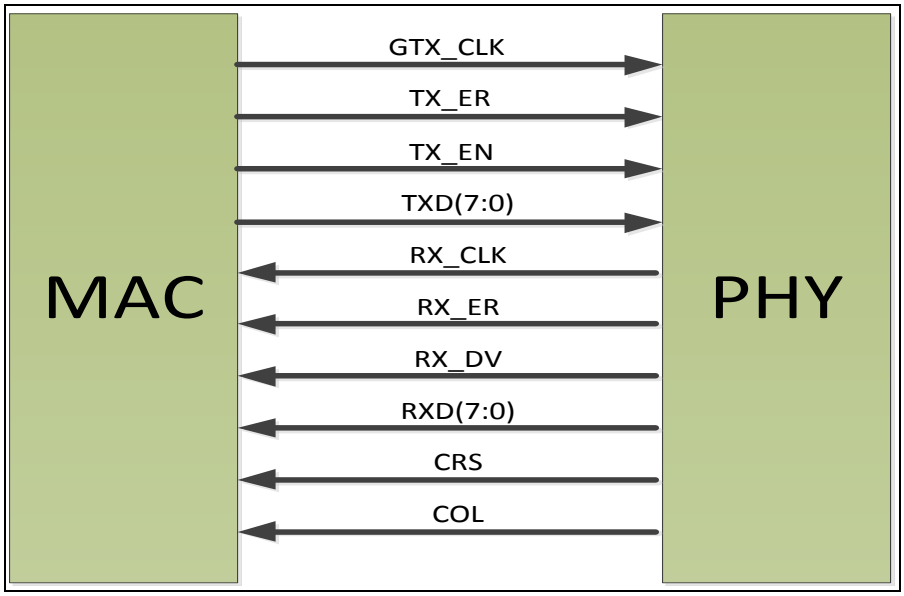
- **Media Independent Interface (MII):** Gönderim ve alım için 4-bit genişliğinde veri yolu içermektedir. MAC'e her iki saat işareti de PHY tarafından sağlanmaktadır. 10Mbps ethernet için saat frekansı 2.5 MHz, 100Mbps ethernet için saat frekansı 25MHz'dir. TX_EN ve RX_DV işaretleri

hatta veri olduğunu, TX_ER ve RX_ER işaretleri hata oluştuğunu belirtir. CRS ve COL işaretleri half-duplex iletişim için kullanılır ve çakışma durumlarını belirtmeye yararlar. Şekil 3.4'de ilgili sinyaller gösterilmiştir.



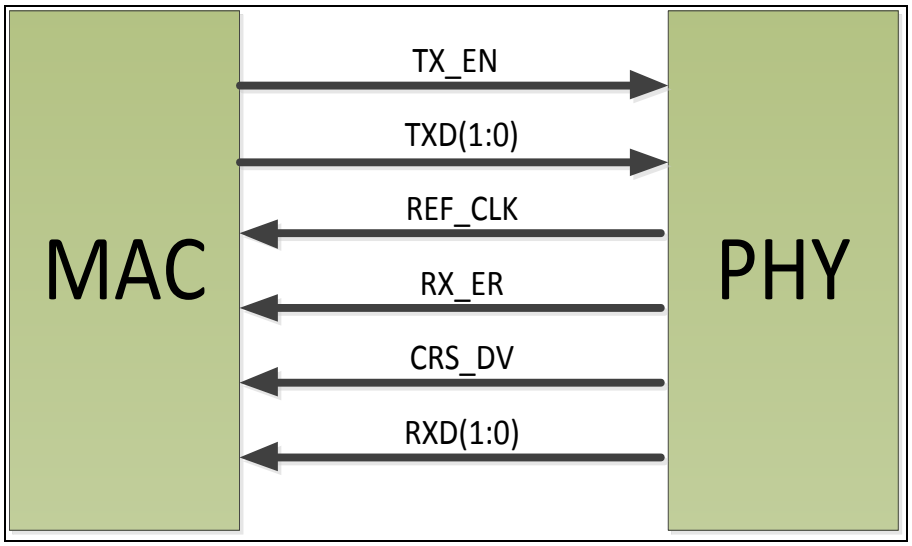
Şekil 3.4 : MII.

- **Gigabit Media Independent Interface (GMII):** Gönderim ve alım için 8-bit genişliğinde veri yolu içermektedir. Gönderim saat işareti MAC tarafından sağlanırken, alım saat işareti PHY tarafından sağlanmaktadır. 1Gbps ethernet için kullanılmaktadır ve saat frekansı 125MHz'dir. Diğer işaretler MII'dakilerle aynıdır. Şekil 3.5'de ilgili sinyaller gösterilmiştir.



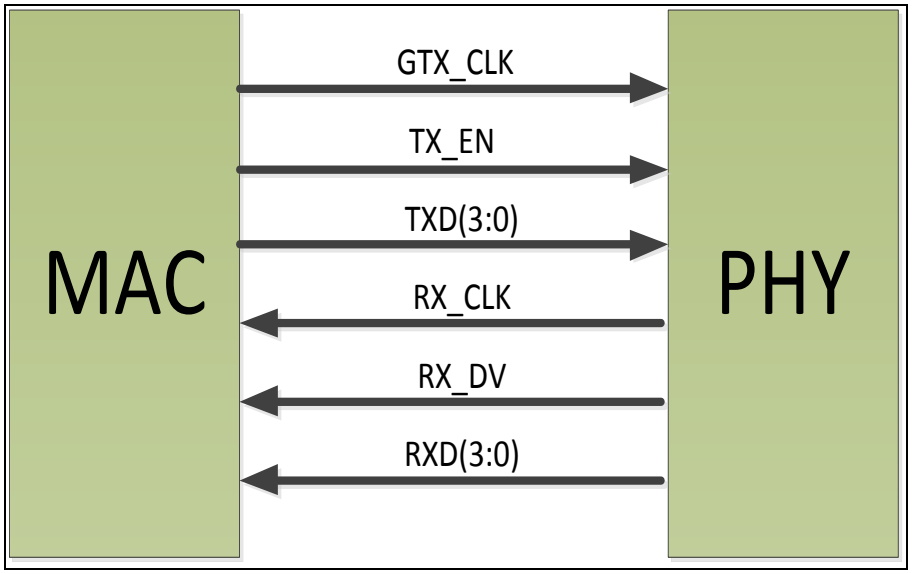
Şekil 3.5 : GMII.

- **Reduced Media Independent Interface (RMII):** Gönderim ve alım için 2-bit genişliğinde veri yolu içermektedir. MAC'e hem gönderim hem alım için kullanılmak üzere tek saat işareti PHY tarafından sağlanmaktadır. 10Mbps ethernet için saat frekansı 5 MHz, 100Mbps ethernet için saat frekansı 50MHz'dir. Şekil 3.6'da ilgili sinyaller gösterilmiştir.



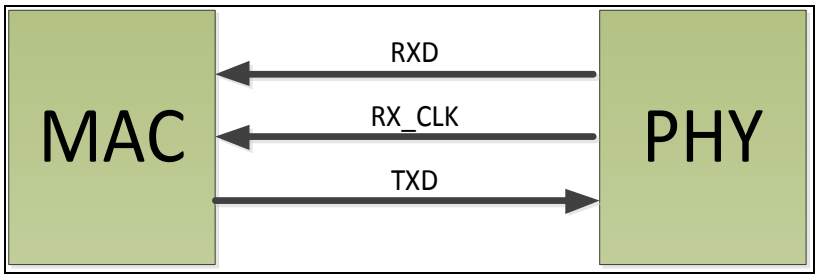
Şekil 3.6 : RMII.

- **Reduced Gigabit Media Independent Interface (RGMII):** Veri yolu 8 bit yerine 4 bit olarak değiştirilmiştir. Saat frekansı 125MHz olup, saat işaretinin hem yükselen hem düşen kenarı kullanılmaktadır. Şekil 3.7'de ilgili sinyaller gösterilmiştir.

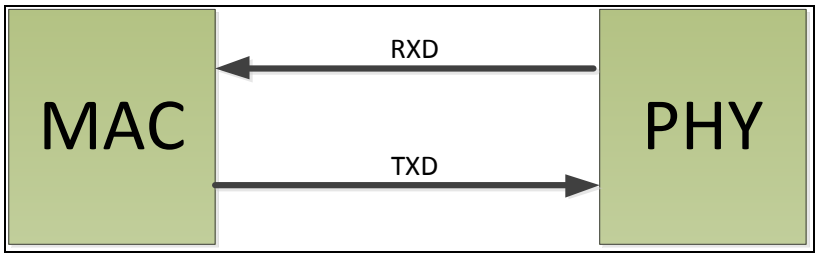


Şekil 3.7 : RGMII.

- **Serial Gigabit Media Independent Interface (SGMII):** Veri yolu 1 bit olarak değiştirilmiştir. Saat işaretli ve saat işaretsiz olmak üzere iki adet tipi bulunmaktadır. Saat işaretsiz olan tipte, saat veri yolundan elde edilmektedir. Saat frekansı 625MHz olup, saat işaretinin hem yükselen hem düşen kenarı kullanılmaktadır. Şekil 3.8 ve şekil 3.9'da ilgili sinyaller gösterilmiştir.

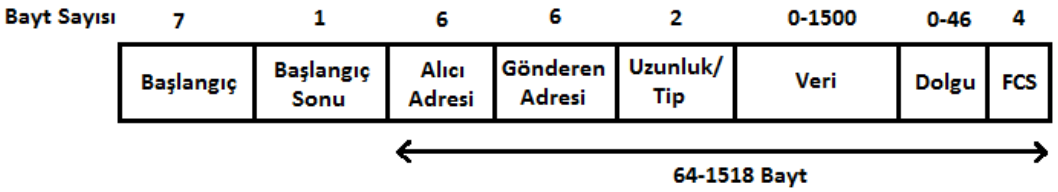


Şekil 3.8 : SGMII Saat İşaretli.



Şekil 3.9 : SGMII Saat İşaretsiz.

3.3 Ethernet Paket Yapısı



Şekil 3.10 : Ethernet Paket Yapısı.

Ethernet verisi paketler halinde parçalara ayrılarak gönderilmektedir. Paketteki bölümlerin gönderilme sırası soldan sağa, bir bayttaki bitlerin gönderilme sırası da en az anlamlı bitten en anlamlı bite doğrudur. MAC, veri büyüklüğünün 1500 bayttan fazla olduğu jumbo paketler de gönderebilmektedir. Şekil 3.10'daki Ethernet paket bileşenleri aşağıda anlatılmaktadır.

- **Başlangıç:** MAC tarafından otomatik olarak eklenmektedir. Senkronizasyon için kullanılmakta ve yedi adet 0x55 sayısından oluşmaktadır. Alıcı tarafında bu kısım otomatik olarak çıkarılmaktadır.

- **Başlangıç Sonu:** Paket başlangıcını belirtmekte ve 0xD5 değerini içermektedir. Gönderme sırasında MAC tarafından otomatik olarak eklenmektedir ve alma sırasında da otomatik olarak çıkarılmaktadır.

- **Alıcı Adresi:** Alıcının MAC adresini belirtmektedir. En az anlamlı biti adresin unicast (0) veya multicast (1) bir adres olduğunu belirtir. Eğer bütün bitler 1 olursa, bu broadcast adres olduğu anlamına gelmektedir.

- **Gönderen Adresi:** Gönderenin MAC adresidir.

- **Uzunluk/Tip:** Değerine göre uzunluk veya tip olarak değerlendirilir. 1536 ve üstü değerler için tip, 1536 altı değerler için ise uzunluk olarak değerlendirilmektedir. Uzunluk olarak değerlendirildiğinde, bu bölümü takip eden verinin uzunluğunu belirtmektedir. Bu uzunluğun içinde dolgu kısmının uzunluğu yoktur. Tip olarak değerlendirildiğinde ise paketin tipini belirtmektedir. Mesela 0x0800 internet protokolünü, 0x0806 ise adres çözümleme protokolünü belirtmektedir.

- **Veri:** Gönderilecek veriyi içermektedir ve 0-1500 bayt uzunluğunda olabilmektedir.

- **Dolgu:** Bu bölümün uzunluğu 0 ile 46 bayt arasında değişmektedir. Amacı ise paket uzunluğunun en az 64 bayt olmasını sağlamaktır. Eğer veri uzunluğu 0 bayt ise dolgu kısmının uzunluğu 46 bayt, veri uzunluğu 46 bayt veya üstü ise dolgu kısmının uzunluğu 0 olacaktır.

- **FCS:** Paket sınaması, alıcı adresi, gönderici adresi, tip/uzunluk, veri ve dolgu kısımları kullanılarak 32-bit Cyclic Redundancy Check (CRC) metoduna göre hesaplanmaktadır. Alıcı kısım da gelen veriye aynı metodu uygulayarak ve gönderim sırasında bir hata olup olmadığını tespit etmektedir.

3.4 Inter Frame Gap (IFG)

Ethernet paketinin son bit'i gönderildikten sonra, diğer paketin ilk bit'ini göndermeden önce IFG kadar süre beklemek gerekmektedir. Bunun sebebi, half-duplex sistemlerde ethernet kartının gönderim işleminden sonra gönderim modundan alım moduna geçmesi için süre gerekmesidir[8]. IFG süresi standartta 96 bit süresi olarak tanımlanmıştır. 10Mbps hızındaki ethernet için 9.6 us, 100Mbps hızındaki ethernet için 960 ns, 1Gbps ethernet için 96 ns'dir.

4. IP VE UDP KATMANLARI

4.1 IP Katmanı

IP katmanı ethernetin üzerinde, OSI mimarisinin 3. katmanı olan ağ katmanında çalışmaktadır. IP katmanının görevi bir bilgisayardan diğer bilgisayara sadece IP adresini kullanarak, bir veya birden fazla IP ağı üzerinden paketlerin gönderilmesidir[9]. IP paketi bilgisayardan gönderildikten sonra router'lardan geçerek alıcı bilgisayara iletilmektedir. IP bağlantısız bir protokoldür, bir bilgisayardan gönderilen IP paketinin diğer bilgisayara erişimini garanti etmemektedir[10]. Bu nedenle IP paketleri alıcıya bozulmuş bir şekilde, değişik sıralarda ulaşabilir, aynı paketten iki defa ulaşabilir veya hiç ulaşmayabilir. Paketin diğer bilgisayara erişiminin garantisi bir üst katmanda görev yapan TCP katmanı tarafından sağlanmaktadır.

İnternete bağlı her bilgisayarın bir IP adresi bulunmaktadır. Her IP adresi tektir ve dünya üzerinde aynı IP adresinden iki adet bulunamaz. Günümüzde hala yoğunlukla kullanılan IP adresleri IPv4 standardında olup 32-bit uzunluğundadır ve toplamda $2^{32} = 4294967296$ adet IP adresi tanımlanabilmektedir. Bazı IP adresleri özel kullanımlar için ayrıldığından normal kullanıma kapalıdır. Dünyadaki IP ağına bağlı cihazlar hızla arttığından dolayı, 32-bit uzunluğundaki IP adresi yeterli kalmamaktadır. Bu amaçla 128-bit uzunluğunda IP adresi tanımlayan IPv6 geliştirilmiştir.

IP adresleri 32-bit onaltılık tabanda veri şeklinde ifade edilebileceği gibi, kolay okunabilmesi için genellikle 8-bitlik dört gruba ayrılarak aralarına nokta konulup onluk tabanda ifade edilmektedir. Buna noktalı ondalık gösterim denmektedir. Örneğin 0xC0A80105 IP adresi, noktalı ondalık gösterimde 192.168.1.5 şeklinde ifade edilmektedir.

Bazı durumlarda ağı alt ağlara bölmek gerekebilmektedir. Bölme işlemi alt ağ maskesi kullanılarak yapılmaktadır. Alt ağa bölmenin sebepleri:

- 32-bitlik IP adresi yeterli kalmadığından dolayı, bölme işlemi kullanılan IP adreslerini arttırmaktadır.
- Broadcast paketlerin etki alanını daraltarak, tüm ağlardaki cihazların gereksiz paket almasını engellemektedir.
- Ağın yönetimi kolaylaşmaktadır.

4.1.1 IP adres sınıfları

IP adresleri A,B,C,D,E olmak üzere 5 gruba ayrılmışlardır.

A Sınıfı Adreslerde, IP adresinin ilk 8 bitlik kısmı 0 ile 127 arasında olup alt ağ maskesi 255.0.0.0'dır. ilk 8 bitlik kısmı ağ ID'sini, geri kalan 24 bitlik kısım ise host ID'sini belirtmektedir. İlk 8 bitin 0 ve 127 olma durumları özeldir ve kullanılmazlar. Dolayısıyla A tipi adres kullanan ağ sayısı maksimum 126 olabilmektedir. Her bir ağ $2^{24} - 2 = 16777214$ adet bilgisayar içerebilmektedir. Örneğin, 54.0.0.0 ağında kullanılabilecek IP adresleri 54.0.0.1 ile 54.255.255.254 aralığında olmalıdır. Çok sayıda bilgisayar içeren ağlar için uygundur.

B Sınıfı Adreslerde, IP adresinin ilk 8 bitlik kısmı 128 ile 191 arasında olup alt ağ maskesi 255.255.0.0'dır. ilk 16 bitlik kısmı ağ ID'sini, geri kalan 16 bitlik kısım ise host ID'sini belirtmektedir. B tipi adres kullanan ağ sayısı maksimum 16384 olabilmektedir. Her bir ağ $2^{16} - 2 = 65634$ adet bilgisayar içerebilmektedir. Örneğin, 167.63.0.0 ağında kullanılabilecek IP adresleri 167.63.0.1 ile 167.63.255.254 aralığında olmalıdır.

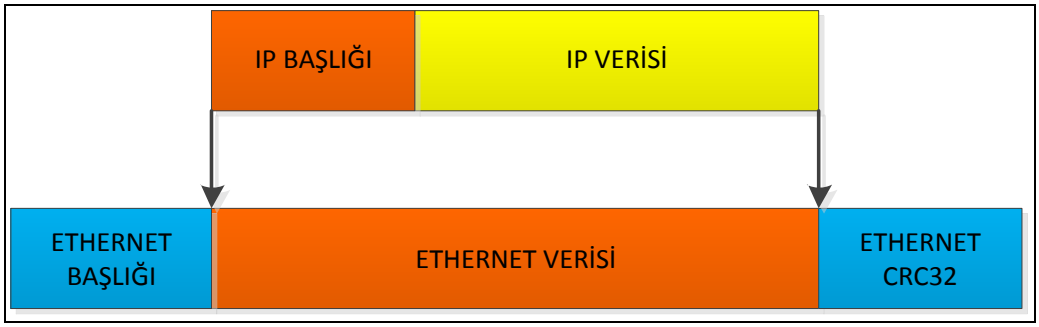
C Sınıfı Adreslerde, IP adresinin ilk 8 bitlik kısmı 192 ile 223 arasında olup alt ağ maskesi 255.255.255.0'dır. ilk 24 bitlik kısmı ağ ID'sini, geri kalan 8 bitlik kısım ise host ID'sini belirtmektedir. C tipi adres kullanan ağ sayısı maksimum 2097152 olabilmektedir. Her bir ağ $2^8 - 2 = 254$ adet bilgisayar içerebilmektedir. Örneğin, 192.168.5.0 ağında kullanılabilecek IP adresleri 192.168.5.1 ile 192.168.5.254 aralığında olmalıdır.

D Sınıfı Adreslerde, IP adresinin ilk 8 bitlik kısmı 224 ile 239 arasındadır ve multicast yayın için kullanılmaktadırlar.

E Sınıfı Adreslerde, IP adresinin ilk 8 bitlik kısmı 240 ile 255 arasındadır ve deneysel amaçlarla kullanılmaktadırlar.

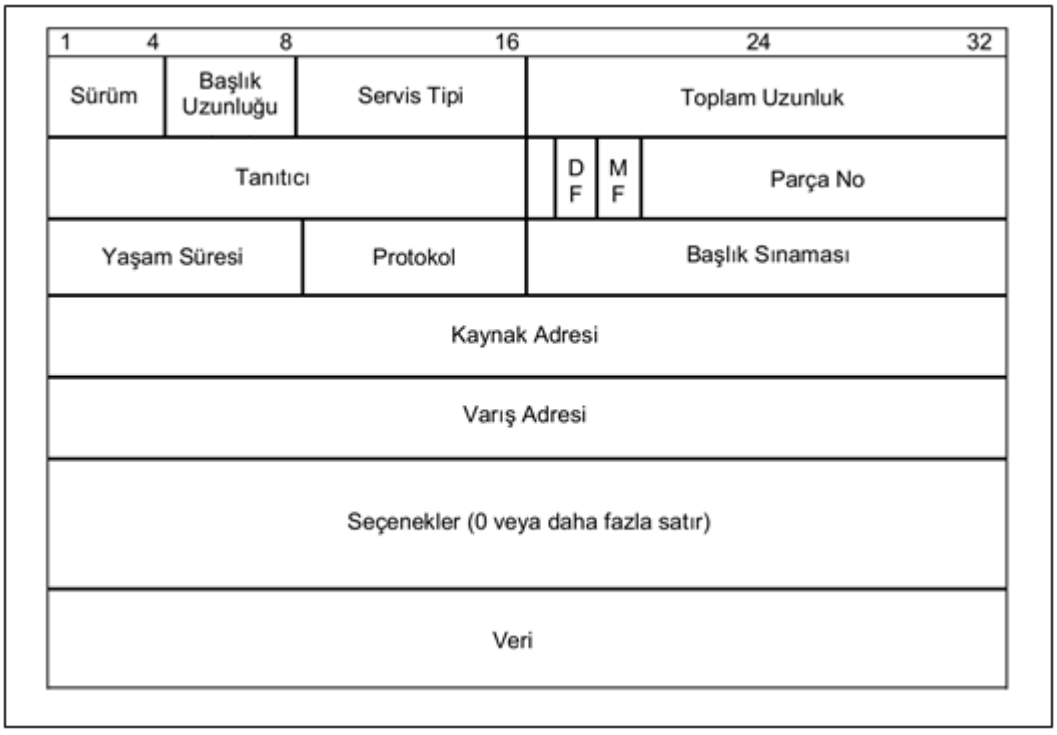
4.1.2 IP paket yapısı

IP verisi paketler halinde gönderilmektedir. Paketteki bölümlerin gönderilme sırası soldan sağadır. IP paketi, bir üst katmandan gelen paketin önüne IP başlıkları eklenerek oluşturulmaktadır. Daha sonra IP paketi de ethernet paketinin veri kısmını oluşturmaktadır. Bu ilişki şekil 4.1'de gösterilmiştir.



Şekil 4.1 : Ethernet ve IP Paketi.

IPv4 paketinin bileşenleri şekil 4.2'de gösterilmiştir ve aşağıda detaylı anlatılmaktadır.



Şekil 4.2 : IP Paket Yapısı.

- **Sürüm:** 4 bit uzunluğundadır. İnternet paketinin sürümünü belirtmektedir. IPv4 için 4 değerini almaktadır.

- **Başlık Uzunluğu:** 4 bit uzunluğundadır. Başlığın toplam uzunluğunu, 32-bit cinsinden vermektedir. 20 baytlık bir başlık uzunluğu için 5 değerini almaktadır.

- **Servis Tipi:** 8 bit uzunluğundadır. IP paketinin önceliğini ayarlama, düşük gecikmeli veya yüksek güvenilirlikli yoldan gitmesi için talepte bulunma gibi fonksiyonları vardır. Pratikte çok fazla kullanılmaz.

- **Toplam Uzunluk:** 16 bit uzunluğundadır. IP paketinin uzunluğunu belirtmektedir. Uzunluk hesaplanırken sadece veri uzunluğu değil, başlık uzunluğu da kullanılmaktadır.

- **Tanıtıcı:** 16 bit uzunluğundadır. Gönderilecek olan IP paketi, alt katmanın gönderebileceğinden daha uzunsa, paket parçalara ayrılarak gönderilmektedir. Her bir parça ayrı IP paketleri haline getirilip gönderilmekte, alıcı tarafında tekrar birleştirilmektedir. Parçalara ayrılan IP paketinin her bir parçasının tanıtıcı değeri aynı yazılmalıdır. Böylelikle alıcı taraf hangi parçaların aynı pakete ait olduğunu anlayabilmektedir.

- **Bayraklar:** 3 bit uzunluğundadır. Bitlerin anlamları:
  - ✓ **Bit 0 :** Kullanılmıyor, 0 olmalıdır.
  - ✓ **Don't Fragment(DF) :** Bu bit Router'in paketi parçalara ayıp ayıramayacağını belirtmektedir. 1 ise router paketi parçalara ayırabilir, 0 ise ayıramaz.
  - ✓ **More Fragments(MF) :** Eğer bir paket parçalanmışsa, son parça hariç bütün parçalarda 1 değerini almaktadır. Son parçada veya parçalanmamış IP paketinde 0 değerini almaktadır.

- **Parça Numarası:** 13 bit uzunluğundadır. Her bir parçanın kaçıncı bayttan itibaren verileri içerdiğini belirtir. Her bir 8 bayt için değeri 1 artar. İlk parça için sıfır değerini alır. Böylece alıcı taraf parçaların sırasını anlayıp parçalanmış IP paketini tekrar oluşturabilmektedir.

- **Yaşam Süresi:** 8 bit uzunluğundadır. Paketin sonsuz döngüye girmesini önlemek için paketin yaşamını kısıtlamaktadır. Her bir router'dan geçtiğinde 1 azaltılmakta ve sıfıra düştüğünde paket çöpe atılmaktadır.

- **Protokol:** 8 bit uzunluğundadır. Bir üst katmanda hangi taşıma katmanının kullanıldığını belirtmektedir. Örneğin 6 TCP, 17 UDP, 1 ICMP olduğunu göstermektedir.

- **Başlık Sınaması:** 16 bit uzunluğundadır. Hata kontrolü için kullanılmaktadır. Alıcı taraf paket için checksum hesaplamakta ve paketin içindeki checksum değeri ile karşılaştırmaktadır. Aldığı ile hesapladığı eşit değilse hata durumunu tespit edip uygun işlemi yapmaktadır. Kendisi hariç IP başlığı kullanılarak checksum oluşturulmaktadır. Az yük getirmesi için hesaplanırken IP paketindeki veri kullanılmaz. Paket her bir router'dan geçtiğinde yaşam süresi bilgisi değişeceğinden checksum'ı tekrar hesaplamak gerekmektedir. Bundan dolayı az yük getiren bir checksum olması önemlidir.

- **Kaynak IP Adresi:** 32 bit uzunluğundadır. Göndericinin IPv4 adresidir.

- **Varış IP Adresi:** 32 bit uzunluğundadır. Alıcının IPv4 adresidir.

- **Seçenekler:** Bazı durumlarda güvenlik gibi seçenekler ihtiyaç olabilmektedir. IP başlığı 32 bitin katları uzunluğunda olacağından, pakete seçenekler eklendiğinde kalan sıfır ile doldurulmalıdır. Aynı zamanda, seçenekler kullanıldığında başlık uzunluğu verisi de uygun değere getirilmelidir.

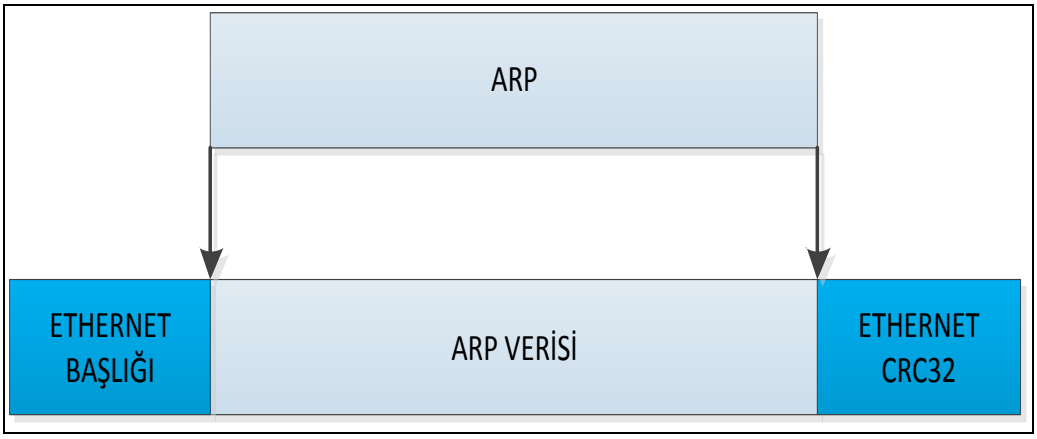
- **Veri:** IP paketinde gönderilmek istenen veridir.

4.1.3 Adres çözümleme protokolü (ARP)

Ağ üzerindeki bir bilgisayar diğer bir bilgisayara 3. katmanda paket göndermek istediğinde, pakete alıcının IP ve MAC adreslerini koymak zorundadır. Alıcının IP adresini bildiği ve MAC adresini bilmediği durumlarda MAC adresini öğrenmek için ARP İstek paketi göndermektedir[11]. ARP İstek paketi gönderenin IP ve MAC adreslerini, alıcının ise sadece IP adresini içermekte, alıcı MAC adresi için de broadcast 0xFFFFFFFFFFFF adresini kullanmaktadır. Ayrıca Length/Type bölümünde 0x0806 değerini içermektedir. Ağdaki bütün bilgisayarlar ARP İstek

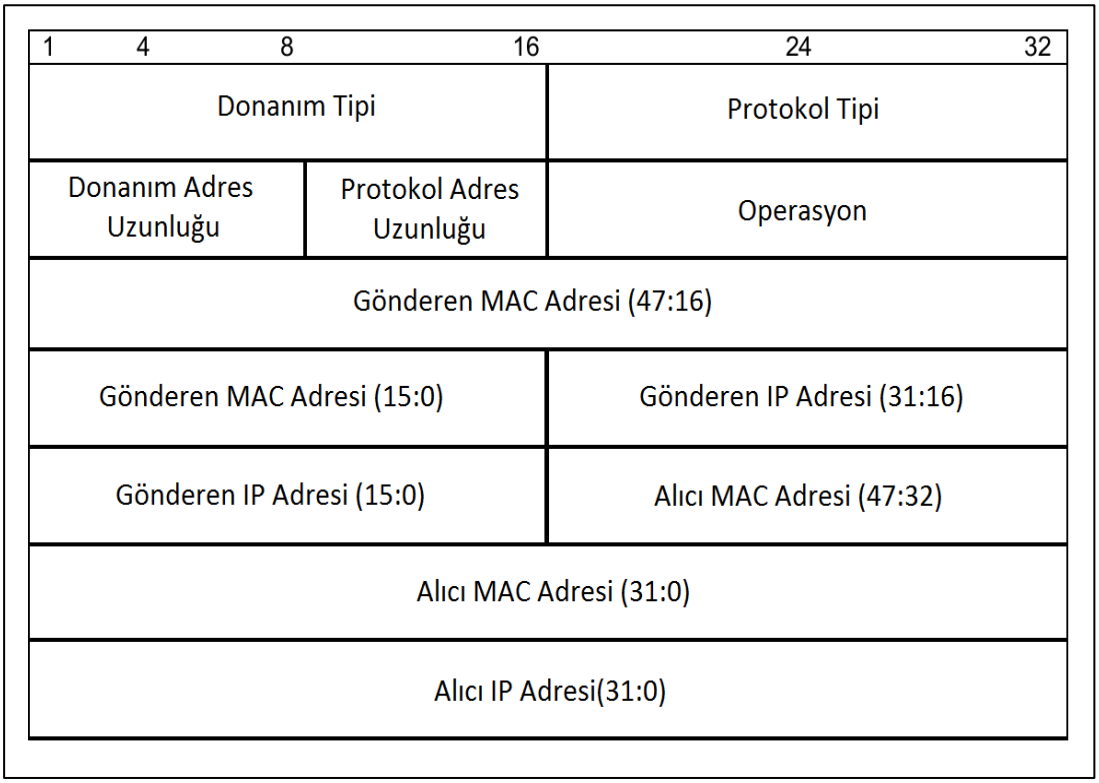
paketini aldıktan sonra, ilgili bilgisayar kendi MAC adresini de koyarak ARP Cevap paketini dönmektedir. Böylece her iki bilgisayar birbirlerinin MAC adresini öğrenip ARP tablolarına kaydetmektedir.

ARP paketindeki bölümlerin gönderilme sırası soldan sağadır. ARP paketi ethernet paketinin veri kısmını oluşturmaktadır. Bu ilişki şekil 4.3'de gösterilmiştir.



Şekil 4.3 : Ethernet ve ARP Paketi.

ARP paketinin bileşenleri şekil 4.4'de gösterilmiştir ve aşağıda detaylı anlatılmaktadır.



Şekil 4.4 : ARP Paket Yapısı.

- **Donanım Tipi:** 16 bit uzunluğundadır. İkinci katman protokolünün tipini belirtmektedir. Örneğin 1 değeri ethernet olduğunu göstermektedir.

- **Protokol Tipi:** 16 bit uzunluğundadır. Hangi üçüncü katman protokol için ARP sorgulaması yapıldığını belirtmektedir. Örneğin 0x0800 değeri IPv4 olduğunu göstermektedir.

- **Donanım Adres Uzunluğu:** 8 bit uzunluğundadır. Donanım adresinin kaç bayttan oluştuğunu göstermektedir. Örneğin 48 bitlik adres uzunluğu olan ethernet için 6 değerini almaktadır.

- **Protokol Adres Uzunluğu:** 8 bit uzunluğundadır. Protokol adresinin kaç bayttan oluştuğunu belirtmektedir. Örneğin 32 bitlik adres uzunluğu olan IPv4 için 4 değerini alır.

- **Operasyon:** 16 bit uzunluğundadır. Paketi gönderenin hangi işlem yapmak istediğini göstermektedir. Örneğin ARP istek için 1 değerini, ARP cevap için 2 değerini almaktadır.

- **Gönderen MAC Adresi:** 48 bit uzunluğundadır. Gönderen bilgisayarın MAC adresini içermektedir.

- **Gönderen IP Adresi:** 32 bit uzunluğundadır. Gönderen bilgisayarın IP adresini içermektedir.

- **Alıcı MAC Adresi:** 48 bit uzunluğundadır. Alan bilgisayarın MAC adresini içermektedir. ARP istek paketinde alıcının MAC adresi bilinmediğinden dolayı 0x000000000000 değerini almaktadır. ARP cevap paketlerinde ise, bilgisayar ARP istek gönderen bilgisayara cevap vereceğinden, ARP istek gönderen bilgisayarın MAC adresini içermektedir.

- **Alıcı IP Adresi:** 32 bit uzunluğundadır. Alan bilgisayarın IP adresini içermektedir.

4.2 UDP Katmanı

Ağ katmanının görevi, ağ içinde veya farklı ağlardaki iki bilgisayarın birbiri ile haberleşebilmesidir[12]. Taşıma katmanının görevi ise, iki bilgisayarda çalışan farklı programların veya işlemlerin birbirleri ile haberleşmesidir. Bu haberleşme, güvenli ve güvensiz olmak üzere iki şekilde gerçekleşebilir.

TCP iki bilgisayar arasında güvenli ve bağlantı tabanlı veri iletişimi sağlamaktadır[13]. İlk paketi göndermeden önce alıcı bilgisayar ile bağlantı kurmaktadır. TCP paket bozulmaları, paket kayıpları, paket sıralarının değişmesi gibi ağ üzerinde olabilecek sorunlara çözüm üretmektedir. Gönderdiği paketlerden sonra alıcı taraf onay mesajı göndermekte, böylece paketlerin kaybolmadan alıcıya ulaştığı garantilenmektedir. Uygulama katmanına bildirmeden, hatalı paketleri tespit etmekte, tekrar gönderilmelerini sağlamakta, uygulama geliştirici hatalı paketlerle uğraşmak zorunda kalmamaktadır. Ayrıca akış kontrolü sağlayarak da alıcı bilgisayar hıza yetişemediğinde göndericiyi yavaşlatabilmektedir.

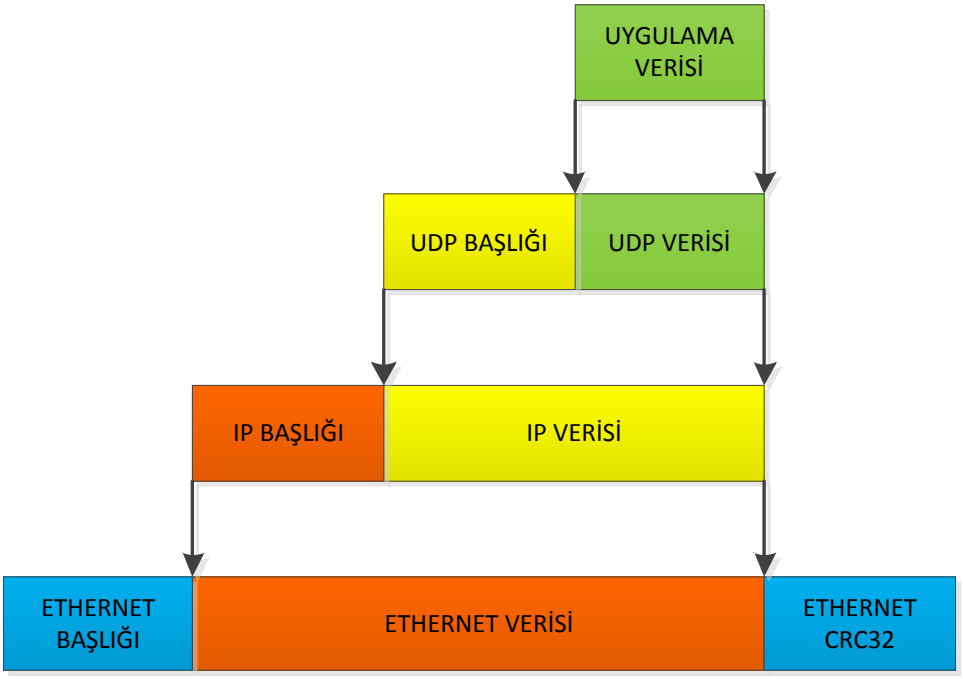
UDP iki bilgisayar arasında güvensiz ve bağlantısız veri iletişimi sağlamaktadır[14]. TCP gibi her paketin karşıya ulaşması garantisini vermemektedir. Bilgisayar paketi gönderdikten sonra paketi unutur ve artık paketle ilgili hiç bir işlem yapmaz. UDP'de hata kontrolü vardır, fakat TCP'deki gibi detaylı bir hata düzeltme mekanizması yoktur. TCP gibi güvenilir bir yöntem yerine UDP gibi bir güvensiz bir yöntemi tercih etme sebepleri aşağıda sıralanmıştır:

- Daha az başlık içerdiğinden daha verimlidir.
- Her paket için onay istemediğinden daha hızlıdır.
- Bağlantısız bir protokol olduğundan işlemesi daha kolaydır.
- IP üzerinden ses gibi zaman kritik uygulamalar için kullanılmaktadır.

4.2.1 UDP paket yapısı

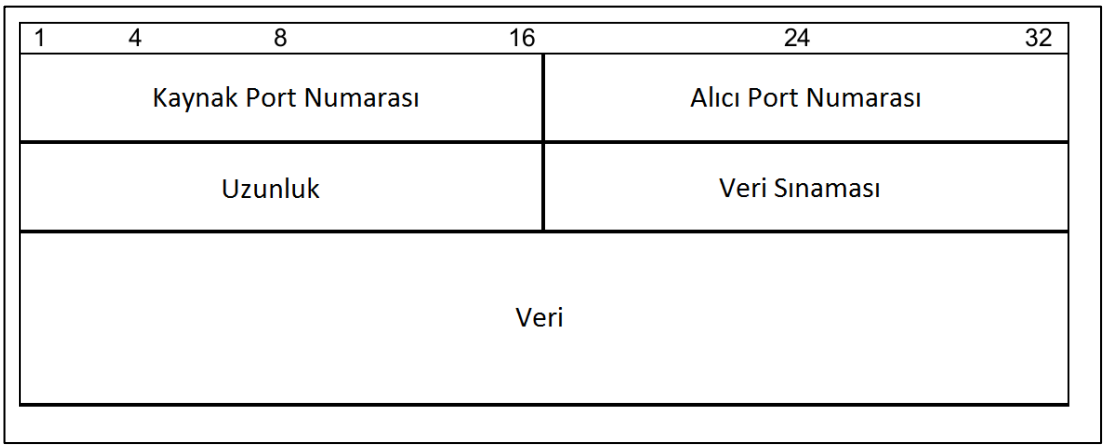
UDP verisi paketler halinde gönderilmektedir. Paketteki bölümlerin gönderilme sırası soldan sağadır. UDP paketi, bir üst katmandan gelen paketin önüne UDP

başlıkları eklenerek oluşturulmaktadır. Daha sonra UDP paketi de IP paketinin veri kısmını oluşturmaktadır. Bu ilişki şekil 4.5'de gösterilmiştir.



Şekil 4.5 : Ethernet, IP ve UDP Paketi.

UDP paket yapısı şekil 4.6'da gösterilmiştir ve paket bileşenleri aşağıda anlatılmaktadır.



Şekil 4.6 : UDP Paket Yapısı.

- **Kaynak Port Numarası:** 16 bit uzunluğundadır. Gönderici bilgisayardaki uygulamanın port numarasıdır. Port numaraları sayesinde aynı bilgisayardaki farklı uygulamalar adreslenebilmektedir. Alıcı bilgisayardaki uygulama cevap vermek istediğinde bu port numarasını kullanmaktadır.

- **Alıcı Port Numarası:** 16 bit uzunluğundadır. Alıcı bilgisayardaki uygulamanın port numarasıdır.

- **Uzunluk:** 16 bit uzunluğundadır. Veri ve 8 baytlık başlığın toplam uzunluğudur.

- **Veri Sınaması:** Verinin alıcı tarafından doğru alınıp alınmadığını kontrol etmek amacı ile kullanılmaktadır. Gönderici veri sınamasını kullanmak istemediğinde bu kısım sıfır ile doldurulmaktadır.

5. YENİDEN PROGRAMLANABİLİR YAPILAR VE FPGA

Eskiden analog gerçeklemeleri olsa da mikroişlemciler, bilgisayarlar, haberleşme sistemleri, işaret ve görüntü işleme sistemleri gibi hayatımızda önemli rol oynayan sistemler, günümüzde büyük oranda sayısal olarak gerçeklenmektedir. İlk sayısal elektronik uygulamaları manyetik olarak kontrol edilebilen röleler kullanılarak gerçekleştirilmesine rağmen sayısal hesaplama çağı, vakum tüpünün bulunması ile başlamıştır.

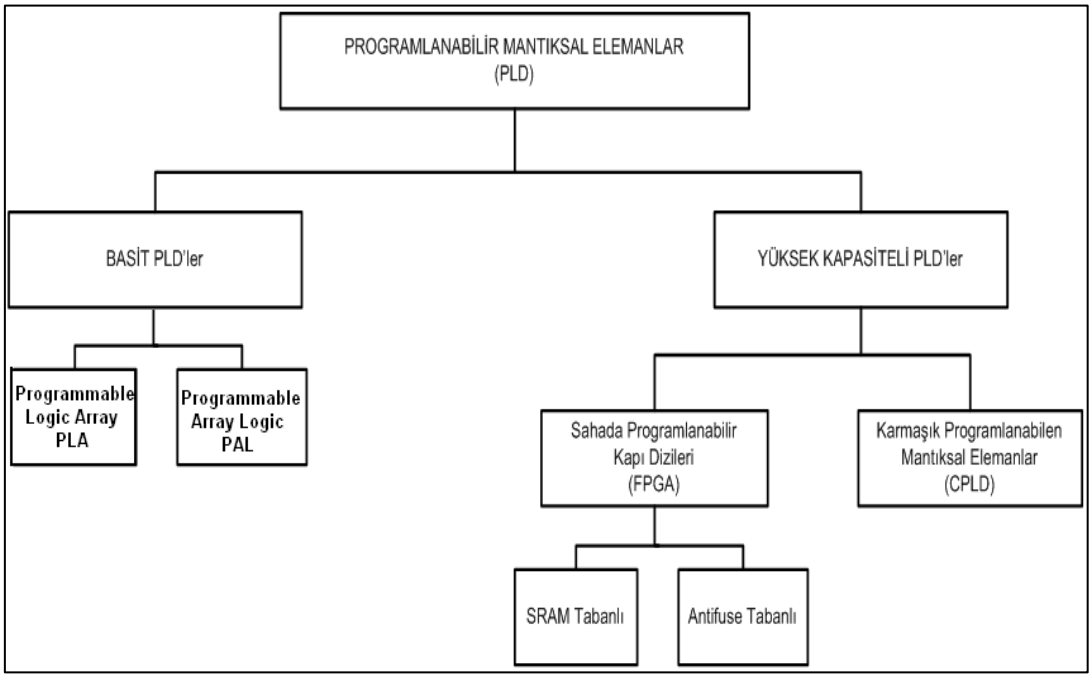
Vakum tüpü kullanılarak ilk bilgisayar olan ENIAC (Electronic Numerical Integrator and Calculator) 1945'de tasarlanmıştır. 2. dünya savaşında Amerikan ordusunun top atış hesaplamaları amacıyla kullanılmıştır. Çok yüksek güç çektiğinden soğutma sorunu vardı. Tüpler sürekli arıza yapıyordu, arıza yapmadan ortalama çalışma süresi 6-7 dakikaydı. ENIAC 25 m uzunluğunda, 3 m yüksekliğinde olup 18000 vakum tüp içermekteydi. İşlem gücü şuan ki bir hesap makinesinden çok daha düşüktü.

1947 yılında Bell Laboratuar'ında ilk bipolar transistörün keşfedilmesi, analog ve sayısal elektroniğin büyük atılım yapmasını sağlamıştır. Bu sayede TTL (Transistor-Transistor Logic) ailesi kapılar ve entegre devreler ticari olarak üretilmiştir. Fakat yüksek güç tüketimi, bir çipe güvenilir bir şekilde sığdırılabilecek transistör sayısını sınırlıyordu. MOS (Metal Oxide Semiconductor) transistörlerin keşfi ve ekonomik olarak üretim yöntemlerinin geliştirilmesi ile Bipolar transistörler yerini daha az güç harcayan MOS transistörlere bırakmıştır.

5.1 Yeniden Programlanabilir Yapılar

ASIC tasarlamanın zorlukları tasarımcıları yeniden programlanabilir yapılara yöneltmiştir. Yeniden programlanabilir yapılar, hazır devre blokları ve konfigüre edilebilir ara bağlantılar içerir[15]. İstenilen devre fonksiyonuna göre bloklar ve ara bağlantılar ayarlanarak yeniden programlanabilir yapılar arzu edilen işleve sahip olur. Yeniden Programlanabilir yapılar PLD (Programmable Logic Device) olarak adlandırılır. PLD kullanıcı tarafından programlanabilen tümdevreler için genel bir

tanımdır. PLD'ler karmaşıklığına göre basit ve yüksek kapasiteli olmak üzere ikiye ayrılmıştır.



Şekil 5.1 : Programlanabilir Mantıksal Elemanlar.

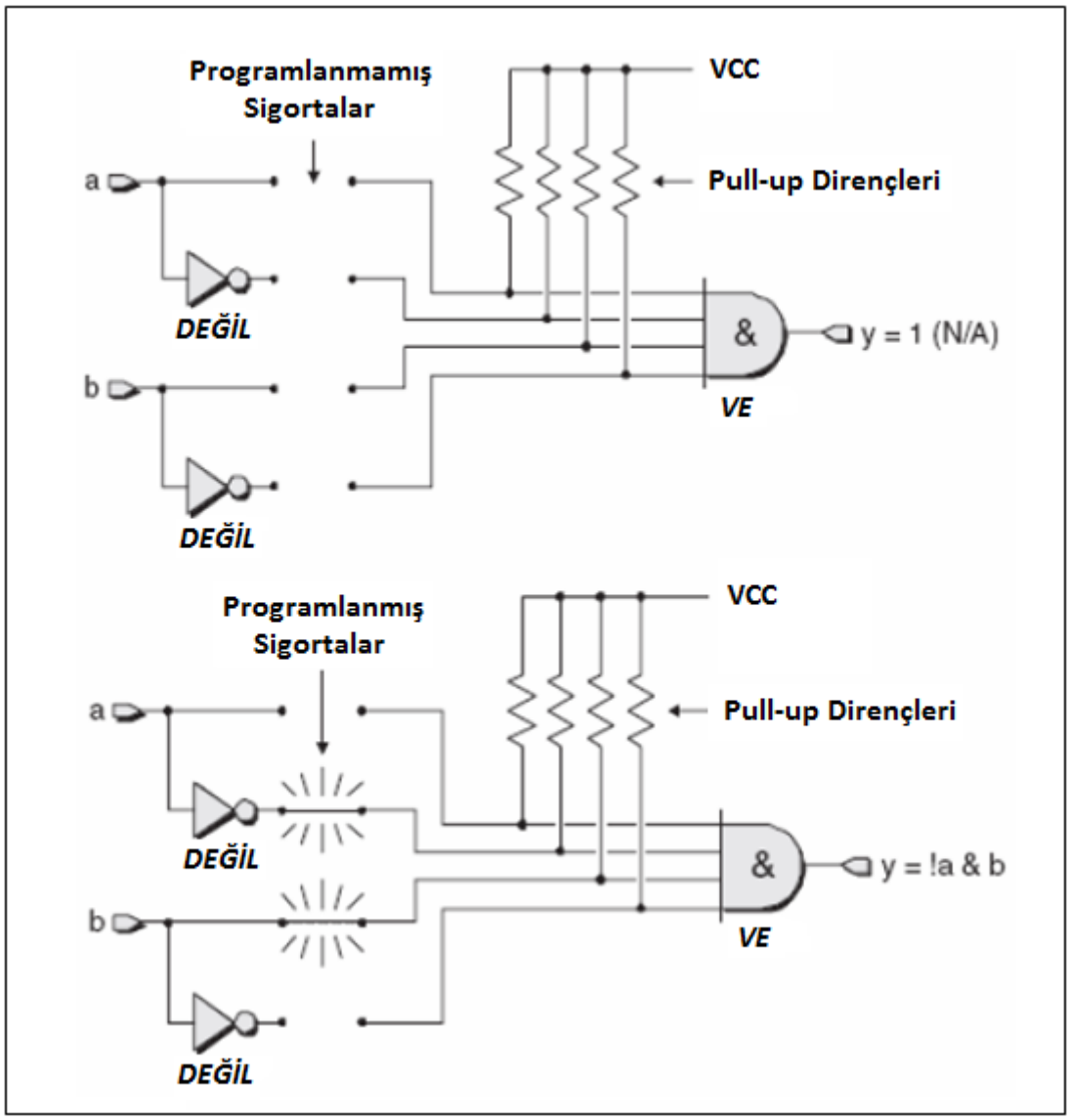
5.1.1 Anti-Sigorta teknolojisi

Kullanıcının kendi devresini programlamasına izin veren ilk teknolojidir[16]. Sigorta normalde yüksek empedans iken (>100M), sigorta yüksek gerilimle deforme edildikten sonra 100 ohm civarında empedansa sahip olur. Maliyeti düşüktür.

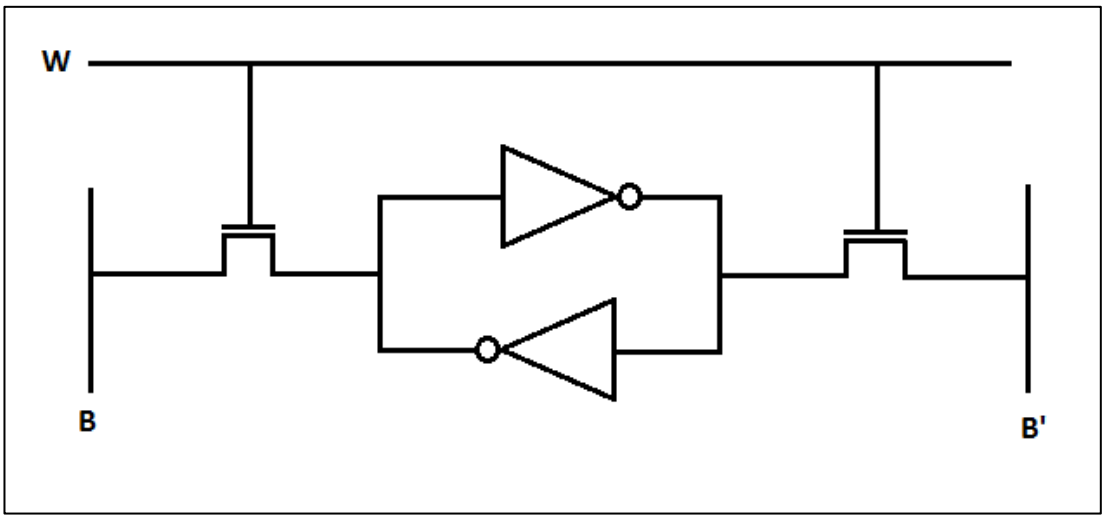
Avantajları arasında konfigürasyonun kalıcı olması, programlamak için harici belleğe ihtiyaç olmaması ve radyasyona karşı dayanıklı olması sayılabilir. Bir kez programlanabilir olduğundan prototip hazırlamak için uygun olmaması da dezavantajıdır. Örnek bir lojik fonksiyonu gerçekleyen devre şekil 5.2'de verilmiştir.

5.1.2 Statik bellek (SRAM) teknolojisi

Şekil 5.3'de gösterildiği gibi iki adet NOT kapısının bağlanması ile elde edilir. 6 adet transistörden oluşur. Saklanan veriyi elektrik kesilmediği sürece saklar. Sakladığı veri ile transistörü açar ya da kapar. Avantajları arasında çok kez ve hızlı programlanabilir olması sayılabilir. Dezavantajları arasında ise konfigürasyon verisini tutmak için harici belleğe ihtiyaç olması ve 6 transistör ile büyük alan kaplaması sayılabilir.



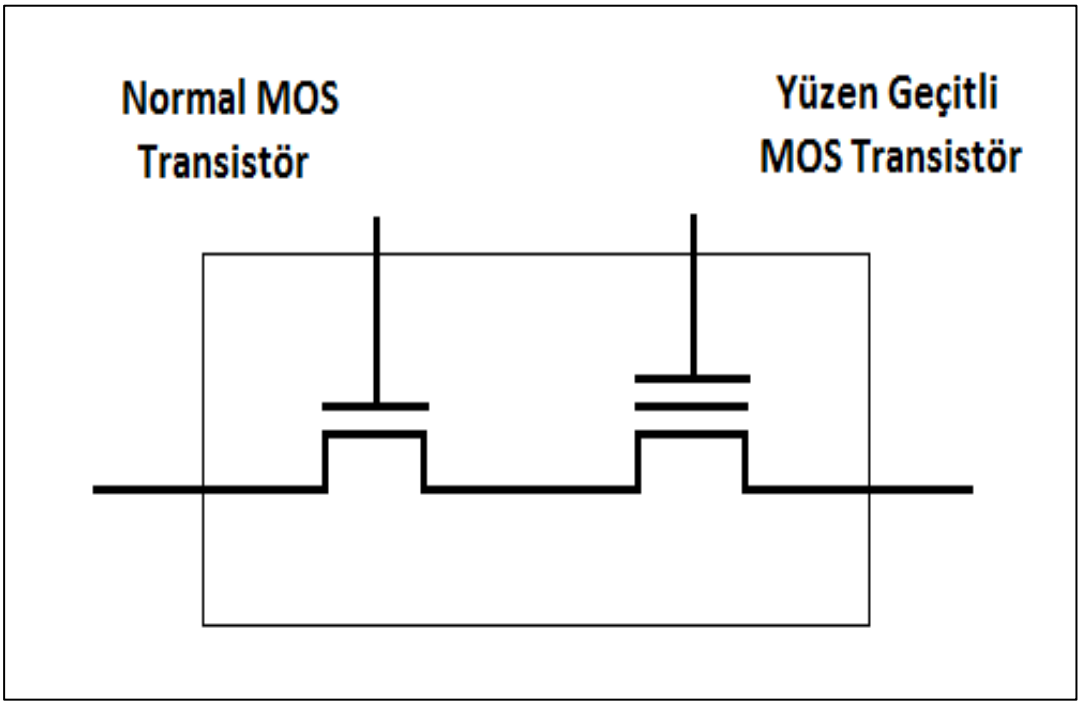
Şekil 5.2 : Anti-Sigorta Teknolojisi[17].



Şekil 5.3 : SRAM Hücresinin Genel Yapısı.

5.1.3 EEPROM/Flash teknolojisi

Bir adet MOS transistör ve bir adet FGMOS (Floating Gate MOS) transistörden oluşur. FGMOS üzerine uygulanan gerilimi yüzen geçidinde tutarak, güç kesildiğinde dahi veriyi kaybetmez. İkinci transistör elektriksel silme işlemi içindir. Avantajları arasında konfigürasyonun kalıcı olması ve harici bellek gerektirmemesi sayılabilir. Dezavantajları arasında ise konfigürasyon süresinin uzun olması ve özel proses gerektirmesi sayılabilir. Yapısı şekil 5.4'de verilmiştir.



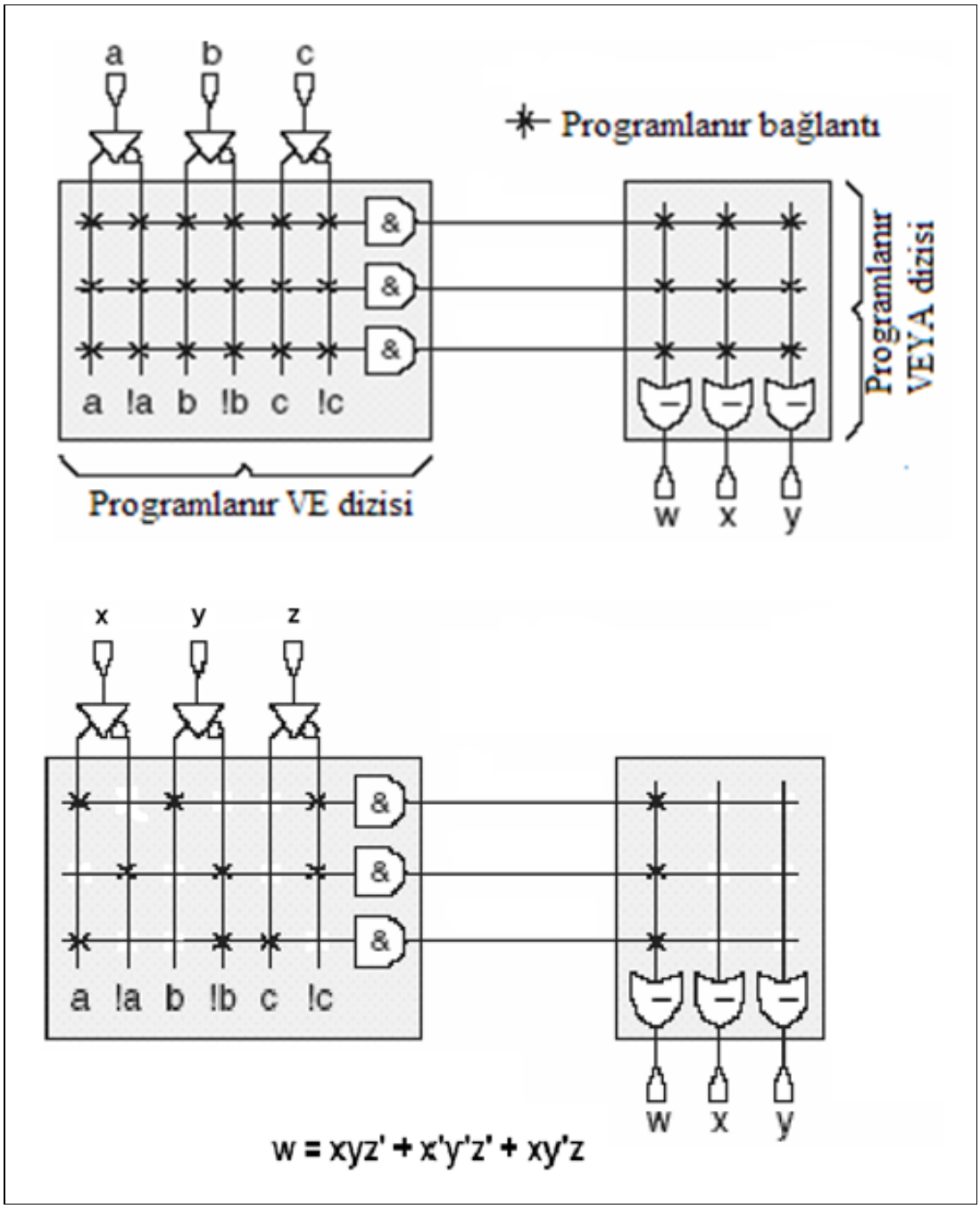
Şekil 5.4 : EEPROM Hücre Yapısı.

5.2 Programlanabilir Lojik Dizi (PLA)

Bütün lojik fonksiyonlar çarpımların toplamı şeklinde ifade edilebilir. PLA'lar ***and*** ve ***or*** matrislerinden oluşur. Çarpma işlemleri ***and*** matrisinde, toplama işlemleri ***or*** matrisinde gerçekleşir. Şekil 5.5'de konfigüre edilmemiş bir PLA ve aşağıdaki fonksiyonu sağlaması amacı ile konfigüre edilmiş PLA gösterilmektedir.

5.3 Programlanabilir Dizi Lojiği (PAL)

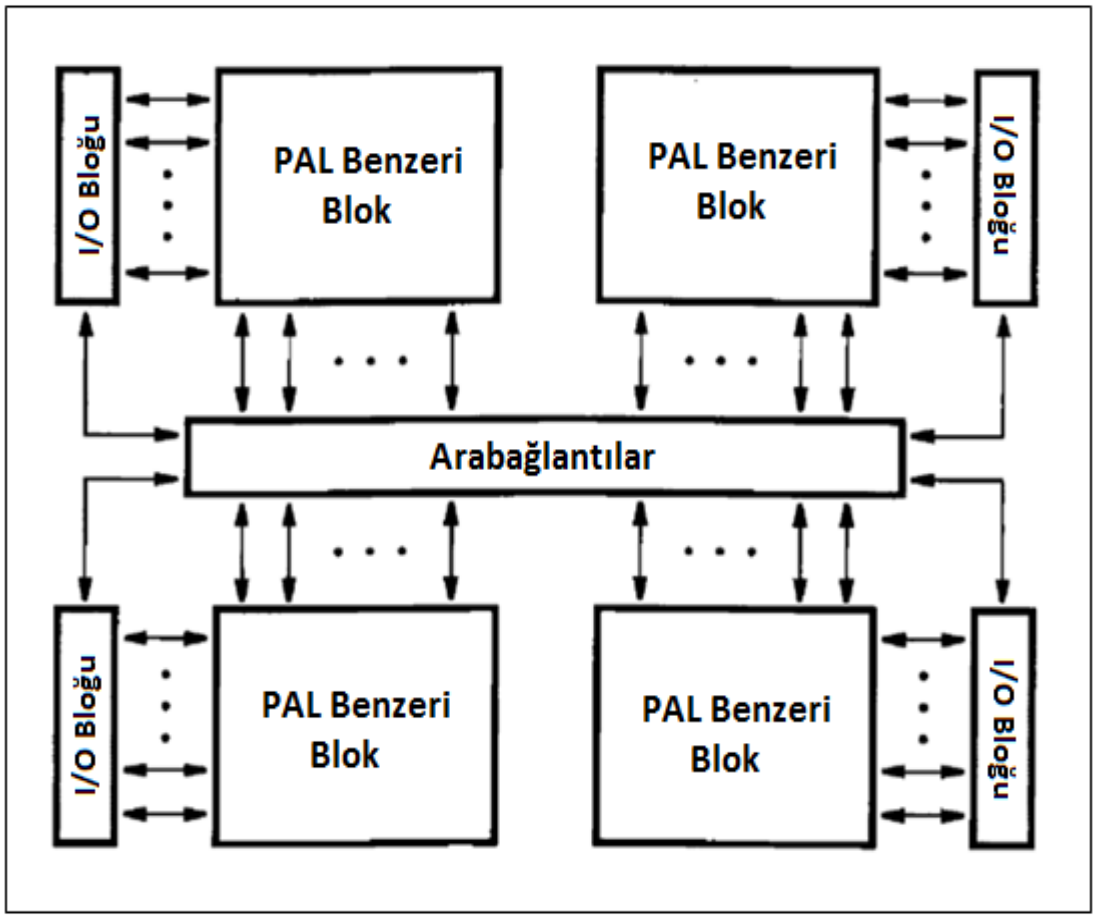
Veya matrisi sabit, ***ve*** matrisi programlanabilirdir. Giriş ve çıkışlara MUX (Çoğullayıcı) ve EXOR (Ayrıcalıklı veya) gibi basit lojik elemanlar eklenmiştir. Flip-flop içermektedir, böylece basit ardışıl devreler yapılabilmektedir.



Şekil 5.5 : PLA Genel Yapısı[17].

5.4 Kompleks Programlanabilir Lojik Yapılar (CPLD)

PLD'lerin performansını iyileştirmek, kullanılan silisyum alanını azaltmak ve güvenilirliği arttırmak için Kompleks PLD'ler üretilmiştir (CPLD). CPLD programlanabilir lojik bloklar ve bunları birbirine bağlayan programlanabilir ara bağlantılar içermektedir. CPLD'ler, basit PLD ile FPGA arasında geçiş döneminde yer alır. Temel yapılar, fonksiyonel bloklar, giriş/çıkış blokları ve ara bağlantı matrisidir. Şekil 5.6'da CPLD'nin genel yapısı verilmiştir.



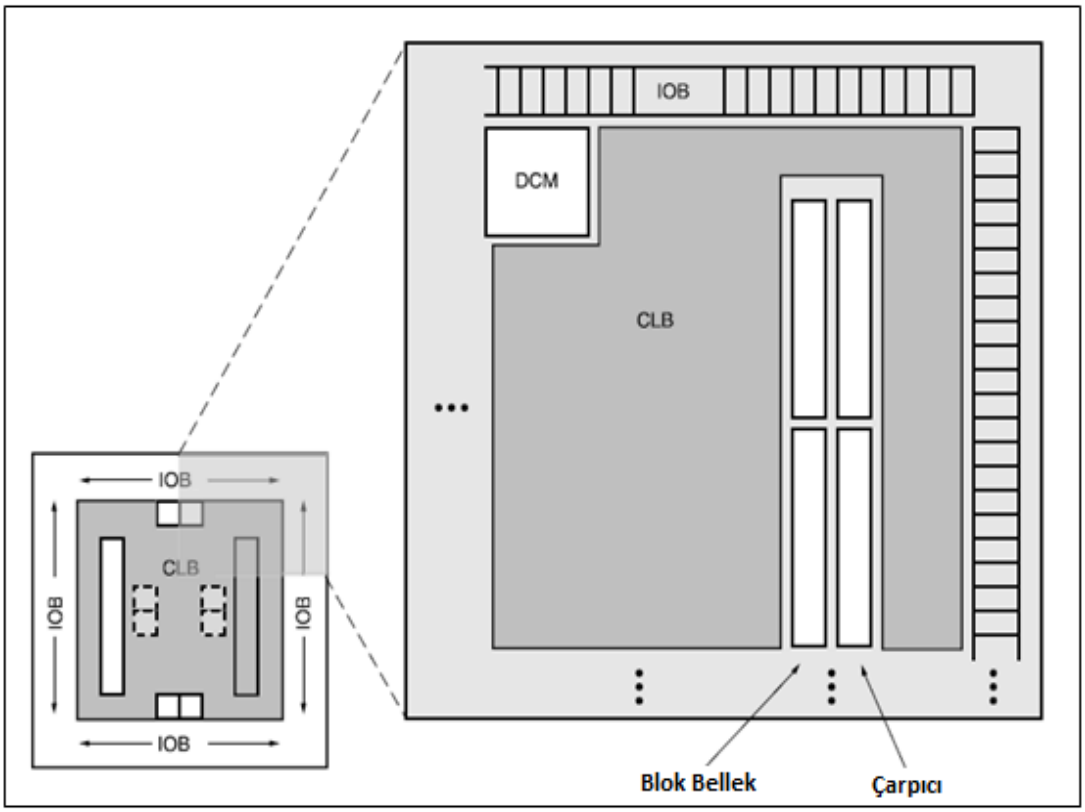
Şekil 5.6 : CPLD'nin Genel Yapısı[16].

Fonksiyonel bloklar genelde 16 PAL yapısından oluşur. 5-20 girişli OR kapısı bulunmaktadır. OR kapısının çıkışındaki XOR kapısının amacı çıkışın tersini alabilmektir. Flip-flop ise ardışıl devre oluşturmak ve veri saklayabilmek için kullanılır (register). Multiplexer ile flip-flopun kullanılıp kullanılmayacağı kararı verilir. Tri-state buffer ise pinin giriş veya çıkış olarak ayarlanmasını sağlar. CPLD'ler en az 2 en fazla yüzlerce fonksiyonel blok içerir.

5.5 Sahada Programlanabilir Kapı Dizileri (FPGA)

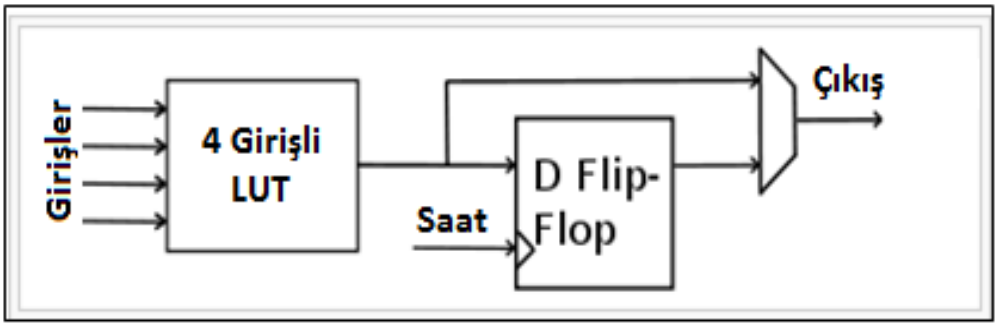
FPGA'ler (Sahada Programlanabilir Kapı Dizileri), CPLD'lerden daha gelişmiş yapıya sahiptirler. VHDL (Very High Speed Integrated Circuit Hardware Description Language) ve Verilog donanım tanımlama dilleri kullanılarak yazılan kodlar, özel yazılımlar sayesinde sayısal devrelere dönüştürülür ve bu devreler FPGA'ların içindeki bloklar kullanılarak oluşturulur. Temel yapı taşları, programlanabilir lojik bloklar (CLB), programlanabilir ara bağlantılar ve giriş/çıkış bloklarıdır. Ayrıca blok bellek, DSP Blokları (çarpıcılar, MAC(multiply and accumulate)), saat yönetimi

(DCM, BUFGMUX), Ethernet MAC, PCI kontrolör ve gömülü PowerPC işlemci gibi bileşenleri bulunmaktadır. Şekil 5.7'de Spartan 3E ailesinin yapısı gösterilmiştir.



Şekil 5.7 : Spartan 3E'nin Genel Yapısı[18].

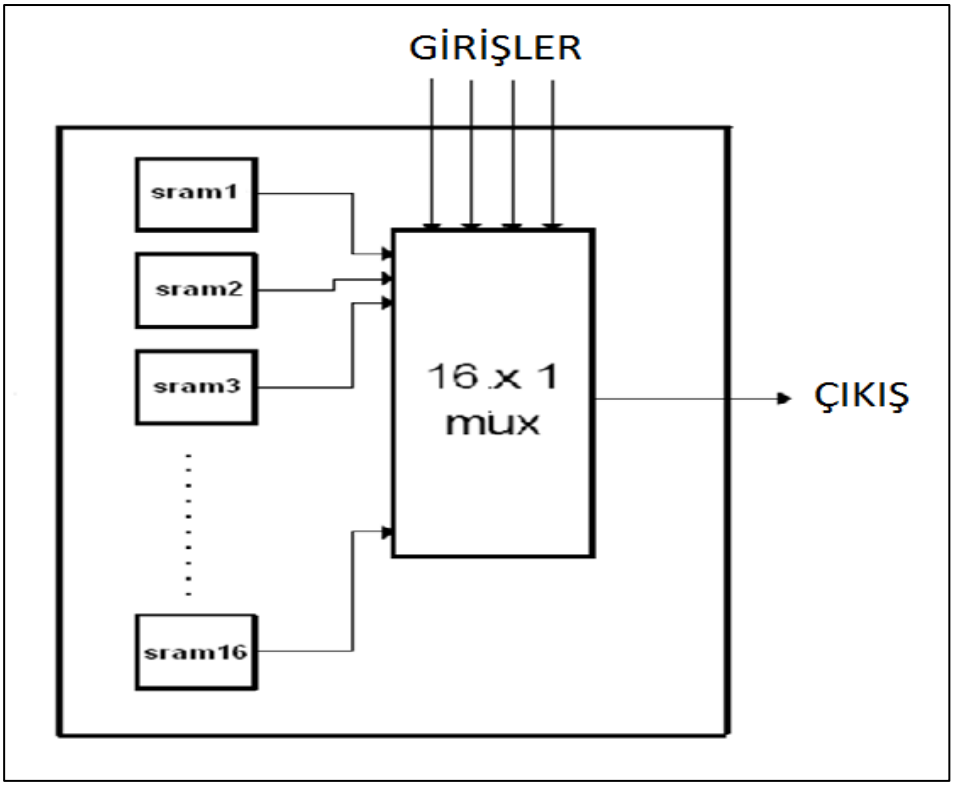
CLB'lerin temel yapısı şekil 5.8'deki gibidir. 4 girişli LUT (Look Up Table) devrenin combinezonsal kısmını, flip-flop ise ardışıl kısmını oluşturur. Multiplexer ise doğrudan LUT'un çıkışı ile flip-flop çıkışı arasında seçim yapar.



Şekil 5.8 : CLB'nin Genel Yapısı.

Günümüzde 3-6 girişli LUT'lar mevcuttur. 4-LUT ile, 4 girişli ve 1 çıkışlı bütün lojik fonksiyonlar karmaşıklıklarından bağımsız olarak gerçeklenebilir. Kapladığı alan ve gecikmesi göz önüne alındığında 4-LUT'lar optimum çözüm olarak

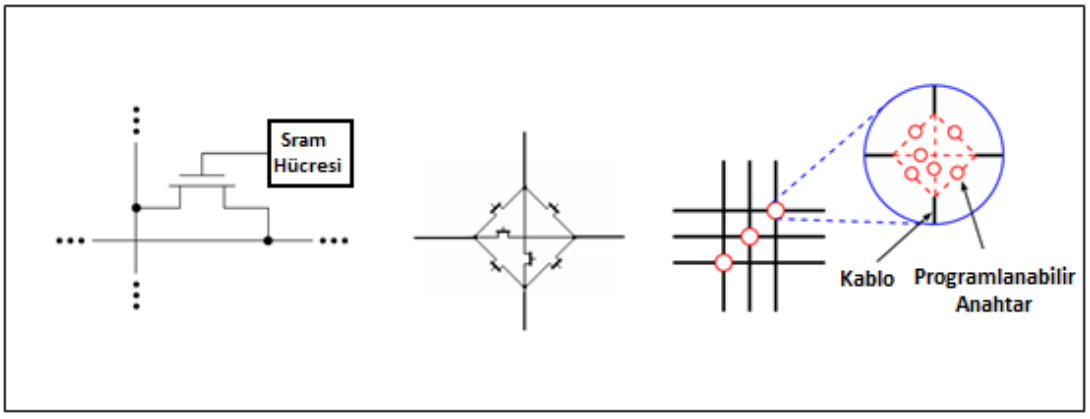
belirlenmiştir. Yapıları girişlerine SRAM kullanılarak sabit değerler verilmiş MUX'a benzetilebilir. Konfigürasyon sırasında SRAM hücrelerine uygun değerler yazılarak LUT'ların fonksiyonları belirlenir. Girişten çıkışa sabit gecikmeye sahiptirler. Şekil 5.9'da 4 girişli LUT'un yapısı gösterilmiştir.



Şekil 5.9 : LUT'un Genel Yapısı.

5.5.1 Konfigüre edilebilir ara bağlantılar

CLB'leri birbirine ve giriş-çıkış bloklarına bağlamak için kullanılırlar. Sram tarafından sürülen MOS transistörler anahtar olarak kullanılır.



Şekil 5.10 : Konfigüre Edilebilir Ara Bağlantıların Detaylı Yapısı.

5.5.2 Giriş çıkış birimleri

Giriş/Çıkış blokları, kılıf bacaklarıyla tasarım için kullanılan birimler (CLB, Blok RAM) arasında bağlantı kurar. FPGA' ların giriş çıkış blokları; giriş, çıkış veya giriş-çıkış olarak kurgulanabilir. FPGA çipinin altına sıralı bir şekilde yerleştirilmiş 1000'den fazla bacak bulunabilir. Giriş/Çıkış blokları LVTTL, LVCMOS, LVDS gibi farklı işaretleşme standartlarını desteklemektedir.

5.5.3 Blok bellek

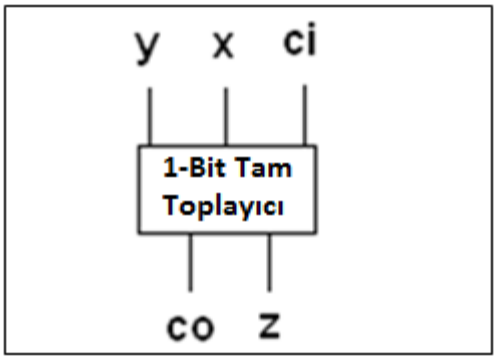
Çoğu uygulamanın bellek ihtiyacı olduğundan dolayı, FPGA'lar büyük miktarda Blok Bellek isimli SRAM içerirler. Bloklar halinde çipe dağıtılırlar. Kapasitesi megabaytlar mertebesinde olabilir. Spartan 3E kiti üzerindeki FPGA'da 45 Kb bulunur. Küçük parçalar halinde kullanılabileceği gibi, büyük bellekler oluşturmak için bir araya getirilebilirler. 18 Kbit'lik (2 KB) bloklar halinde bulunurlar. 100 baytlık blok ram yaratılsa dahi, bir blok harcanmış olur. Çift portlu ram, FIFO gibi yapılar oluşturmak için kullanılabilirler. FPGA'nın programlanması sırasında içeriği istenilen şekilde doldurulabilir.

5.5.4 Dağınık bellek

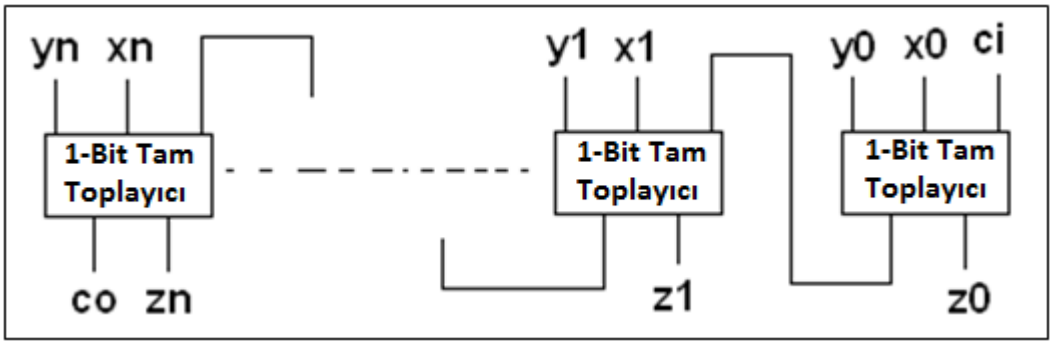
4-LUT'lar istenildiğinde 16x1 bitlik ram olarak kullanılabilirler. 8 tanesi birleştirilerek 16x8'lik (16 bayt) ve üstü miktarda ram oluşturulabilir. FPGA'nın her yerine dağıldığından dolayı bunlara dağınık (distributed) ram denir. Mümkünse blok ram kullanmak gerekir çünkü Distributed ram fonksiyonel kısımlardan(LUT) harcar. Düşük miktarlarda bellek ihtiyacı olduğunda kullanmak uygun olurlar. Ayrıca LUT'lar 16 bitlik kaydırıcı (shift register) olarak kullanılabilir. Arka arkaya bağlanarak daha büyük shift registerlar oluşturulabilir.

5.5.5 Hızlı elde zincirleri

FPGA'larda toplama işlemleri Ripple Carry Adder kullanılarak gerçeklenir. Ripple Carry Adder, tam toplayıcıların arka arkaya dizilmesi ile oluşur. Toplayıcının hızını etkileyen en önemli faktör elde zinciridir. Bu amaçla LUT'lar arasında hızlı elde zincirleri bulunmaktadır. Böylece toplama işlemleri çok hızlı gerçeklenebilir. Şekil 5.11'de 1-bit tam toplayıcı, şekil 5.12'de bundan oluşturulmuş N-Bit tam toplayıcı devresi verilmiştir.



Şekil 5.11 : 1-Bit Tam Toplayıcı.



Şekil 5.12 : N-Bit Tam Toplayıcı.

5.5.6 Gömülü çarpıcılar

Çarpıcılar programlanabilir lojik bloklar kullanılarak gerçeklenirse, hem çok yer kaplar hem de yavaş olurlar. Çarpıcılar matematiksel işlemlerin kullanıldığı pek çok uygulamada gerekli olduğu için, FPGA'lar gömülü çarpıcılar içerirler. 18x18 bitlik işaretli çarpma yapabilmektedirler. Beraber kullanılarak daha büyük boyutlarda çarpma işlemlerine izin vermektedirler.

6. SİSTEMİN FPGA ÜZERİNDE GERÇEKLENMESİ

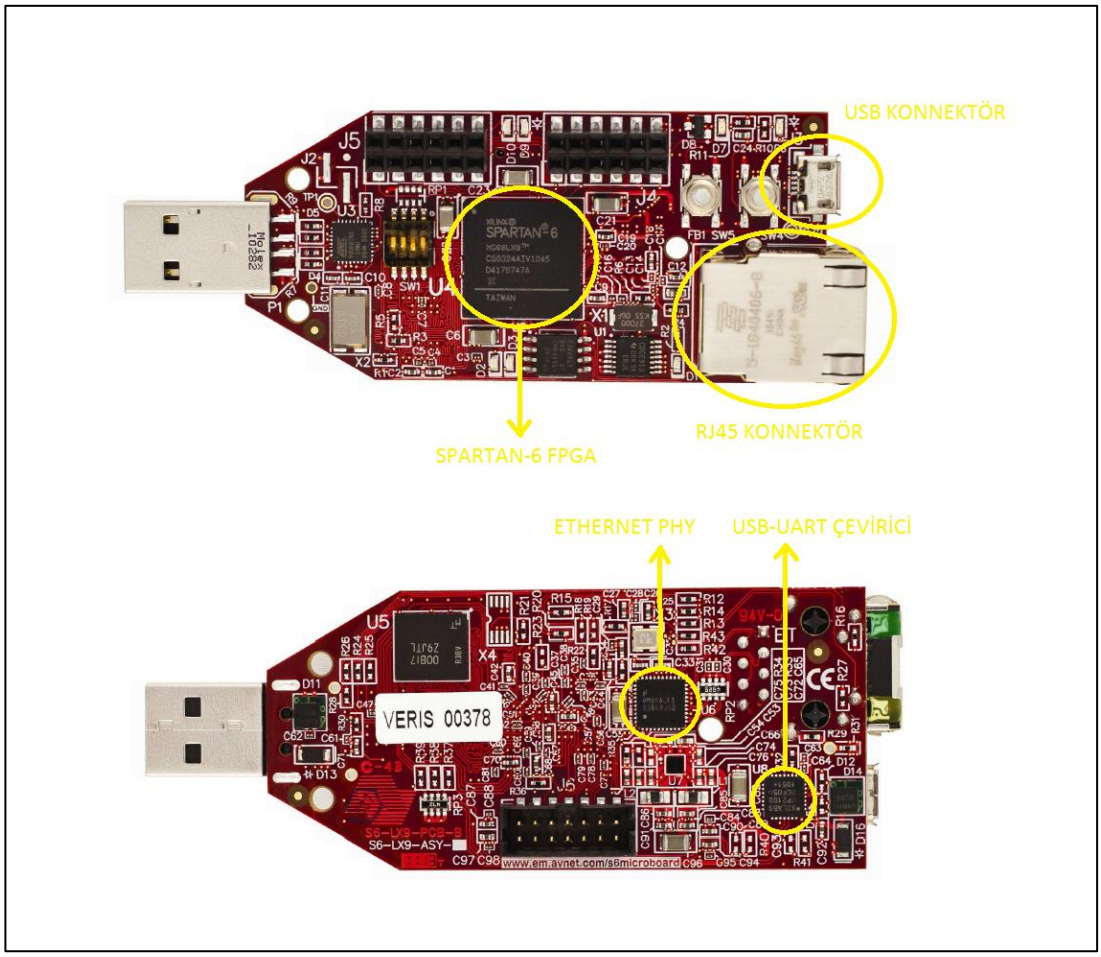
Bu bölümde MAC, IP ve UDP bloklarının her biri detaylı bir şekilde anlatılacaktır. Sistem, donanım tanımlama dili olan VHDL kullanılarak gerçeklenmiştir. Projedeki modüllerin hiçbiri, hazır çekirdek üretmeye yarayan Core Generator yazılımı kullanılarak üretilmemiştir. Bütün kodlar saf VHDL olarak yazılmıştır. Böylece ileride farklı bir FPGA kullanılmak istendiğinde firmanın yazılımına bağımlı kalmadan VHDL kodları aynı şekilde kullanılabilecektir. Kodlar, Xilinx firmasının sağladığı ISE 14.1 yazılımı kullanılarak sentezlenmiş, Xilinx ISIM Simulator yazılımı kullanılarak benzetimleri yapılmış, gene ISE 14.1 kullanılarak gerçeklenmiş ve FPGA'ya gömülebilecek bit uzantılı dosya oluşturulmuş, Xilinx Impact yardımı ile FPGA'ya gömülmüştür.

Platform olarak Avnet firmasının tasarladığı, Xilinx Spartan-6 ailesinden XC6SLX9-2CSG324C tipi FPGA kullanılan Spartan-6 LX9 Microboard geliştirme kiti kullanılmıştır. Kit üzerindeki 10/100 Ethernet PHY ve RJ45 ethernet konnektörü kullanılmıştır. Ayrıca demo amaçlı olarak, kit üzerindeki USB-UART çevirici çipi de bilgisayarla FPGA'nın Uart üzerinden haberleşebilmesi amacı ile kullanılmıştır. Şekil 6.1'de LX9 Microboard'un genel görüntüsü ve üzerindeki çipler gösterilmiştir.

Sistemin test edilmesi aşamasında, bilgisayar üzerinden FPGA kartına Colasoft Packet Builder programı kullanılarak UDP test paketleri atılmış, dönen Uart paketleri Docklight terminal programı kullanılarak gözlemlenmiştir. Aynı şekilde, bilgisayar üzerinden FPGA kartına Docklight terminal programı kullanılarak Uart test paketleri atılmış, dönen UDP paketleri Wireshark programı kullanılarak gözlemlenebilmiştir.

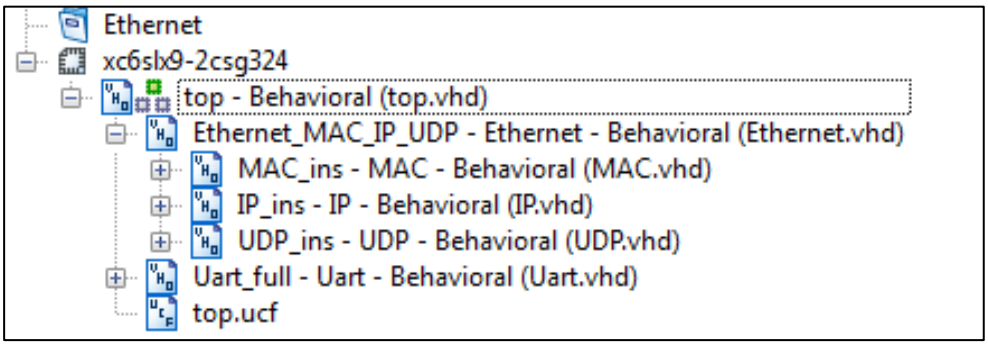
Ayrıca, sistemin performansını test edebilmek amacı ile Microsoft Visual Studio platformu kullanılarak C# dilinde test programları yazılmıştır. Yazılan programlar istenilen sayıda Uart ve UDP paketleri atarak, dönen paket sayısı ile karşılaştırmaktadır.

Son olarak, tasarlanan FPGA kodunun kullanımına örnek olması açısından ev güvenlik uygulaması yapılmıştır. Tasarlanan uygulama bir adet PIR hareket sensörü, bir adet darbe sensörü ve C# kodu içermektedir.

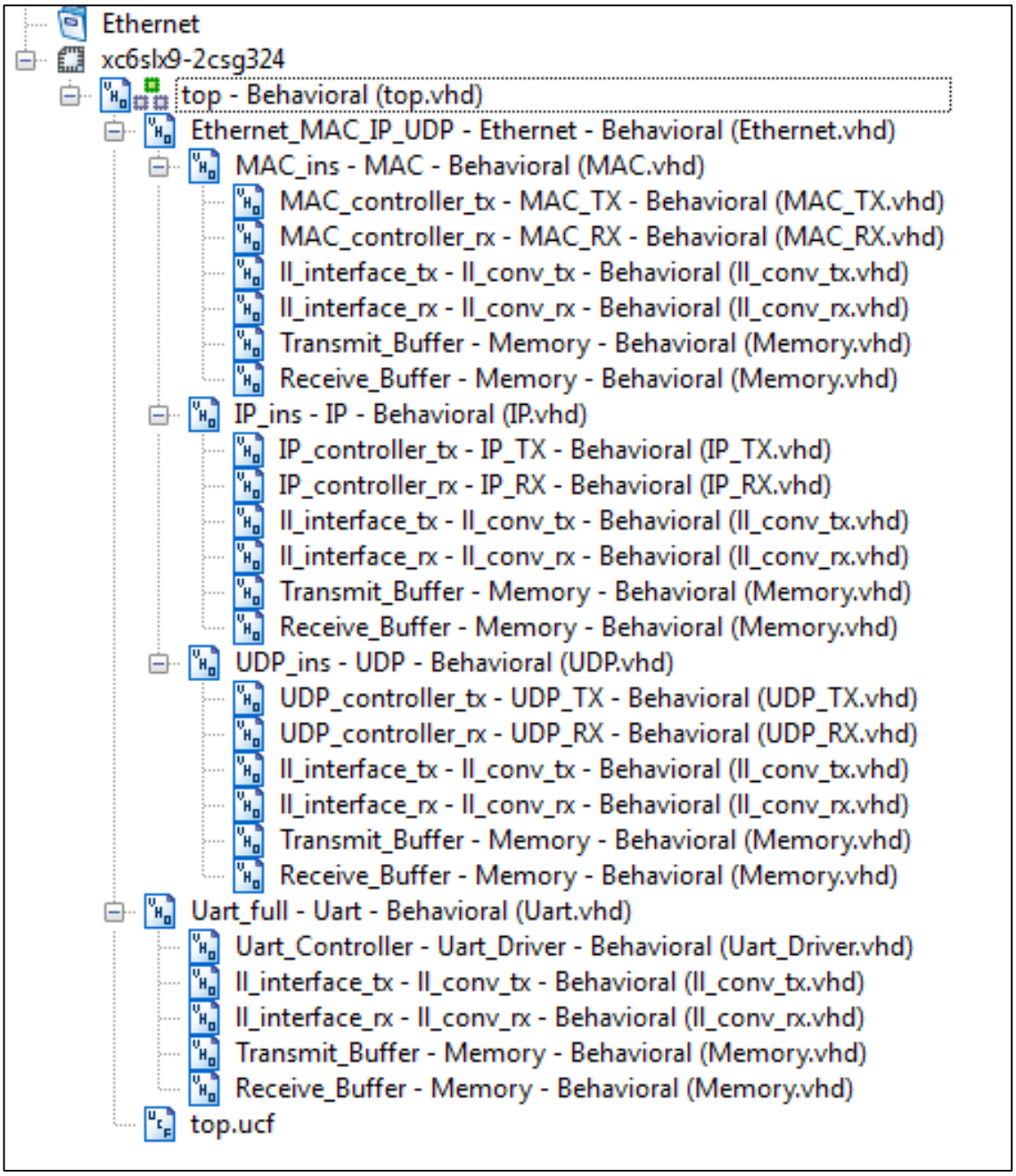


Şekil 6.1 : LX9 Microboard.

6.1 Sistemin Genel Yapısı



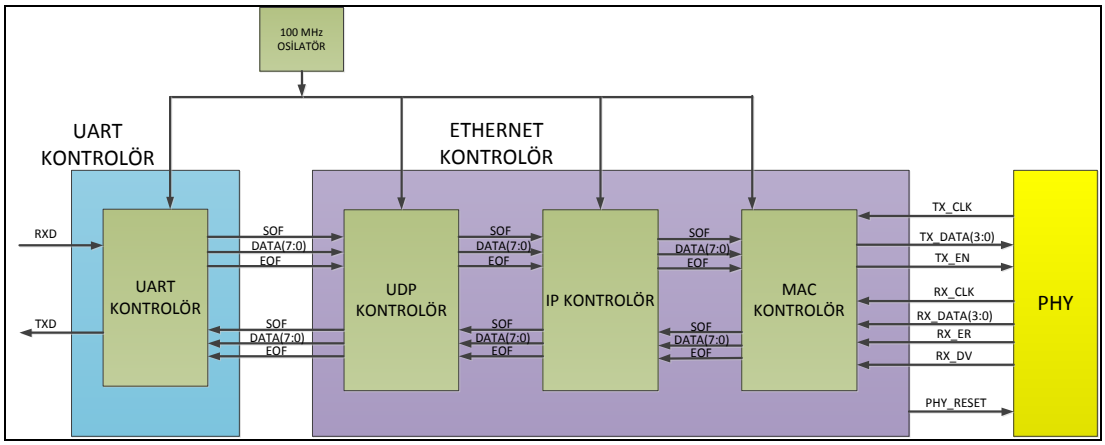
Şekil 6.2 : Sistemdeki Ana Modüllerin Hiyerarşik Yapısı.



Şekil 6.3 : Sistemin Detaylı Hiyerarşik Yapısı.

Şekil 6.2 ve şekil 6.4'de gösterildiği gibi sistem ana modülün altında Ethernet ve UART olmak üzere iki modülden oluşmaktadır. FPGA'ya UART üzerinden gelen paketler ethernet üzerinden gönderilmekte, ethernet üzerinden gelen paketler de UART üzerinden gönderilmektedir.

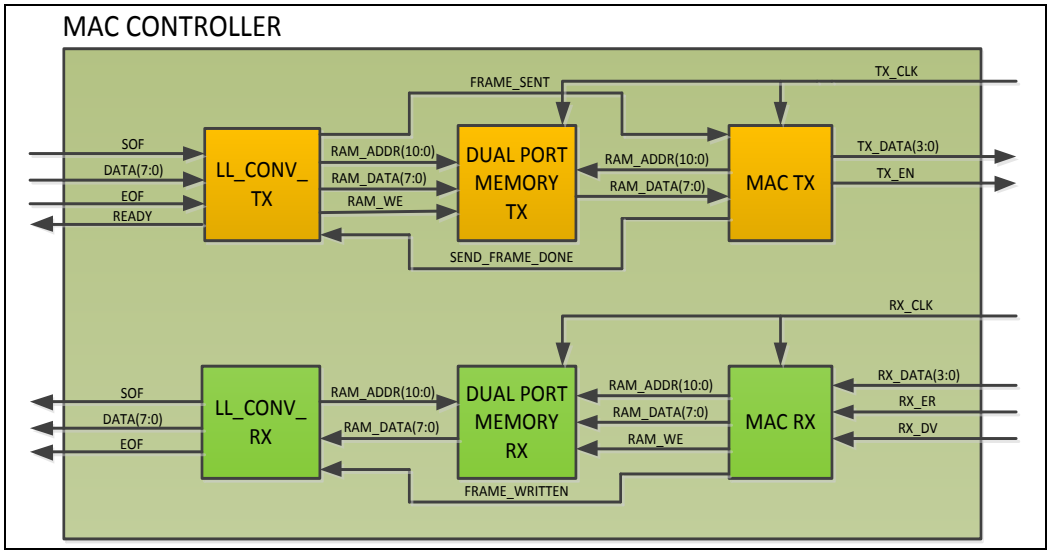
Şekil 6.3 ve şekil 6.4'de görüleceği üzere ethernet modülü MAC, IP ve UDP modüllerinden oluşmaktadır. Her bir blok kendi içinde, alıcı ve gönderici kontrolörü, alıcı ve gönderici belleği, alıcı ve gönderici ara yüz kontrolörü olmak üzere 3 ana bloktan oluşmaktadır. Her bir blok detaylı olarak aşağıda anlatılmaktadır.



Şekil 6.4 : Sistemin Üst Seviye Çizimi.

6.1.1 MAC bloğu

Ethernet paketlerinin oluşturulması, gönderilmesi, alınması ve ayrıştırılmasından sorumludur. Gönderim için üç, alım için üç olmak üzere altı alt modülden oluşmaktadır. Length/Type alanının tip olarak kullanıldığı durumda, 46 bayttan daha az veri içeren paketlerin kaç bayt padding içerdiği bilgisi elde edilemez. Veriden padding bilgisini çıkarıp ham veriyi elde etme işlemi bir üst katman olan IP katmanına bırakılmıştır.



Şekil 6.5 : MAC Bloğunun Üste Seviye Çizimi.

6.1.1.1 MAC_RX ve MAC_TX

MAC_RX modülü, RX_DV sinyali aktif olduktan sonra RMII üzerinden gelen yedi adet 0x55 ve bir adet 0xD5 olan preamble dizisini bekler, dizi hatalı gelirse hata

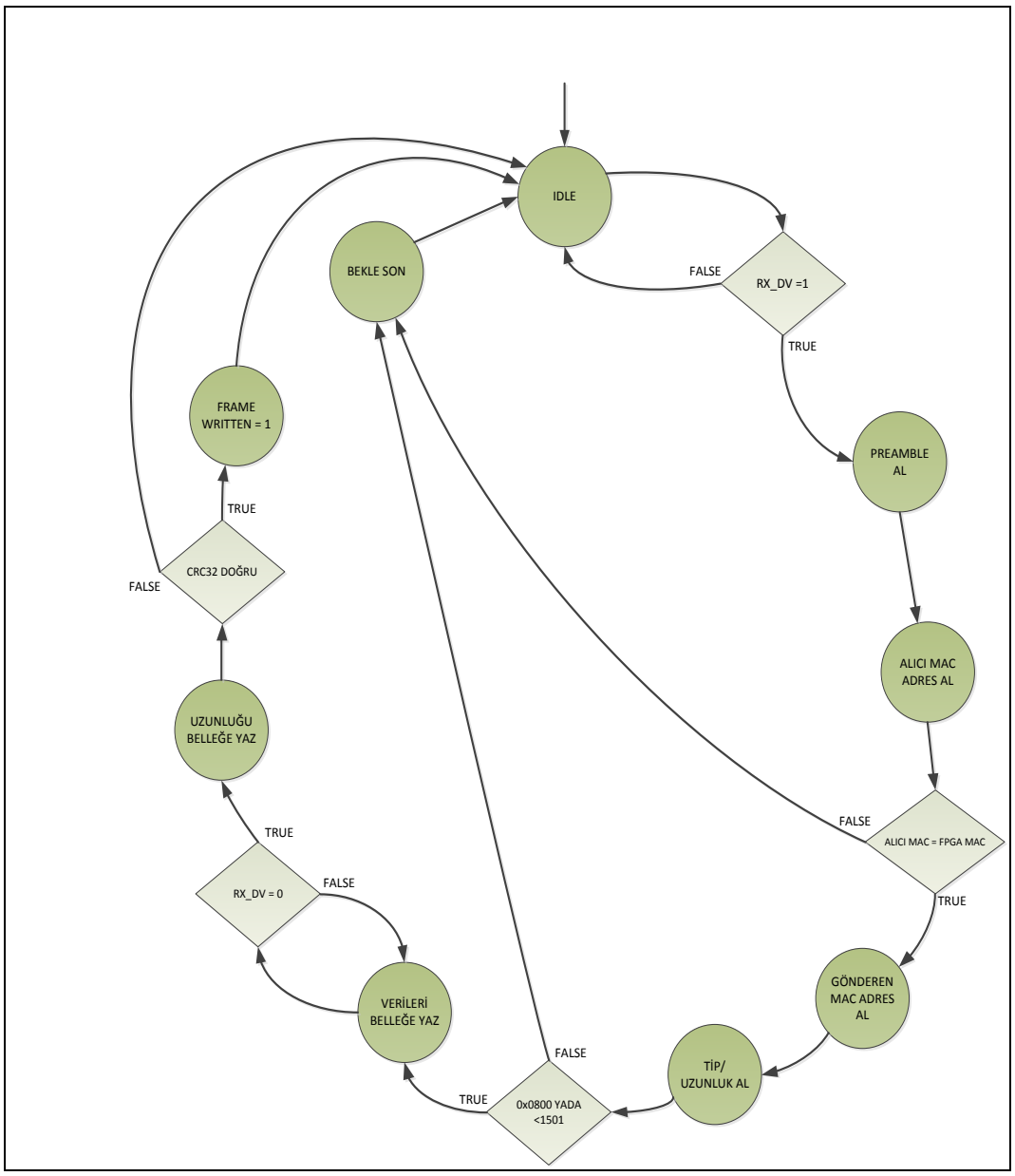
durumuna geçip paket sonunu bekler. Preamble alındıktan sonra gönderici MAC adresini alır ve kendi MAC adresi ile karşılaştırır, paket kendine ait ise işleme devam eder, değil ise paket sonunu bekler. Length/Type alanın tip için kullanıldığı durumda sadece IP paketlerini kabul etmek amacı ile bu alanın 0x0800 olup olmadığını kontrol eder, 0x0800 ise işleme devam eder, değilse bitirmek için paket sonunu bekler. Bu alanın uzunluk olarak kullanıldığı durumda ise maksimum 1500 bayt olan normal paketleri kabul edip jumbo paketleri işleme almamak amacı ile bu alanın 1501'den küçük olup olmadığını kontrol eder, küçükse işleme devam eder, değilse bitirmek için paket sonunu bekler. Ardından gelen verileri hem belleğe yazar hem de CRC32 hesaplar. RX_DV sinyali paket sonunu belirtene kadar gelen veriler belleğe yazılır. Veriler bittikten sonra belleğin son iki gözüne 16 bitlik paket uzunluğunu yazar. Böylece diğer bloklar paketin içeriğine bakmadan paketin uzunluk bilgisini elde edebilirler. Paketin en sonunda alınan CRC32 değeri ile hesaplanan değer karşılaştırılır, pakette bozulma olmadığına karar verilirse frame_written sinyali aktif hale getirilip, MAC_INTERFACE_RX bloğu tetiklenir.

MAC_TX modülü, LL_CONV_TX modülünden gelen frame_sent sinyali ile bellekte paket olduğunu anlar. Öncelikle belleğin son iki gözünden uzunluk bilgisini alır ve TX_EN sinyali ile beraber preamble dizisini RMII hattı üzerinden göndermeye başlar. Daha sonra ETHERNET modülünde tanımlanan alıcı MAC adresini ve gönderen MAC adresini gönderir. Gönderilen paket IP paketi olduğundan Length/Type alanında 0x0800 verisini gönderir. Bellekten okuduğu verileri sırası ile hattan göndermeye başlar. Eğer veri uzunluğu, ethernetin minimum veri uzunluğu olan 46'dan küçükse, eksik veriyi 0 ile doldurarak paket uzunluğunu 46'ya tamamlar. Veri gönderilirken aynı zamanda CRC32 bilgisini de hesaplar. Son olarak CRC32 verisini de gönderir ve TX_EN sinyalini '0' yapar. İki paket arasında 96 bit olan interframe gap süresi kadar bekledikten sonra frame_written sinyalini '1' yapar ve bir sonraki paket için frame_sent sinyalini dinlemeye başlar.

6.1.1.2 MEMORY_RX ve MEMORY_TX

MEMORY_RX ve MEMORY_TX modülleri aynıdır. FPGA'nın içinde bulunan blok bellek kullanılarak oluşturulmuştur. Veri akışı tek yönlü olduğundan, bir porttan sadece yazılırken diğer porttan sadece okunabilen konfigürasyon olan basit çift portlu bellek konfigürasyonu kullanılmıştır. Ethernet paketindeki maksimum veri

boyutu 1500 bayt olduğundan, belleğin boyutu 2048 olarak ayarlanmıştır. Farklı bir FPGA kullanıldığında firmanın yazılımına bağlı kalmamak için, Core Generator kullanılarak oluşturulmak yerine, saf VHDL kodu ile yazılmıştır.

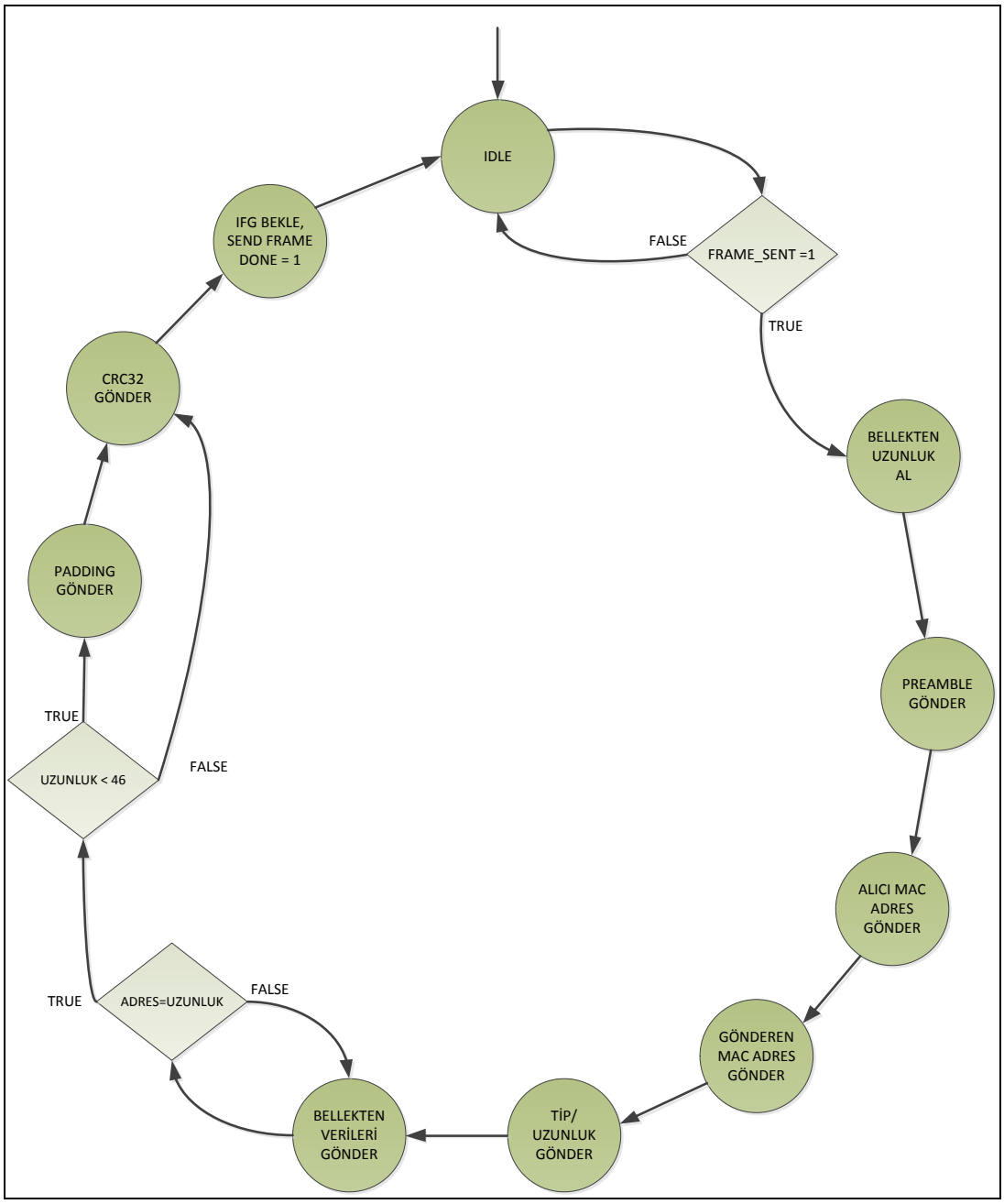


Şekil 6.6 : MAC_RX Bloğunun Basitleştirilmiş Durum Diyagramı.

6.1.1.3 LL_CONV_RX ve LL_CONV_TX

LL_CONV_RX modülünde, modüller arasında veri bloklarını aktarmak için Start of Frame, Data ve End of Frame sinyallerinden oluşan bir ara yüz kullanılmıştır. Gönderilen verinin ilk baytı ile beraber SOF sinyali ‘1’ olmakta, son bayt ile beraber ise EOF sinyali ‘1’ olmaktadır. Böylece modüller arasında veri aktarırken uzunluk

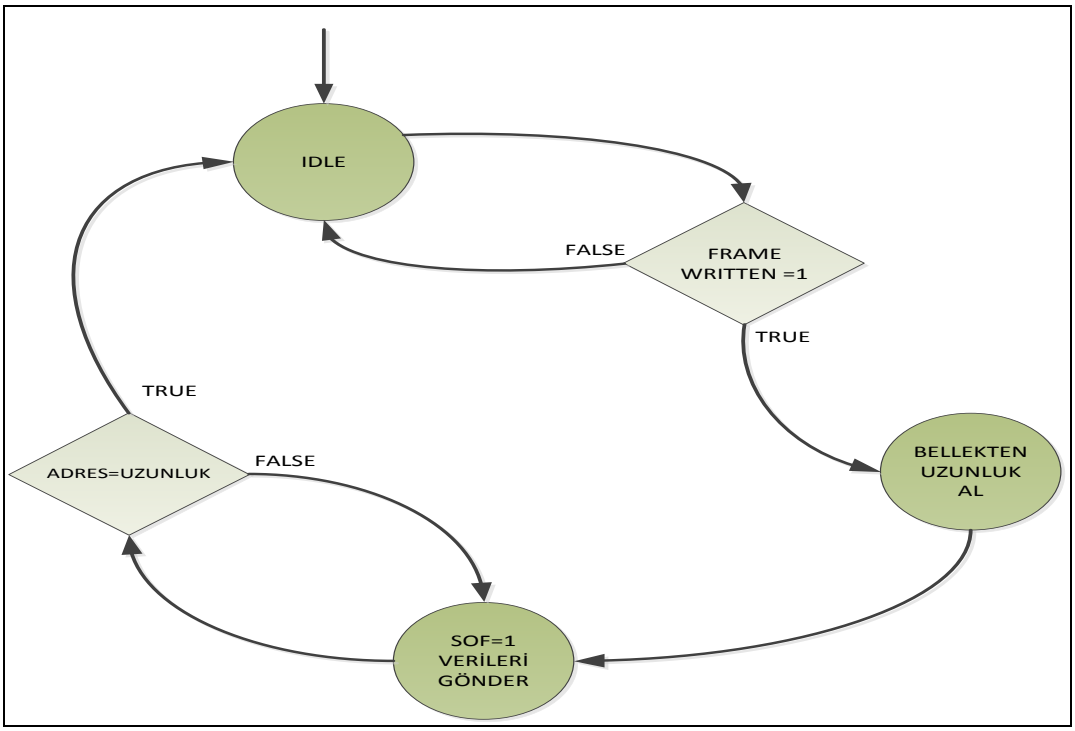
bilgisi ve paket başlangıcı gibi bilgileri sağlama zorunluluğu ortadan kalkmaktadır. Modül, MAC_RX modülünün sağladığı frame_written sinyali ile tetiklendikten sonra belleğin son iki gözünden verinin uzunluk bilgisini alır. Uzunluk sayısındaki kadar veriyi bellekten okur ve yukarıda bahsedilen format ile IP bloğuna aktarır, bir sonraki frame_written sinyalini beklemeye başlar.



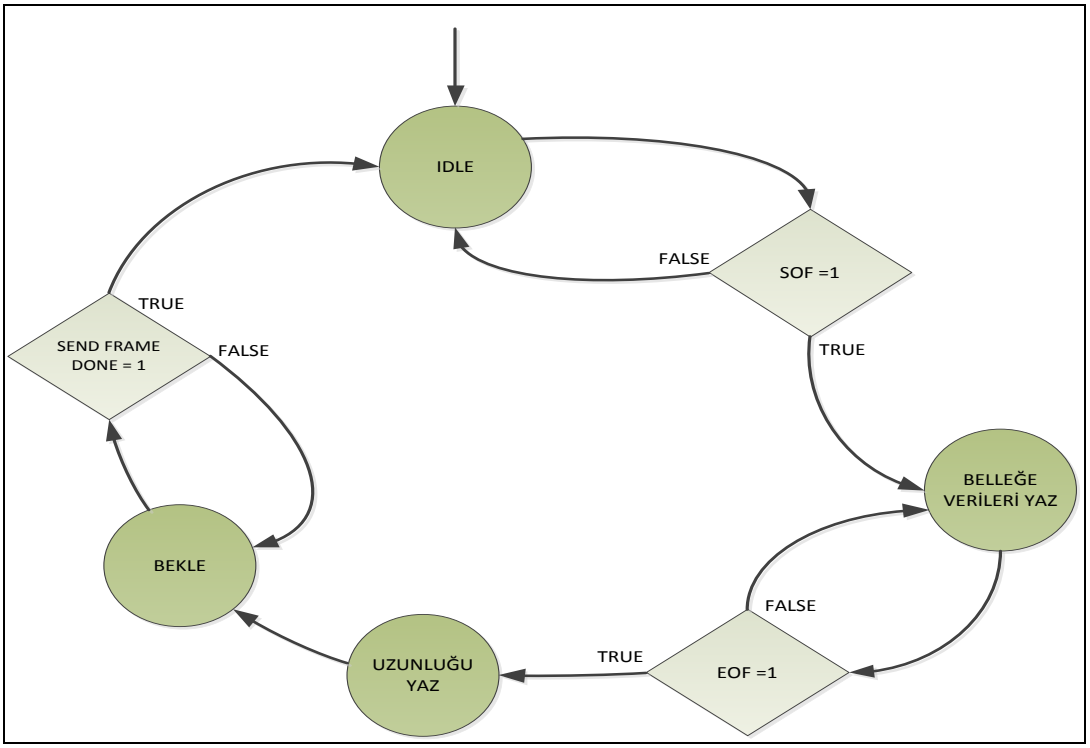
Şekil 6.7 : MAC_TX Bloğunun Basitleştirilmiş Durum Diyagramı.

LL_CONV_TX modülü, IP bloğundan gelen SOF sinyalini dinler ve '1' olduktan sonra gelen verileri belleğe yazmaya başlar, aynı zamanda gelen verinin uzunluk bilgisini hesaplar. Belleğe yazma işlemi EOF sinyali gelene kadar devam eder. EOF

sinyali geldikten sonra belleğin son iki gözüne uzunluk verisi yazılır ve frame_sent sinyali ‘1’ yapılır. MAC_TX modülünden, paket sonunu belirten Send_frame_done sinyali beklenir, sinyal alındıktan sonra yeniden SOF sinyalini dinlemeye başlar.



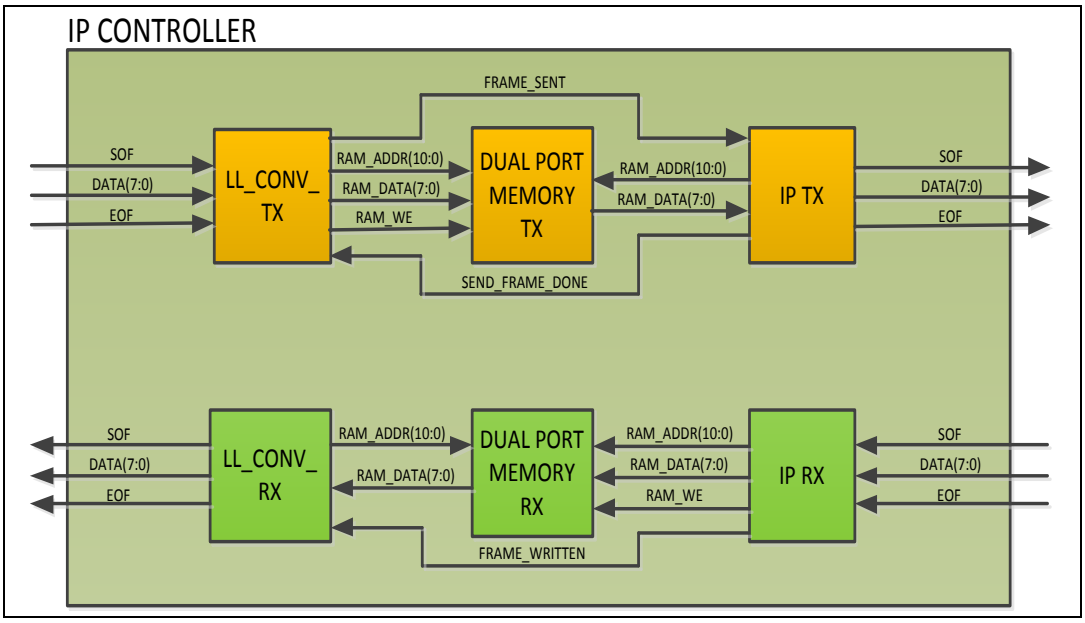
Şekil 6.8 : LL_CONV_RX Bloğunun Basitleştirilmiş Durum Diyagramı.



Şekil 6.9 : LL_CONV_TX Bloğunun Basitleştirilmiş Durum Diyagramı.

6.1.2 IP bloğu

IP paketlerinin oluşturulması, gönderilmesi, alınması ve ayrıştırılmasından sorumludur. Gönderim için üç, alım için üç olmak üzere altı alt modülden oluşmaktadır.



Şekil 6.10 : IP Bloğunun Üste Seviye Çizimi.

6.1.2.1 IP_RX ve IP_TX

IP_RX modülü, MAC modülünden gelen SOF sinyalini dinler ve SOF aktif olduktan sonra gelen verileri işlemeye başlar. Sürüm, başlık uzunluğu ve servis tipi başlıklarını atlar ve işleme almaz. Toplam uzunluk verisini padding miktarını belirlemek amacı ile bir değişkene kaydeder. IP paketlerinin parçalara ayrılma durumu ele alınmadığından tanıtıcı, bayraklar ve parça numarası başlıkları atlanır ve işleme alınmaz. Aynı şekilde yaşam süresi de işleme alınmaz. Protokol bilgisi UDP Checksum hesaplamasında kullanılacağı için bir değişkene kaydedilir ve UDP modülüne iletilir. IP başlık sınaması kısmı, paketin sonunda hesaplanan başlık sınaması ile karşılaştırmak amacı ile kaydedilir. Kaynak IP adresi herhangi bir işleme tabi tutulmaz. Varış IP adresi ise, FPGA'nın kayıtlı IP adresi ile karşılaştırılır, aynı ise işleme devam edilir, değilse paketi bitimek için paket sonu beklenir. EOF sinyali gelene kadar gelen bütün veriler belleğe yazılır. Belleğin son iki gözüne de paketin uzunluk bilgisi yazılır. Uzunluk bilgisi MAC bloğundan gelen verilerin sayılarak elde edilmesi yerine IP paketindeki uzunluk verisi kullanılmıştır. Bunun amacı MAC

paketindeki olası padding verilerini elemektir. Son olarak IP başlık sınaması bilgisi hesaplanır ve alınan başlık sınaması ile karşılaştırılır. Eşit ise frame_written sinyali '1' yapılarak LL_CONV_RX modülüne belleğe paket yazıldığı bilgisi iletilir, eşit değil ise frame_written sinyali çıkarılmaz ve dinleme durumuna geri dönülür.

IP_TX modülü, LL_CONV_TX modülünden gelen frame_sent sinyali ile bellekte paket olduğunu anlar. Öncelikle belleğin son iki gözünden uzunluk bilgisini alır ve bir değişkene kaydeder. IP başlığında kullanılmak üzere bu değerle IP başlık uzunluğu olan 20 sayısını toplar ve başka bir değişkene kaydeder. MAC paketinin tersine IP başlık sınaması IP başlıkları ile beraberinde gönderildiğinden ve hesaplanırken sadece IP başlıkları kullanıldığından, veriler gönderilmeye başlamadan önce IP başlık sınaması hesaplanır. IP verilerini SOF sinyali ile birlikte MAC bloğuna göndermeye başlar. Sürüm olarak 4, başlık uzunluğu olarak 5 ve servis tipi olarak 0 verisi gönderilir. Uzunluk verisi olarak biraz önce hesaplanan uzunluk bilgisi gönderilir. IP paketlerinin parçalara ayrılma durumu ele alınmadığından tanıtıcı, bayraklar ve parça numarası başlıkları 0 olarak gönderilir. Uyumlu olması açısından yaşam süresi 128 olarak gönderilir. Bir üst katmanda UDP kullanılacağından, protokol kısmında 17 sayısı gönderilir. Daha sonra işlemin başında hesaplanan başlık sınaması gönderilir. ETHERNET modülünde tanımlanan sırası ile kaynak IP adresi ile varış IP adresini gönderir. Seçenekler kısmı tanımlı değildir. En son olarak da bellekten uzunluk kadar veri çekilerek MAC bloğuna gönderilir.

6.1.2.2 MEMORY_RX ve MEMORY_TX

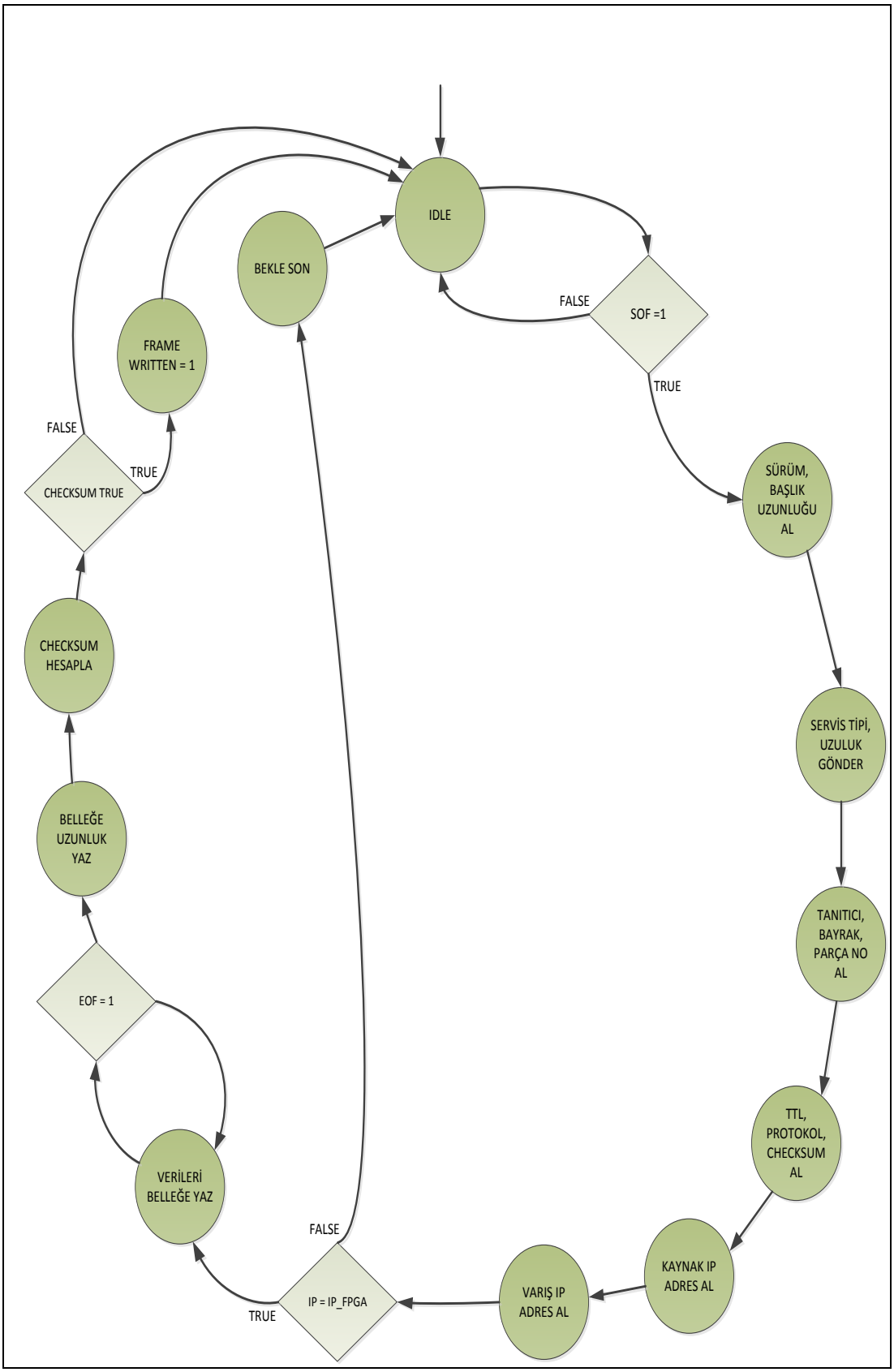
MAC bloğunda kullanılan MEMORY_RX ve MEMORY_TX modülleri ile aynıdır.

6.1.2.3 LL_CONV_RX ve LL_CONV_TX

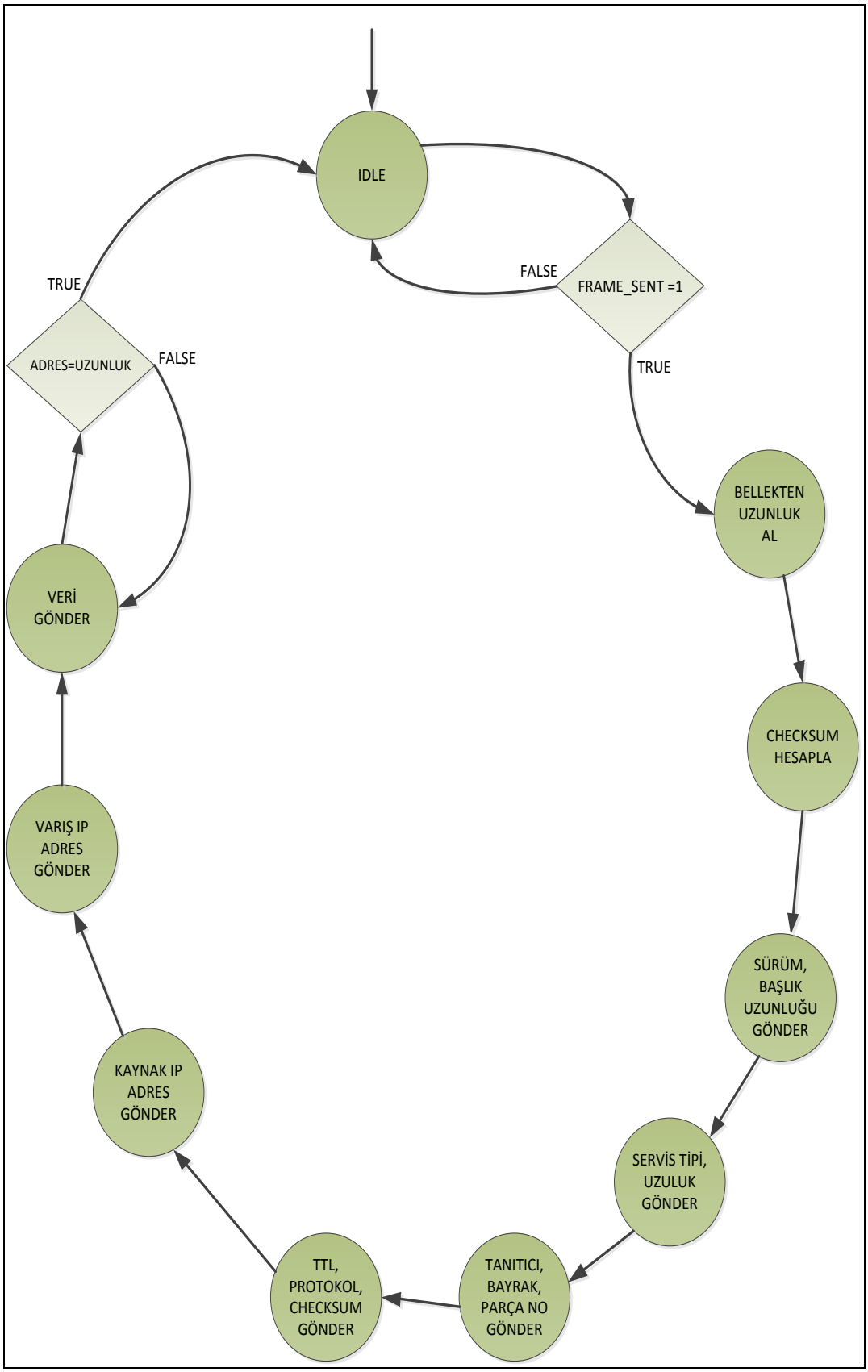
MAC bloğunda kullanılan LL_CONV_RX ve LL_CONV_TX modülleri ile aynıdır.

6.1.3 UDP bloğu

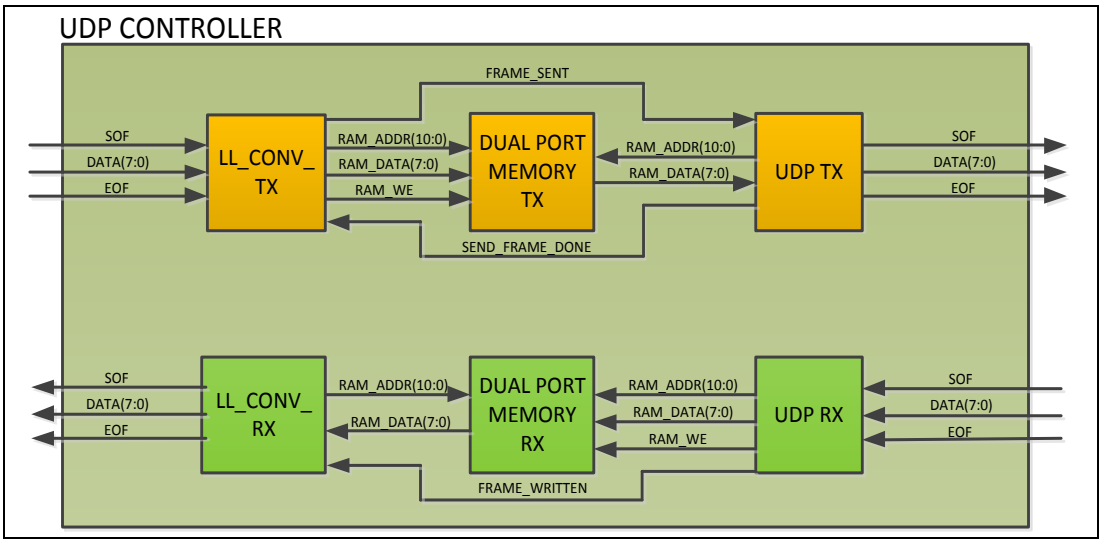
UDP paketlerinin oluşturulması, gönderilmesi, alınması ve ayrıştırılmasından sorumludur. Gönderim için üç, alım için üç olmak üzere altı alt modülden oluşmaktadır.



Şekil 6.11 : IP_RX Bloğunun Basitleştirilmiş Durum Diyagramı.



Şekil 6.12 : IP_TX Bloğunun Basitleştirilmiş Durum Diyagramı.



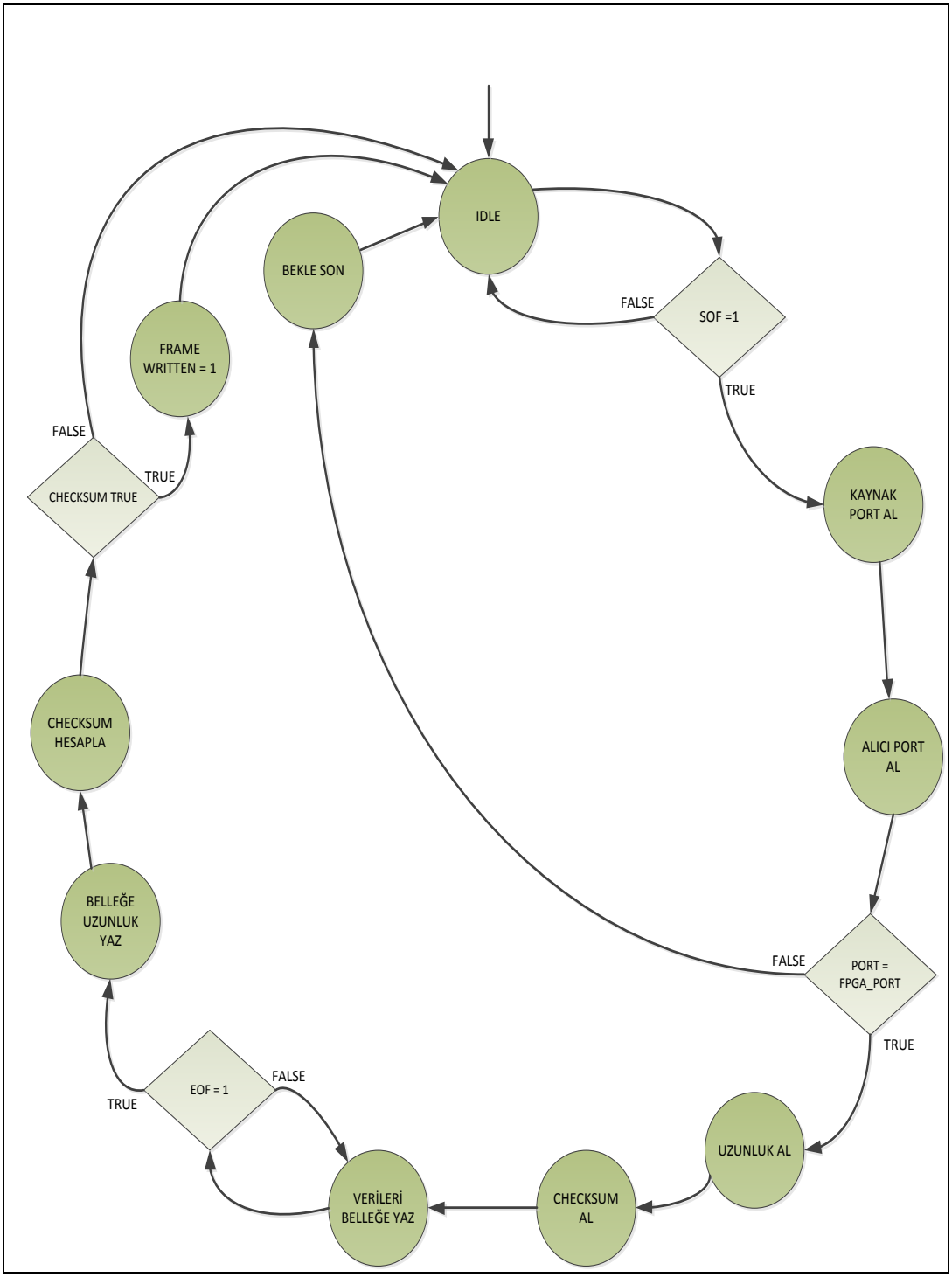
Şekil 6.13 : UDP Bloğunun Üste Seviye Çizimi.

6.1.3.1 UDP_RX ve UDP_TX

UDP_RX modülü, IP modülünden gelen SOF sinyalini dinler ve SOF aktif olduktan sonra gelen verileri işlemeye başlar. Sırası ile kaynak port numarası ve alıcı port numarasını iki farklı değişkene kaydeder. Alıcı port numarası ile FPGA port numarasını karşılaştırır, eşit ise işleme devam eder, değil ise paketi bitirmek için paket sonu beklenir. Uzunluk bilgisi alınır, bir değişkene kaydedilir ve belleğe yazmak amacı ile UDP başlık uzunluğu olan 8 sayısı çıkartılır. Başlık sınaması bilgisi alınır ve bir değişkene kaydedilir. Gelen veriler EOF sinyali gelene kadar belleğe yazılır, aynı zamanda UDP başlık sınaması hesaplanmaya devam eder. Veriler bittikten sonra belleğin son iki gözüne uzunluk bilgisi yazılır. Başlık sınaması hesaplaması bittikten sonra alınan ve hesaplanan başlık sınaması karşılaştırılır, eşit ise frame_written sinyali ‘1’ yapılır ve LL_CONV_RX modülüne belleğe paket yazıldığı bilgisi iletilir, eşit değil ise frame_written sinyali ‘1’ yapılmaz ve dinleme durumuna geri dönülür.

UDP_TX modülü, LL_CONV_TX modülünden gelen frame_sent sinyali ile bellekte paket olduğunu anlar. Öncelikle belleğin son iki gözünden uzunluk bilgisini alır ve bir değişkene kaydeder. UDP başlığında kullanılmak üzere bu değerle UDP başlık uzunluğu olan 8 sayısını toplar ve başka bir değişkene kaydeder. IP paketine benzer şekilde UDP başlık sınaması UDP başlıkları ile beraberinde gönderildiğinden ve hesaplanırken hem başlıklar hem veriler kullanıldığından, öncelikler bellekten veriler okunur ve başlık bilgileri de kullanılarak başlık sınaması oluşturulur. UDP verilerini

SOF sinyali ile birlikte MAC bloğuna göndermeye başlar. Sırası ile kaynak port numarası ile alıcı port numarasını gönderir. Uzunluk verisi olarak hesaplanan uzunluk bilgisi gönderilir. Daha sonra işlemin başında hesaplanan başlık sınaması gönderilir. En son olarak da bellekten uzunluk kadar veri çekilerek IP bloğuna gönderilir.



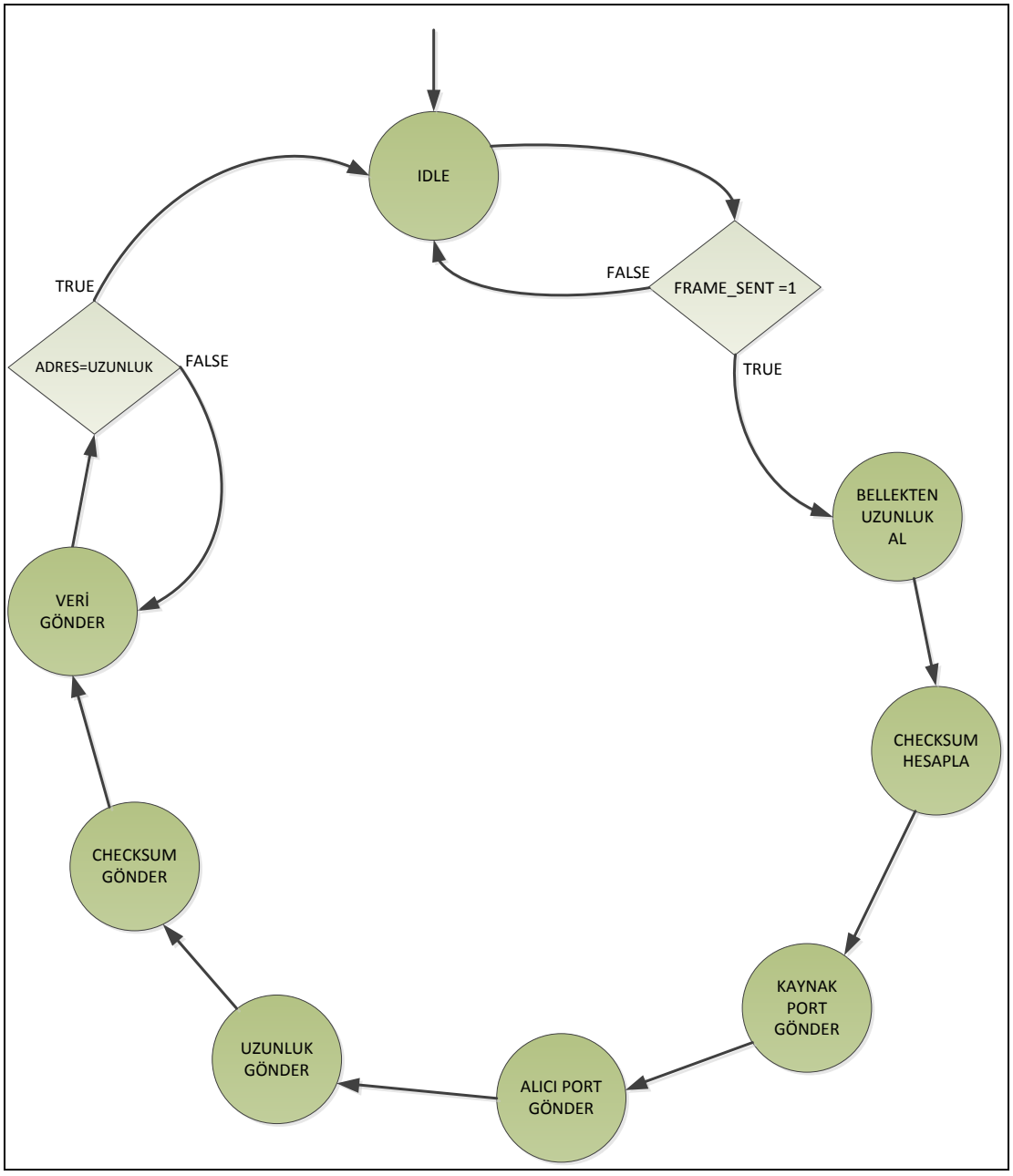
Şekil 6.14 : UDP_RX Bloğunun Basitleştirilmiş Durum Diyagramı.

6.1.3.2 MEMORY_RX ve MEMORY_TX

MAC ve IP bloklarında kullanılan MEMORY_RX ve MEMORY_TX modülleri ile aynıdır.

6.1.3.3 LL_CONV_RX ve LL_CONV_TX

MAC ve IP bloklarında kullanılan LL_CONV_RX ve LL_CONV_TX modülleri ile aynıdır.



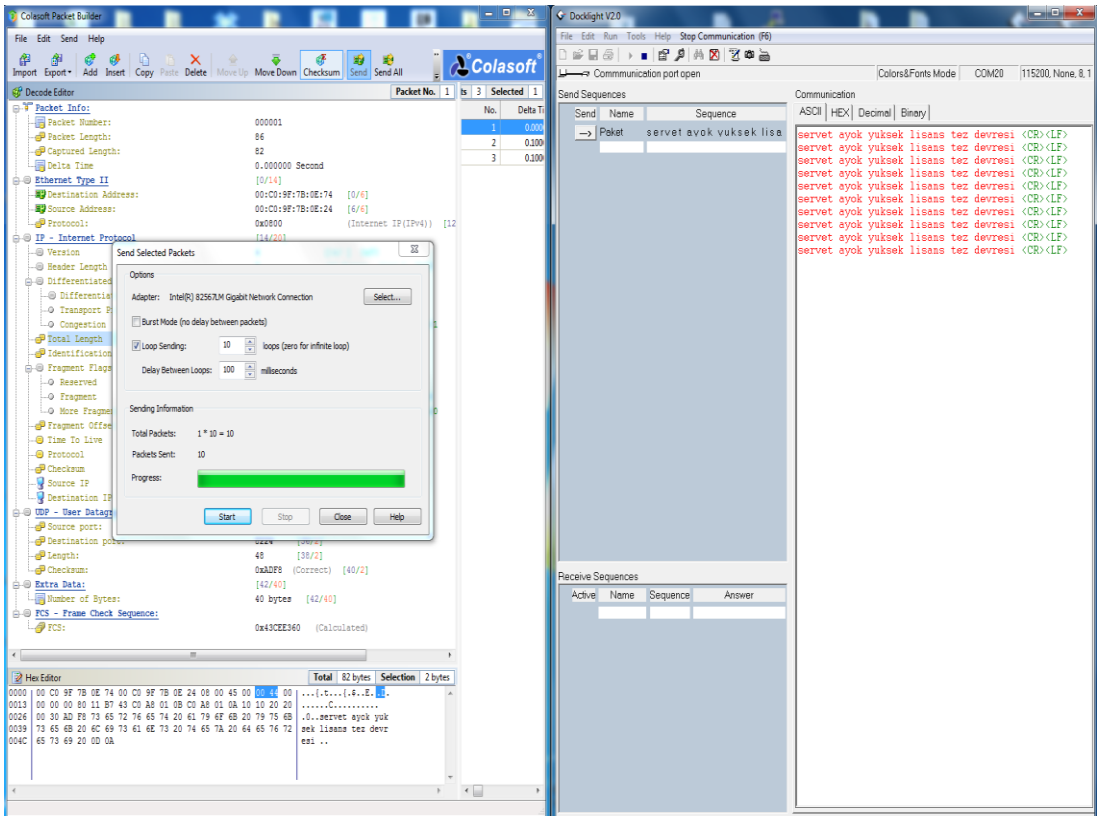
Şekil 6.15 : UDP_TX Bloğunun Basitleştirilmiş Durum Diyagramı.

6.2 Sistemin Test Edilmesi

Sistem ilk olarak fonksiyonel açıdan test edilmiş, daha sonra performans testi yapılmıştır.

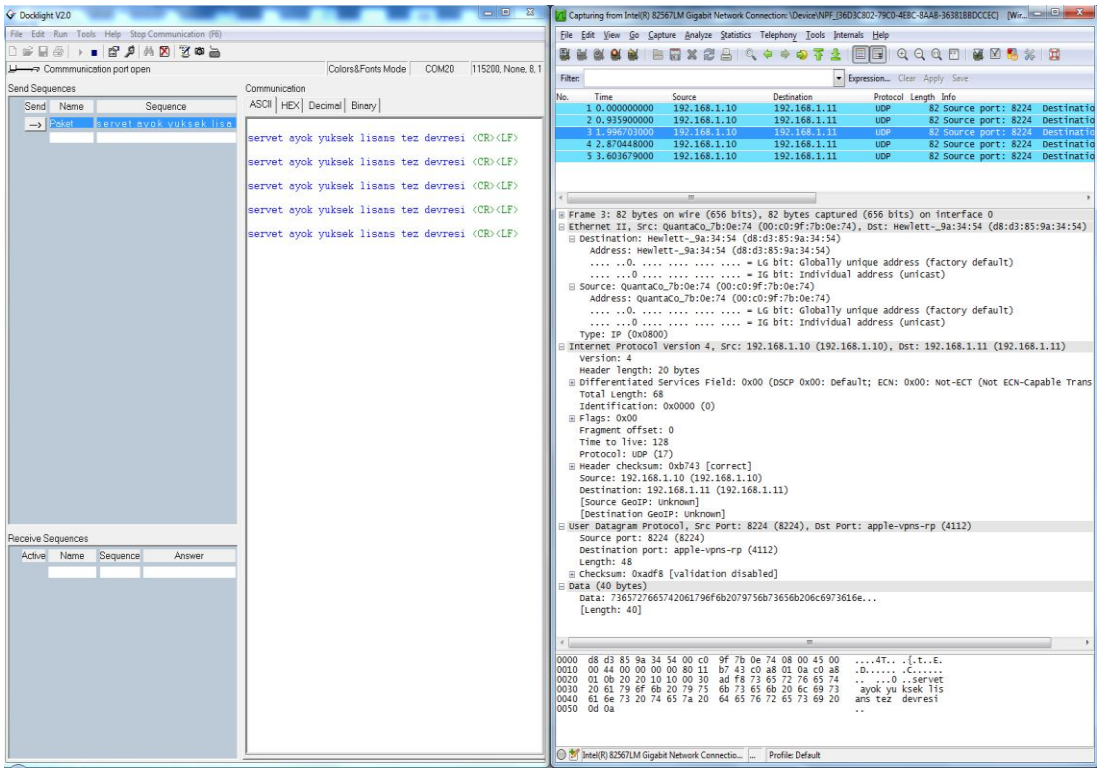
6.2.1 Fonksiyonel test

Fonksiyonel testin ilk aşamasında, bilgisayar üzerinden FPGA kartına Colasoft Packet Builder programı kullanılarak 10 adet UDP test paketi atılmış, dönen Uart paketleri Docklight terminal programı kullanılarak gözlemlenmiştir. Colasoft tarafından gönderilen paketlerin Docklight tarafından hatasız olarak alındığı gözlemlenmiştir. Şekil 6.16'da gönderilen ve alınan paketlere ilişkin ekran görüntüleri verilmiştir.



Şekil 6.16 : UDP'den Gönderilip Uart'dan Alınan Paketler.

Fonksiyonel testin ikinci aşamasında ise, bilgisayar üzerinden FPGA kartına Docklight terminal programı kullanılarak 5 adet Uart test paketi atılmış, dönen UDP paketleri Wireshark programı kullanılarak gözlemlenmiştir. Docklight tarafından gönderilen paketlerin Wireshark tarafından hatasız olarak alındığı gözlemlenmiştir. Şekil 6.17'de gönderilen ve alınan paketlere ilişkin ekran görüntüleri verilmiştir.

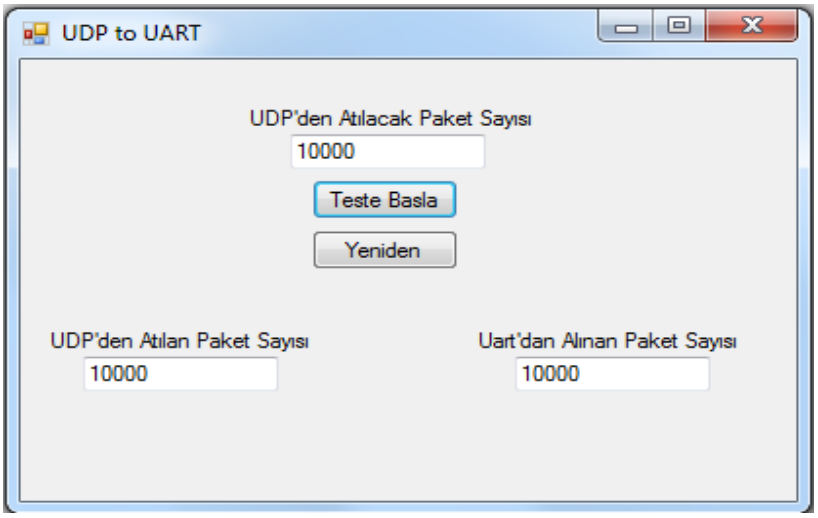


Şekil 6.17 : Uart'dan Gönderilip UDP'den Alınan Paketler.

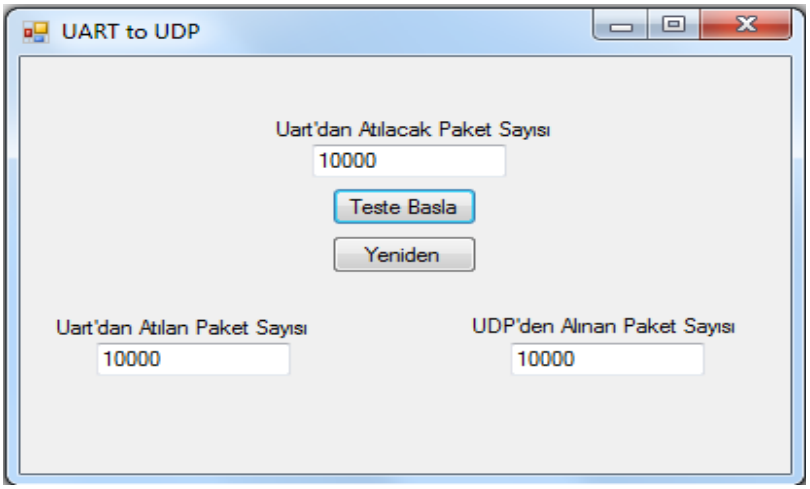
6.2.2 Performans testi

Performans testinin ilk aşamasında, bilgisayar üzerinden 10000 adet UDP paketi atılmış ve dönen Uart paketleri ile karşılaştırılmıştır. Bu amaçla C# dili kullanılarak Visual Studio platformunda bir test kodu yazılmıştır. Test kodu istenilen sayıdaki UDP paketini 20 ms aralıklarla gönderip, gönderilen paket sayısını ekranda göstermektedir. Aynı anda, Uart arayüzünden aldığı paketleri de gönderilen paketler ile karşılaştırdıktan sonra, doğru alınan paket sayısını ekranda göstermektedir. Yapılan testlerde hiçbir paket kaybı ile karşılaşılmamıştır. Şekil 6.18'de programın ekran görüntüsü verilmiştir.

Testin ikinci aşamasında, bilgisayar üzerinden 10000 adet Uart paketi atılmış ve dönen UDP paketleri ile karşılaştırılmıştır. Bu amaçla C# dili kullanılarak Visual Studio platformunda bir test kodu yazılmıştır. Test kodu istenilen sayıdaki Uart paketini 20 ms aralıklarla gönderip, gönderilen paket sayısını ekranda göstermektedir. Aynı anda, UDP arayüzünden aldığı paketleri de gönderilen paketler ile karşılaştırdıktan sonra, doğru alınan paket sayısını ekranda göstermektedir. Yapılan testlerde hiçbir paket kaybı ile karşılaşılmamıştır. Şekil 6.19'de programın ekran görüntüsü verilmiştir.



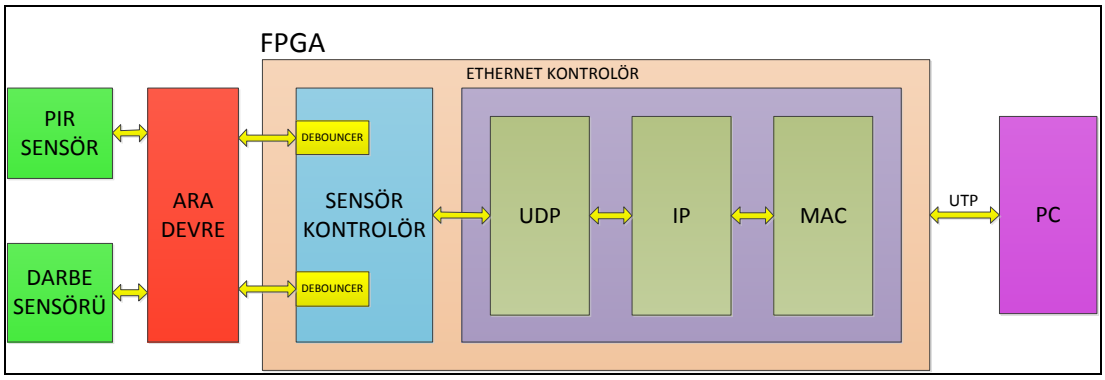
Şekil 6.18 : UDP Test Programının Ekran Görüntüsü.



Şekil 6.19 : Uart Test Programının Ekran Görüntüsü.

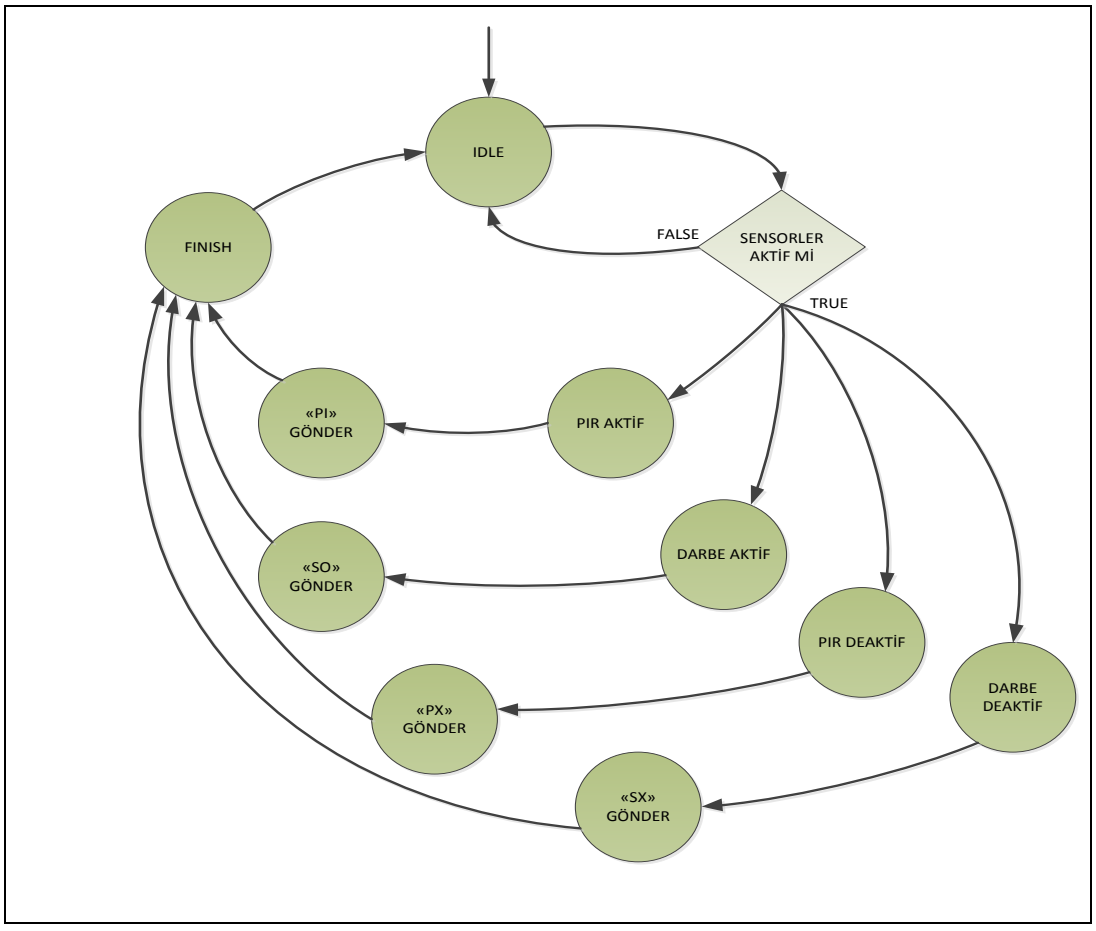
6.3 Örnek Uygulama

Tasarlanan ethernet çekirdeğinin kullanılabileceği örnek alanları göstermek amacı ile ev güvenlik uygulaması geliştirilmiştir. Uygulamada ethernet kontrolör çekirdeğine ek olarak sensör kontrol modülü tasarlanmıştır. Sensör kontrol modülü, bağlı bulunan PIR sensör ve derbe sensörünün herhangi birinden alarm geldiğinde ethernet modülüne belirli bir veri göndermesi için tetikler. Sensörler hem aktif olduğunda hem deaktif olduğunda bilgisayara veri gönderilir. Ayrıca sensörlerden gelen sinyaller durum değiştirme sırasında zıplamalar oluşacağından, sensör kontrol modülü zıplamaları engelleyen Debouncer modüllerini içermektedir. Şekil 6.20'de sistemin üst seviye çizimi gösterilmektedir. Şekil 6.21'de sensör kontrolör modülünün basitleştirilmiş durum diyagramı verilmiştir.

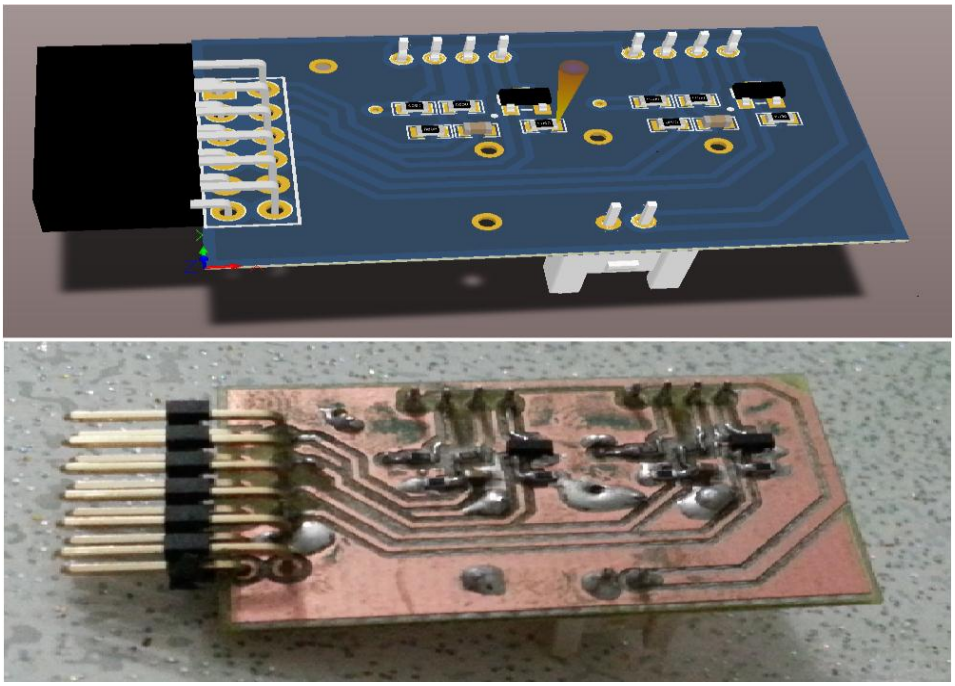


Şekil 6.20 : Güvenlik Uygulama Devresinin Genel Yapısı.

Sensörler 12V ile dışarıdan beslenmekte ve kuru kontak alarm çıkışları vermektedir. Bu nedenle 3.3V sayısal çıkış verebilmesi için uygunlaştırma devresine ihtiyaç vardır. Uygunlaştırma devresi, Altium Designer programı kullanılarak şematik ve PCB dosyaları hazırlanmıştır. Daha sonra PCB prototip makinesi ile üretilip komponentler dizilmiş ve devre çalıştırılmıştır. Şekil 6.22'de Altium Designer tarafından üretilen 3 boyutlu şekil ve basılmış devrenin fotoğrafı bulunmaktadır.

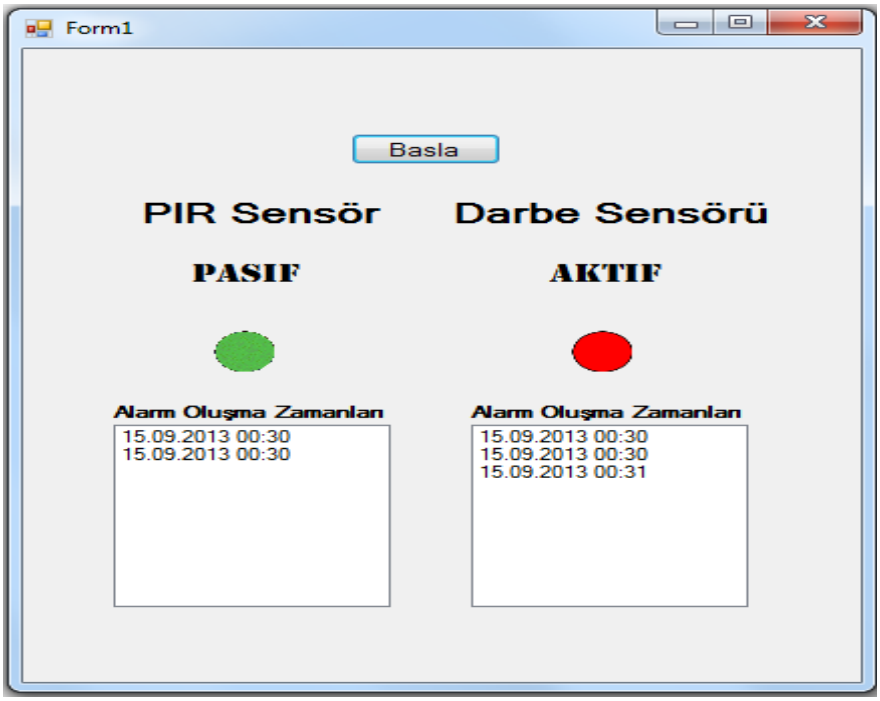


Şekil 6.21 : Sensör Kontrolör Modülünün Basitleştirilmiş Durum Diyagramı.



Şekil 6.22 : 3 Boyutlu Çizim ve Basılmış Devre.

FPGA'nın ethernet üzerinden bilgisayarla haberleşmesini sağlayan ve sensörlerdeki alarm durumlarını gösteren bir C# uygulaması geliştirilmiştir. Şekil 6.23'de gösterildiği üzere PIR ve darbe sensörlerinin durumlarını ve alarm oluşma zamanlarını göstermektedir. Ayrıca alarm oluştuğunda farklı tonlarda bip sesi üretmektedir.



Şekil 6.23 : Güvenlik Uygulaması C# Yazılımı.

7. SONUÇ ve DEĞERLENDİRME

Tasarlanan sistemin hem benzetimleri yapılmış hem de LX9 Microboard üzerinde çalıştırılmıştır. Hem Ethernet hem Uart üzerinden paket alıp vermesinde bir sorunla karşılaşılmamıştır. Çizelge 7.1'de tasarlanan sistemin sadece MAC, IP ve UDP modüllerini içerdiği durumda Spartan-6 XC6SLX9-2CSG324C FPGA üzerinde kullandığı kaynak miktarı verilmiştir. Diğer çalışmalarla karşılaştırma yapmanın daha mantıklı olması adına verilen değerler Uart modülünü içermemektedir.

Çizelge 7.1 Devrenin Harcadığı Kaynak

Kaynak	Değer	%
Slice	834 / 1430	58
Register	1364 / 11440	11
LUT	2535 / 5720	44
RAMB16	6 / 32	18
RAMB8	0 / 64	0

Ayrıca gerçeklenen devredeki her 3 saat işareti için zamanlama değerleri çizelge 7.2'de verilmiştir. PHY çipi tarafından sağlanan MAC_RX_CLK ve MAC_TX_CLK saat işaretleri 25 MHz olduğundan, bu saat işaretleri tarafından beslenen devre bloklarında bir sorun görülmemektedir. Clock_100 saat işareti tarafından beslenen devrenin geri kalan kısımları da maksimum 153.16 MHz frekansında çalışabilmektedir.

Çizelge 7.2 Devrenin Zamanlama Değerleri

Saat İşareti	Değer	Frekans (MHz)
MAC_RX_CLK	6.674 ns	149.83
MAC_TX_CLK	5.605 ns	178.41
Clock_100	6.529 ns	153.16

7.1 Literatürdeki Diğer Tasarımlar ile Karşılaştırılması

Literatürde ethernet MAC, IP ve UDP katmanlarının hepsini, iki tanesini ve sadece bir tanesini içeren çalışmalar mevcuttur. Her çalışma farklı katmanları veya farklı özellikleri gerçeklediğinden, karşılaştırma yapmak için tezdeki çalışmayı aynen

kullanmak uygun değildir. Bu amaçla tezdeki devre, karşılaştırma yapılmak istenen çalışmadaki devreye en benzer hale getirildikten sonra kapladığı alan ve zamanlama değerleri açısından karşılaştırılmıştır. Ayrıca, her devre aynı katmanı gerçeklese bile, istatistik, hata kodları gibi farklı özellikleri içerebileceğinden veya değiştirilebilir IP ve MAC adresi gibi özellikleri sağlayabileceğinden, çekirdekleri tam olarak karşılaştırmanın mümkün olmadığını belirtmekte fayda vardır.

İlk olarak tezdeki devre Andreas Löfgren'in tasarımı ile karşılaştırılmıştır[19]. Tezde tasarlanan sistemi Löfgren'in tasarladığı sistem ile karşılaştırabilmek için devreden Uart modülleri çıkarılmış, sadece MAC, IP ve UDP modülleri ile devre tekrar sentezlenmiştir. Löfgren, ethernet çekirdeğini düşük, orta ve ileri seviye olarak 3 farklı seviyede tasarlamıştır. Bu tezdeki devre [19]'daki düşük seviyeli devre ile karşılaştırılabilecek seviyede olmasına rağmen bazı üstün özellikleri bulunmaktadır. Ayrıca karşılaştırmanın daha sağlıklı olabilmesi için kullanılan FPGA, [19]'da kullanılan Spartan3 XC3S200-4FT256 olarak seçilmiştir. Fakat tasarım bu FPGA'ya sığmadığından dolayı bir üst model olan XC3S400-4FT256 seçilerek sentezlenme işlemi tekrar yapılmıştır. Çizelge 7.3'de tasarımlar karşılaştırılmıştır.

Çizelge 7.3 Devrenin Harcadığı Kaynaklar

Kaynak	Löfgren	Tez
Slice	517	2440
BRAM	3	6
Fmax(MHz)	90.7	89

Tezde tasarlanan devrenin kapladığı alan [19]'daki devreden bir hayli yüksektir. İlk olarak tezde tasarlanan devre modüler yapıdadır, yani MAC, IP ve UDP modülleri kendi başlarına modüler olup bağımsız şekilde kullanılabilmektedir. Maksimum paket boyutu 1518'dir. Full-Duplex çalışabilmektedir. [19]'daki devre modüler değildir ve ayrı ayrı kullanılamaz. Maksimum paket boyutu 256 olup, [19]'da belirtildiği üzere gönderim ve alım kısmı ortak olduğundan full-Duplex çalışamamaktadır.

İkinci olarak, devre Comblock firmasının COM-5401SOF isimli ürünü ile karşılaştırılmıştır[20]. COM-5401SOF ürünü sadece ethernet MAC içerdiğinden, tezdeki devre sadece MAC içerecek şekilde düzenlenmiş ve sentezlenmiştir. Ürünün kullanım kılavuzunda verilen değerler Spartan6 XC6SLX16 kullanılarak

üretildiğinden, tezdeki devre de aynı FPGA kullanılarak sentezlenmiştir. Çizelge 7.4'de tasarımlar karşılaştırılmıştır.

Çizelge 7.4 Devrenin Harcadığı Kaynaklar

Kaynak	COM-5401	Tez
Slice Register	738	483
LUT	1077	1542
BRAM	3	2
Fmax(MHz)	153	203.6

Tezdeki devre daha az Flip-Flop kullanmasına rağmen daha fazla LUT kullanmıştır. Maksimum çalışma frekansı [20]'nin üzerindedir. COM-5401 1Gbps hızında olup, tezdeki devre 100Mbps hızındadır. Çekirdek firmanın sitesinde 750$'a satılmaktadır.

7.2 Yapılabilecek Çalışmalar

Bu tez çalışmasında, kolaylığından dolayı 4. katman olarak UDP seçilmiştir. İleriki çalışmalarda MAC ve IP katmanlarını kullanarak daha güvenli haberleşme sağlayan TCP protokolü geliştirilebilir.

Devre ARP sorgularına cevap vermemektedir. ARP sorgularına cevap vermesi, kendi bünyesinde ARP tablosu tutması ve gerektiğinde ARP sorgusu atması gibi yeni özellikler kazandırılabilir.

Daha hızlı FPGA'lar kullanılarak 1Gbps hızında çalışan MAC/IP/UDP uygulaması geliştirilebilir.

KAYNAKLAR

[1] **Cisco**. (1999). Internetworking Basics [Online]. [Son erişim tarihi 01.05.2013]. Web erişim adresi: http://docwiki.cisco.com/wiki/Internetworking_Basics

[2] **Çakıroğlu, S.** (t.y.). Referans Modelleri [Online]. [Son erişim tarihi 07.04.2013]. Web erişim adresi: http://www.cs.sakarya.edu.tr/sites/muratc/file/Aglar_Bolum2.pdf

[3] **University of South Florida.** (t.y.). Topology [Online]. [Son erişim tarihi 07.04.2013]. Web erişim adresi: http://fcit.usf.edu/network/chap5/chap5.htm

[4] **Spurgeon, C.E.** (2000). Ethernet The Definitive Guide. O'Reilly

[5] **Microchip** (2008). AN1120 Veri Sayfası.[Son erişim tarihi 15.03.2013]. Web erişim adresi: http://ww1.microchip.com/downloads/en/AppNotes/01120a.pdf

[6] **Marvell** (2006). 88E1111 Veri Sayfası.[Son erişim tarihi 15.03.2013]. Web erişim adresi: http://ww1.microchip.com/downloads/en/AppNotes/01120a.pdf

[7] **Xilinx** (2011). UG138 Kullanım Kılavuzu. [Son erişim tarihi 13.02.2013]. Web erişim adresi: http://www.xilinx.com/support/documentation/ip_documentation/tri_mode_eth_mac_ug138.pdf

[8] **IEEE.** (2008). IEEE Std 802.3-2008

[9] **Piscitello, D.M. ve Chapin, A.L.** (1993). Open Systems Networking TCP/IP and OSI. Addison-Wesley

[10] **IETF**. (1989). Communication Layers [Online]. [Son erişim tarihi 13.04.2013]. Web erişim adresi: http://tools.ietf.org/html/rfc1122

[11] **Kachris, C.** (2001). Design and Implementation of a TCP/IP Core for Reconfigurable Logic. Tez. Technical University of Crete

[12] **Lowe, D.** (2005). Networking for Dummies. Wiley Publishing

[13] **Parziale, L., Britt, T.D. ve Davis, C.** (2006). TCP/IP Tutorial and Technical Overview.

[14] **Alachiotis, N., Berger, S.A. ve Stamatakis, A.** (t.y.). A Versatile UDP/IP Based PC FPGA Communication Platform.

[15] **Aydın, A.** (2005). FPGA Yonga Mimarisi ve Kullanımı. Tez. Süleyman Demirel Üniversitesi

[16] **Brown, S. ve Vranesic, Z.** (2005).Fundamentals of Digital Logic with VHDL Design. McGraw Hill

[17] Maxfield, C. (2004). Design Warrior's Guide to FPGAs. Newnes

[18] Xilinx (2013). DS312 Kullanım Kılavuzu. [Son erişim tarihi 06.08.2013]. Web erişim adresi: http://www.xilinx.com/support/documentation/data_sheets/ds312.pdf

[19] Löfgren, A., Lodesten, L. ve Sjöholm, S. (2005) An Analysis of FPGA-based UDP/IP Stack parallelism for Embedded Ethernet Connectivity

[20] Comblock (2012). Com-5401Soft Tri-Mode 10/100/1000 Ethernet MAC VHDL Source Code Overview

ÖZGEÇMİŞ

Ad Soyad: Servet AYOK

Doğum Yeri ve Tarihi: Aydın/1985

Adres: Mimar Sinan Mah. Şair Ruhi Sok. 42/9 Üsküdar

E-Posta: ayok09@hotmail.com

Lisans: İstanbul Teknik Üniversitesi

Elektronik Mühendisliği (2003-2007)

Mesleki Deneyim ve Ödüller:

Beko A.Ş. (2007-2008) – FPGA Üzerinde Görüntü Filtreleme

Pavo A.Ş. (2008-2010) – ANKA İnsansız Hava Aracı Haberleşme Ünitesi Donanım Tasarımı ve FPGA Yazılımı

Turkcell Teknoloji (2011- …) – Donanım Tasarımı