T.C.

KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

GÖRÜNTÜ İŞLEME ALGORİTMALARININ FPGA DONANIMI ÜZERİNDE GERÇEKLENMESİ

ALİ RECAİ ÇELİK

YÜKSEK LİSANS TEZİ

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

KAHRAMANMARAŞ 2013

T.C.
KAHRAMANMARAŞ SÜTÇÜ İMAM ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GÖRÜNTÜ İŞLEME ALGORİTMALARININ FPGA DONANIMI ÜZERİNDE GERÇEKLENMESİ

ALİ RECAİ ÇELİK

Bu tez,
Elektrik-Elektronik Mühendisliği Anabilim Dalında
YÜKSEK LİSANS
derecesi için hazırlanmıştır.

KAHRAMANMARAŞ 2013

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Ali Recai ÇELİK

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

GÖRÜNTÜ İŞLEME ALGORİTMALARININ FPGA DONANIMI ÜZERİNDE GERÇEKLENMESİ

ÖZET

Görüntü işleme uygulamaları; sağlık sektöründe, güvenlik sistemlerinde, robot uygulamalarında, radar ve uydu sistemleri gibi birçok alanda kullanılmaktadır. Coğrafi haritaların çıkarılması, fraktal resim oluşturulması, uçak simülasyonlarının hazırlanması, bilgisayar grafiklerinin ve oyunlarının oluşturulması gibi birçok uygulamada görüntü işleme algoritmaları kullanılmaktadır. Bu çalışmada, çeşitli algoritmalar ve görüntü işleme teknikleri kullanılarak 'Sahada Programlanabilir Kapı Dizileri (FPGA)' donanımı üzerinde bazı uygulamalar gerçekleştirilmiştir. Gerçekleştirilen ilk uygulamada 'difüzyon ile sınırlı tanecik kümeleşme modeli (DLA)' kullanılarak fraktal bir şeklin görüntüsü elde edilmiş ve bu görüntünün piksellerine kırmızı, yeşil ve mavi renklerin atamaları yapılmıştır. İkinci uygulamada; 'elmas-kare algoritması' kullanılarak, farklı yükseklik seviyelerine sahip şekillerden oluşan bir görüntü elde edilmiştir. Kırmızı, yeşil, mavi renklerin ve bu renklerin kombinasyonlarıyla oluşturulan renklerin, görüntüdeki yükseklik seviyelerine atamaları yapılmıştır. Üçüncü uygulamada; görüntüdeki gürültünün azaltılmasında kullanılan ve görüntü işlemenin temel tekniklerinden biri olan 'filtreleme işlemi' gerçekleştirilmiştir. Dördüncü uygulamada ise; 'döndürme matrisi' kullanılarak, 2 boyutlu resimlerin 3 boyutlu olarak görüntülenmesi sağlanmıştır. Uygulamalar için gerekli olan kodlar, donanım tanımlama dillerinden biri olan Verilog HDL dili ile oluşturulmuştur. Uygulamaların sonuçları, FPGA donanımı üzerinde bulunan VGA çıkış birimi aracılığı ile ekranda gösterilmiştir.

IMPLEMENTATION IMAGE PROCESSING ALGORITHMS

ON FPGA HARDWARE

SUMMARY

Image processing algorithms are used in different areas such as health, security, robot applications, radar and satellite systems. Also they can be used in mapping techniques, fraktal picture building, flight simulation, computer graphics and games. In this study, several algorithms and image processing techniques were implemented by using 'Field Programmable Gate Arrays (FPGA)' hardware. In first application, image of the fractal shape was generated by 'Diffusion Limited Aggregation (DLA)' method and red, green, blue colors were assigned to the shape. In second application, an image which consists of some shapes have different heights was generated by using 'diamond-square algorithm'. Red, green, blue and some other colors which can be created by combinations of these colors were assigned to the each height levels on shape. In third application, one of the basic technique of image processing is named filtration, which is used for decreasing the noise on the image, was implemented. In fourth application, two-dimensional image was displayed as three dimensional by using 'rotation matrix'. Necessary codes for these applications were created by Verilog HDL language which is one of the hardware description language. Results of the applications were displayed on the screen through VGA output unit of FPGA hardware.

TEŞEKKÜR

Çalışmalarıma değerli yorum ve önerileri ile katkıda bulunan danışman hocam Doç. Dr. Ahmet ALKAN'a teşekkürü bir borç bilirim.

Öğrenim hayatım boyunca gösterdikleri maddi ve manevi destekler ile bugünlere ulaşmamı sağlayan aileme ve tez çalışmam süresince gösterdiği sabır ve çaba için eşime teşekkür ederim.

Ali Recai ÇELİK

İÇİNDEKİLER

Sayfa No

Sayfa No

SİMGELER VE KISALTMALAR DİZİNİ

mhz: Mega Hertz, frekans birimi

bit: En küçük veri birimi

kbit: Kilo bit

FPGA: Field Programmable Gate Arrays

HDL: Hardware Description Language

VHDL: Very High Speed Integrated Circuit Hardware Description Language

MR: Magnetic Resonance

CMOS: Complementary Metal-Oxide-Semiconductor

IC: Integrated Circuit

DLA: Diffusion Limited Aggregation

RGB: Red-Green-Blue

VGA: Video Graphics Array

DSP: Digital Signal Processing

ASIC: Application Specific Integrated Circuit

LUT: Look-Up Table

MUX: Multiplexer

SRAM: Static Random Access Memory

ASSP: Application Specific Standart Product

I/O: Input/Output

PLD: Programmable Logic Device

SPLD: Simple Programmable Logic Device

CPLD: Complex Programmable Logic Device

USB: Universal Serial Bus

RS-232: Recommended Standart 232

RAM: Random Access Memory

SDRAM: Synchronous Dynamic Random Access Memory

LED: Light Emitting Diode

LCD: Liquid Crystal Display

PLL: Phase-Locked Loop

RTL: Register Transfer Level

XOR: Exclusive OR

ŞEKİLLER DİZİNİ

Sayfa No

Sayfa No

1.GİRİŞ

Görüntü işleme uygulamaları; sağlık sektöründe, güvenlik sistemlerinde, robotik uygulamalarda, radar ve uydu sistemleri gibi birçok alanda kullanılmaktadır. Coğrafi haritaların çıkarılması, fraktal resim oluşturulması, uçak simülasyonlarının hazırlanması, bilgisayar grafiklerinin ve oyunlarının oluşturulması gibi birçok uygulamada görüntü işleme algoritmaları kullanılmaktadır. Üzerinde çeşitli işlemler gerçekleştirilecek olan görüntü, kamera aracılığı ile alınan bir video veya fotoğraf görüntüsü olabileceği gibi; kullanıcılar tarafından kırmızı, yeşil, mavi ve bu renklerin birleşimleriyle elde edilen bir görüntü de olabilir. Bu çalışmada ikinci yöntem tercih edilmiş ve görüntüler algoritmalar aracılığı ile oluşturulmuştur. Oluşturulan görüntüler üzerinde filtreleme ve döndürme işlemleri gerçekleştirilmiştir.

Tasarımcılar tarafından programlanabilme özelliğine sahip olarak üretilmiş olan devrelere 'programlanabilir mantık devreleri' adı verilir. 'Sahada programlanabilir kapı dizileri (FPGA)' de bu amaçla üretilen devrelerden biridir. FPGA'ler, sayısal devre oluşturabilmek için gerekli olan mantık kapılarından oluşurlar. Tasarımcılar, istedikleri fonksiyon ve görevleri gerçekleştirmek amacıyla bu kapı bloklarını programlayabilirler. Bu işlem, Verilog veya VHDL gibi donanım tasarlama dilleri kullanılarak gerçekleştirilir. FPGA kullanılarak yapılan tasarımlar, değiştirilerek tekrar yüklenip çalıştırılabilir. Yani aynı FPGA üzerinde defalarca kod yazma ve silme işlemi yapılabilir. Bu durum tasarımcılar için büyük kolaylık sağlamaktadır [1]. FPGA devrelerinin hızlı işlem yapabilme, içine çoğullayıcı gömülebilme, aritmetik işlemleri başarıyla gerçekleştirebilme gibi özellikleri ve paralel işlem yapabilme yeteneği vardır. Paralel işlem yapabilme yeteneği sayesinde FPGA'ler, aynı anda birden fazla görevi yerine getirebilirler. Bu nedenle yüksek hız gerektiren görüntü işleme uygulamalarında FPGA kullanımı sıklıkla tercih edilmektedir. Mesela, (Rodirguez ve ark., 2002) FPGA üzerinde gerçek zamanlı medyan filtresi tasarlayıp çalıştırarak, üretim merkezlerindeki hataların görsel olarak belirlenmesini sağlayan bir sistem tasarlamışlardır [2]. (Chun He ve ark., 2009) ise insan yüzü tespiti ile ilgili çalışma yapmış; yüz bulma ve tanıma algoritmalarını FPGA ile gerçeklemeyi başarmışlardır [3]. (Christe ve ark., 2011) MR cihazından alınan beyin görüntülerinde bulunan tümörlerin tespiti için, Xilinx marka bir FPGA geliştirme kartından faydalanmışlardır [4]. Sayısal sistem tasarımlarında sık kullanılan FPGA'lerin popülaritesi ülkemizde de son yıllarda artmaya başlamıştır.

Örneğin; Caner (2006), yüksek lisans tezinde plaka yerini tespit eden ve plaka içeriğini tanıyan bir sistemi FPGA kullanarak gerçekleştirmiştir [5]. Sarı (2006), yüksek lisans çalışması olarak görüntü yakalayıp işleyen bir sistemi FPGA ile tasarlamıştır [6]. Yeniçeri (2007) ise, FPGA tabanlı gerçek zamanlı görüntü işleme çalışmalarının temel gereksinimi olan görüntünün yakalanması problemine, sensör kartı donanımı tasarlayarak çözüm sunmuş ve CMOS görüntü sensörü ile FPGA kartı kullanarak sayısal fotoğraf makinesi gerçeklemeyi başarmıştır [7].

Özcan (2009), FPGA tabanlı gerçek zamanlı görüntü işleme çalışması yapmıştır. Bu çalışmasında görüntü üzerinde gerçek zamanlı konvolüsyon işlemi gerçekleştirmiştir [8]. Taşcı (2011), FPGA kontrollü robotik göz oluşturarak, robotik sistemler için gerçek zamanlı bir görüntü işleme ve tanıma sistemi geliştirmiş, görüntü üzerinde uygulanacak olan tüm işlemleri FPGA kontrolü ile yapmıştır [9]. Özçelik (2012) ise FPGA üzerinde gerçek zamanlı olarak görüntüde renk değiştirme, kırmızı renkli nesne takibi, ten rengi tanıma, sobel filtresinin görüntüye uygulanması gibi bir takım görüntü işleme uygulamalarını gerçekleştirmiştir [10].

Bu çalışmada ise, çeşitli algoritmalar ve görüntü işleme teknikleri kullanılarak FPGA donanımı üzerinde bazı uygulamalar gerçekleştirilmiştir. Donanım olarak Altera firmasının üretmiş olduğu DE2 model FPGA platformu kullanılmıştır. Bu donanımda Cyclone II adı verilen kart bulunmaktadır. Gerçekleştirilen ilk uygulamada; 'difüzyon ile sınırlı tanecik kümeleşme modeli (DLA)' kullanılarak fraktal bir şeklin görüntüsü elde edilmiş ve bu görüntünün piksellerine kırmızı, yeşil ve mavi renklerin atamaları yapılmıştır. Bu uygulamanın gerçekleşmesi için gerekli olan tanecik kümeleşme algoritması ve rastgele sayı üretme algoritması Verilog programlama dili ile gerçeklenmiştir. Uygulamada rastgele oluşturulan fraktal görüntünün, farklı renk tonlarıyla renklendirilmesi sağlanmıştır. İkinci uygulamada; elmas-kare algoritması Verilog dili ile gerçeklenerek farklı yükseklik seviyelerine sahip şekillerden oluşan bir görüntü elde edilmiştir. RGB renk uzayına göre üç temel renk olan kırmızı, yeşil, mavi renklerin ve bu renklerin kombinasyonlarıyla oluşturulan renklerin, görüntüdeki yükseklik seviyelerine atamaları yapılmıştır. Böylece çeşitli renk ve yükseklik seviyelerinden oluşan bir resmin kuşbakışı görünümü elde edilmiştir. Üçüncü uygulamada; görüntüdeki gürültünün yani bozucu etkinin azaltılmasında kullanılan ve görüntü işlemenin temel tekniklerinden biri olan filtreleme işlemi FPGA donanımı ile gerçekleştirilmiştir.

Dördüncü uygulamada ise; 'döndürme matrisi' kullanılarak, 2 boyutlu resmin 3 boyutlu olarak görüntülenmesi sağlanmıştır. Bahsedilen tekniklerin ve algoritmaların uygulanması neticesinde oluşan görüntüler, FPGA donanımı üzerinde bulunan VGA çıkış birimi aracılığıyla ekranda görüntülenmiştir.

Bu çalışmanın giriş bölümünü takip eden ikinci bölümünde, FPGA'lerin özelliklerinden ve avantajlarından bahsedilmiştir. Daha sonra görüntü işleme ile ilgili temel bilgiler verilerek, görüntü işlemenin teknikleri ve kullanım alanları belirtilmiştir. Ayrıca fraktal geometri konusu incelenerek, bazı fraktal resim örneklerine yer verilmiştir.

Üçüncü bölümde ise, uygulamalarda kullanılan FPGA platformundan ve sonuçların görüntülendiği ekranın özelliklerinden bahsedilmiştir. Donanımlar ile ilgili olan temel bilgilerin verilmesinden sonra, FPGA üzerinde uygulama gerçekleştirebilmek için gerekli olan Verilog HDL programlama dili kısaca anlatılmıştır. Hazırlanan program kodunun derlenmesini ve donanıma aktarılmasını sağlayan Quartus II yazılımı hakkında bilgi verilmiştir.

Dördüncü bölümde, gerçekleştirilen uygulamaların aşamaları gösterilmiş, uygulamalar için gerekli olan algoritmalar detaylı olarak açıklanmıştır. Görüntünün filtrelenmesinde kullanılan Gauss filtresi incelenmiş ve görüntüyü 3 boyutlu duruma getirmek için gerekli olan sistem anlatılmıştır.

Sonuç bölümünde ise; çalışmadan elde edilen sonuçların ve görüntü işleme uygulamalarında FPGA kullanımının değerlendirilmesi yapılmıştır. Çalışmada gerçekleştirilen dört uygulamada kullanılan FPGA donanımının performansı değerlendirilmiştir.

2.GENEL BİLGİLER

Bu bölümde öncelikle ‘sahada programlanabilir kapı dizileri’ hakkında detaylı bilgiler verilmiş, daha sonra görüntü işlemenin tekniklerinden ve kullanım alanlarından bahsedilmiştir. Son olarak fraktal geometrinin temelleri anlatılmış ve fraktallar ile ilgili örnek resimler sunulmuştur.

2.1. Sahada Programlanabilir Kapı Dizileri (FPGA)

2.1.1. FPGA’in tanımı

Programlanabilir mantık devreleri, mantık kapılarının ve bellek hücrelerinin birbirlerine bağlanması ile oluşturulan devrelerdir. Mantık kapılarının gerçekleştirdiği fonksiyonların tanımlanmasında, kontrolünde ve birbirleriyle olan giriş-çıkış ilişkilerinin kayıtlı tutulmasında bellek hücreleri kullanılır. Farklı mimarilerde tasarlanıp üretilen çok sayıda ürün, mantık olarak aynı temel prensiplerle çalışır. FPGA’ler de bu alanda üretilmiş olan donanımlardan biridir. FPGA’ler, ihtiyaç duyulan mantıksal fonksiyonların tasarımcı tarafından oluşturulabilmesi amacıyla sahada programlanabilir olarak üretilmişlerdir. FPGA’in tanımında kullanılan ‘sahada’ kelimesi, bu aygıtların sahada yani üretimden sonra kullanıcıların kendileri tarafından programlandığını ifade eder [11].

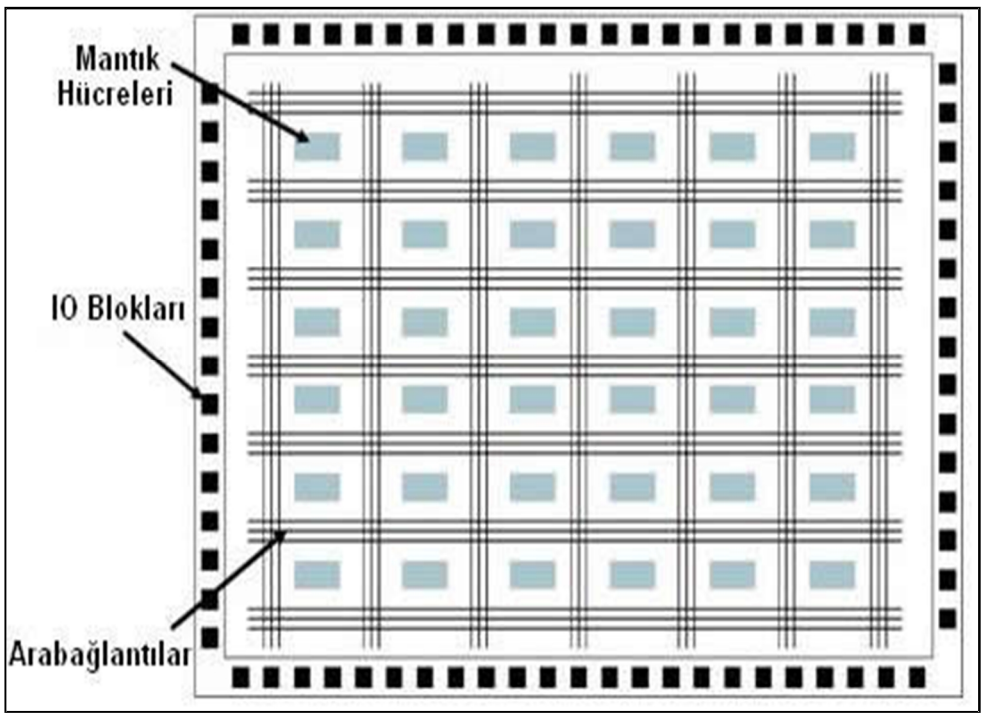
FPGA’ler, üretimden sonra istenilen fonksiyona göre, donanım yapısı kullanıcı tarafından değiştirilebilen entegre devreler olarak tanımlanabilir. FPGA, içindeki transistörleri birbirinden bağımsız ve serbest olarak üretilmiş ham bir entegre olarak düşünülebilir. Kullanıcının belirlediği fonksiyona göre, FPGA içindeki transistörler birbirlerine bağlanır ve istenilen uygulama gerçekleştirilir.

FPGA donanımlarında, sayısal işaret işleyici (DSP)’lerin esnek programlanabilen yapısı ile uygulamaya özgü tümleşik devre (ASIC)’ lerin yüksek performans özelliği birleşmiştir. Bir FPGA, binlerce mantık elemanından ve küçük boyutta bir rastgele erişimli bellekten oluşur. Tüm bu birimler FPGA içerisinde birbirine bağlıdır. Tasarımcı bu mantık ve hafıza birimlerini istediği şekilde programlar. FPGA’ler teorik olarak sınırsız sayıda yeniden programlanmaya izin vermektedir [12].

2.1.2. FPGA'in yapısı

FPGA cihazı; ayarlanabilir bir mantık blok dizisi, bu dizinin çevresinde halka oluşturan giriş-çıkış birimleri ve bütün bu birimleri bağlayan programlanabilir ara bağlantılardan oluşur. Mantık blokları, kullanıcının amacına göre ayarlanabilen donanım bloklarıdır. Bu blokların programlanmasıyla, sayısal devre donanımını oluşturacak elemanlardan meydana gelen birim hücreler oluşur. Birim hücreler kendi aralarında bağlantılara sahiptirler. Giriş-çıkış birimleri, FPGA ile haberleşmeyi sağlayan birimlerdir. Programlanabilir bağlantılar ise, mantık blokları arasındaki bağlantılardır. Bu bağlantılar, birden fazla öbek kullanılarak gerçekleştirilen donanımsal devrelerde, devre elemanlarının birbirleriyle olan bağlantısını sağlarlar [13]. Bir FPGA yapısı Şekil 2.1'de gösterilmektedir.

Mantık bloklarını oluşturan birim hücrelere mantık hücreleri adı verilir. Bir mantık hücresi; 1 adet bakma tablosu (LUT), 1 adet D flip-flop ve 1 adet 2:1 çoğullayıcıdan (MUX) oluşur. Birim hücre mimarisine gore MUX tabanlı ve LUT tabanlı FPGA'lar vardır. Karmaşık donanımları tasarlamak için LUT tabanlı mimariler daha uygundur. LUT mimarisinin gereği olarak bakma tablosunu oluşturmak için SRAM'ler kullanılır. Bu durum esnek hafıza kullanımına imkan verir. FPGA üretiminde söz sahibi olan firmalardan Xilinx ve Altera firmaları, üretimlerinde LUT tabanlı mimarileri öne çıkarmışlardır [14].



Şekil 2.1. FPGA'lerin genel yapısı [15]

2.1.3. FPGA kullanımının avantajları

Gömülü sistemlerde, sistem mimarisi bakımından çeşitli alternatifler mevcuttur. Bunlar standart ASIC'ler, ASSP'ler, DSP'ler veya FPGA'ler gibi programlanabilir çözümlerdir. Yüksek performans, esnek programlanabilme, zahmetsiz güncelleme, düşük bakım maliyeti, güvenilirlik ve üretim adedi arttıkça düşük maliyete geçebilme gibi özellikler ideal mimaride olması beklenen özelliklerdir. Yüksek hız için gerekli olan yüksek veri boyutu, yüksek performans ve esnek programlanabilme özellikleri FPGA'lerde mevcuttur. Yüksek hızlı sinyal işleme uygulamaları genellikle sayısal sinyal işleyici (DSP) kullanılarak gerçeklenmektedir. Fakat FPGA devrelerinin aynı işlemi daha hızlı yapabilme yeteneği ve içine çoğullayıcı gömülebilme, aritmetik işlemleri başarıyla gerçekleştirebilme, çok sayıda çipi içinde barındırabilme gibi özellikleri nedeniyle, günümüzde sinyal işleme uygulamalarında FPGA kullanımı artmaya başlamıştır. FPGA'in sahip olduğu bu özelliklere ek olarak paralel işlem yapabilme yeteneği de vardır. Bu yetenek ve özellikleri sayesinde, FPGA kulanılarak gerçekleştirilen bir uygulama en hızlı DSP yongasından 500 kez daha hızlı olacak şekilde gerçekleştirilebilir [16].

Paralel işlem yapabilme yeteneği, aynı anda birden fazla işlemi yapabilmek demektir. Paralel işlemlere örnek olarak video sinyali işleme ve filtreleme uygulaması gösterilebilir. Bir video görüntüsü üzerinde filtreleme işleminin yapılabilmesi için; videonun bir resim karesinin giriş portlarından alınması, alınan resimin filtrelenmesi ve filtrelenen resim karesinin çıkış portlarından gönderilmesi aşamalarının gerçekleşmesi gerekir. Daha sonra ikinci resim karesi için de aynı işlemlerin gerçek zamanlı olarak tekrarlanması gerekmektedir. Mikroişlemciler gibi standart entegreler kullanıldığında, bu üç işlem (alma, filtreleme, gönderme) sırayla yapıldıktan sonra, gelen ikinci resmi alınmaya başlanır. Eğer bu işlemler yeterince hızlı yapılmazsa sıradaki resim kaçırılmış olur. FPGA'de ise bu işlemler paralel olarak devam eder. Yani ilk resim karesi alınıp filtreleme işlemi yapılırken ikinci resmin alımına başlanır. İlk resim gönderilirken ikinci resim filtrelenmeye ve üçüncü resim alınmaya başlanır. Bu yetenek, paralel işlem gerektiren uygulamalarda FPGA'leri avantajlı duruma getirmektedir. Paralel işlem yapabilme yeteneği FPGA'yi işlemciden ayıran ve birçok alanda üstün kılan en önemli özelliklerdendir. Yukarıda bahsedilen filtreleme örneğinde FPGA yerine mikroişlemci kullanılırsa, birinci alma-filtreleme-gönderme işlemi bittikten sonra ikinci resim alınabilir, bu da hız gerektiren işlemler için dezavantaj oluşturur.

İşlemcinin yapabileceği toplama, çarpma, I/O kontrolü gibi işlemler önceden tanımlıdır. Tasarımcı, kullanacağı yazılımlar ile bu işlemleri kendi amacına uygun olarak 'sıralı bir şekilde' yaptırabilir. FPGA'de ise donanım yapısı sabit değildir ve kullanıcı tarafından tanımlanır. FPGA'lerin gerçekleştireceği fonksiyonlar ve mantık hücreleri arasındaki bağlantılar kullanıcı tarafından belirlenir. Dolayısıyla FPGA'in yapabileceği işlemler önceden tanımlanmış değildir. Yazılan donanım tanımlama dili (HDL) koduna göre, istenilen işlemler 'paralel olarak' yani aynı anda yaptırılabilir [17].

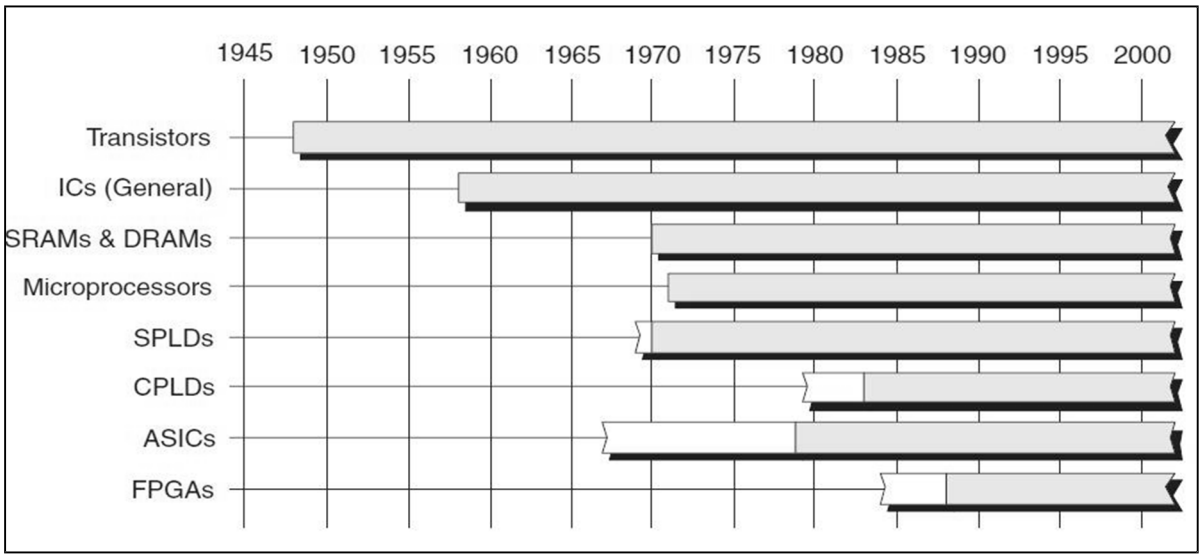
Teknolojinin hızlı ilerleyişine paralel olarak, donanımların mimarileri de esnek ve kolay güncellenebilir olmalıdır. Bu durum FPGA donanımlarını ASIC'lere göre de avantajlı hale getirmektedir. ASIC devrelerinin oldukça pahalı olması, üretimleri için uzun bir sürecin gerekiyor olması ve tasarımlarının geri dönüşü olmadan yalnızca bir uygulamaya özgü olarak oluşturulması FPGA'lere göre ASIC'lerin dezavantajlarıdır .

Güvenilirlik yönünden ele alınca da FPGA kullanımı diğer sistemlere göre daha avantajlı olarak ön plana çıkmaktadır. DSP uygulamalarında yazılım geliştirme araçları programlama için gerekli ortamı sağlarlarken, FPGA yongası ise donanımsal olarak programlanırlar. İşlemci temelli mimariler, görevleri yönetmek ve birden çok işlemci varsa bunlar için kaynakları yönetmek gibi uygulamaları içerirler. Bu mimarilerde sürücü katmanı donanım kaynaklarını yönetirken, işletim sistemi hafıza ve işlemci operasyonlarından sorumludur. Bir işlemci bir birim zamanda ancak tek bir komut işleyebilir. Dolayısı ile bu mimarilerde kritik görevler için her zaman işlemci kuyruğunda bekleme riski bulunmaktadır. FPGA'lerin işletim sistemi kullanmıyor olmaları ve paralel komut işleyebilme yetenekleri sayesinde bu tür güvenlik riskleri en aza indirgenir.

Sonuç olarak; mikroişlemci kullanmanın yeterli olduğu işlemlerde işlemci kullanmak, daha karışık ve yoğun veri işleme gibi performans gerektiren işlemlerde ise FPGA kullanmak isabetli olacaktır.

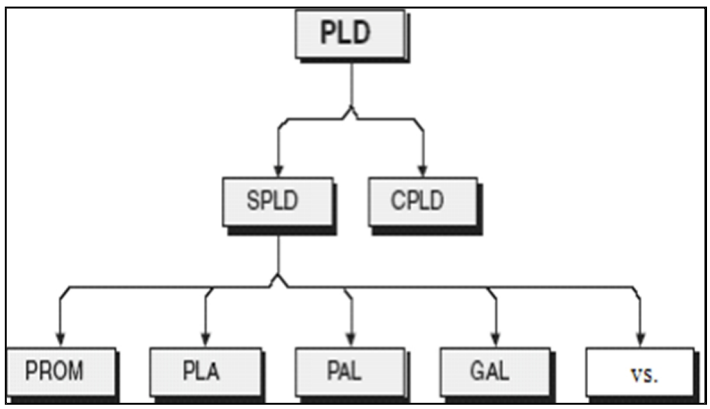
2.1.4. FPGA'in gelişim süreci

Şekil 2.2'de görülen zaman çizelgesindeki beyaz çubuklar o tekniğin vücut bulduğunu fakat bu zaman zarfında uygulanmasına geçilmediğini göstermektedir. Örneğin, Xilinx firması FPGA kartını 1984 yılında geliştirmesine rağmen bu alana ancak 90' lı yıllarda öncelik vermiştir.



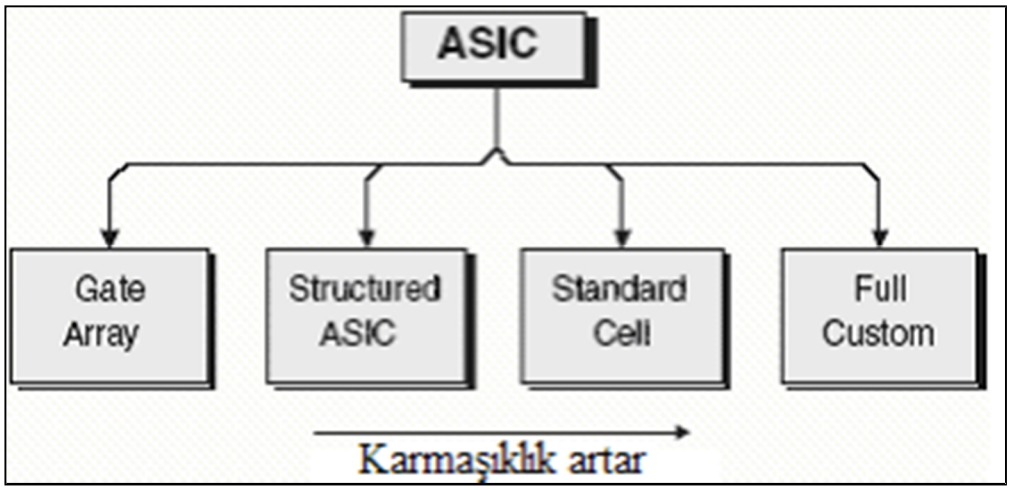
Şekil 2.2. Çeşitli devrelerin gelişimi ile ilgili zaman çizelgesi [18]

Zaman çizelgesinden görüldüğü üzere, ilk programlanabilen entegre devreler genel olarak PLD olarak takdim edilmiştir. PLD sınıflandırması şekil 2.3'de gösterilmiştir.



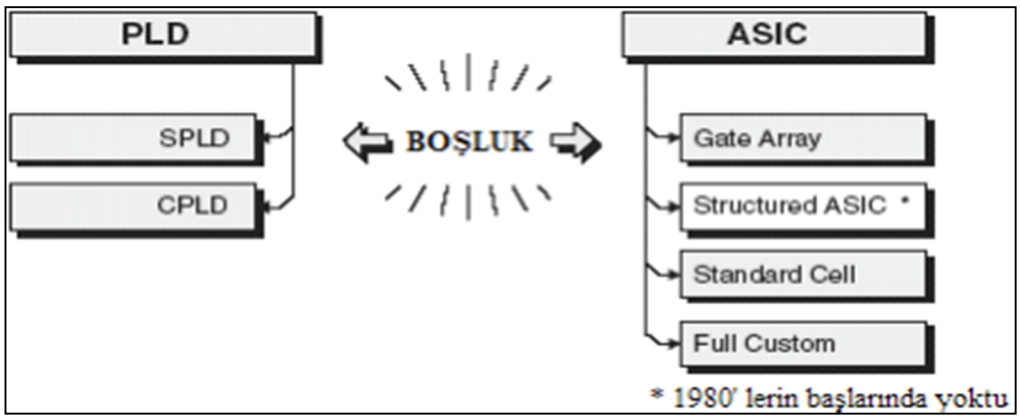
Şekil 2.3. Programlanabilen mantık aygıtlarının genel sınıflandırılması [19]

Zaman çizelgesinde görülen diğer bir entegre devre çeşidi ise ASIC devrelerdir. ASIC devreler de kendi içinde dört alt başlıkta incelenir. Şekil 2.4'de bu alt başlıklar gösterilmiştir.



Şekil 2.4. ASIC tasarım yöntemleri [20]

Şekil 2.5'te görüleceği üzere sayısal tümdevre sürecinde 80'li yıllarda belli boşluklar görülmeye başlanmıştır. Bir tarafta SPLD ve CPLD gibi yüksek yapılandırılma özelliğine ve hızlı tasarım sürelerine sahip programlanır yongalar vardı. Fakat PLD'ler geniş ve karmaşık tasarımları desteklemiyorlardı. Diğer tarafta ise kompleks sayısal sistemleri gerçekleştirmek için, üreticisi tarafından bir kereye mahsus olarak oluşturulan ASIC'ler vardı. ASIC devreleri oldukça hızlı çalışıyorlardı ama bu devrelerin yeniden programlanamayışı ve oluşturulma süreçlerinin oldukça uzun ve maliyetli olması gibi dezavantajları vardı.



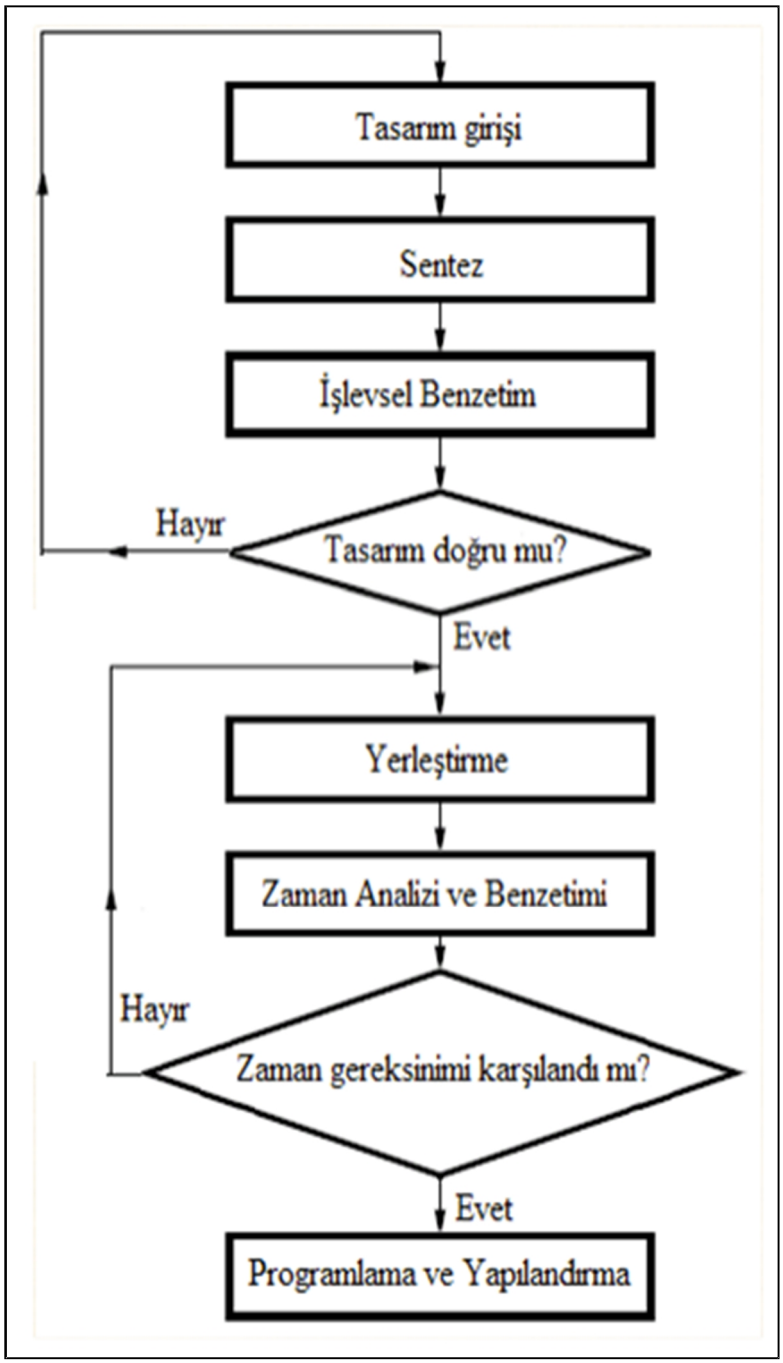
Şekil 2.5. FPGA üretimine duyulan ihtiyaç [21]

İşte bu boşluğun doldurulması amacıyla Xilinx firması FPGA adını verdiği yeni bir IC sınıfı geliştirdi ve 1984 yılında pazara sunulacak hale getirdi. İlk FPGA'ler CMOS tabanlı olarak çalışıyorlardı ve yapılandırılmaları için SRAM hücreleri kullanılıyordu. Temelde var olan mimariler, günümüzde birçok açıdan halen kullanılmaktadır.

2.1.5. FPGA ile tasarım aşamaları

FPGA'ler ile uygulama gerçekleştirme işlemi; FPGA'lerin içindeki bağlantıları ve lojik blokları çevresel aygıtlarla birleştirme ve istenilen amaca uygun olacak bir duruma getirme işlemidir. Tasarımın ilk adımı olarak, istenilen bir lojik devre fonksiyonu bilgisayar ortamında tasarlanır. Bu tasarım bir şematik çizim veya yüksek seviyeli bir donanım tasarlama dili şeklinde olabilir. FPGA ile uygulama gerçekleştirmek için gerekli olan tasarımın akış şeması Şekil 2.6'da görülmektedir. Bu tasarım süreci şu aşamalardan oluşur:

1. Tasarım girişi: Devrenin oluşturulması amacıyla VHDL veya Verilog HDL donanım tanımlama dillerinden herhangi biri kullanılarak tasarım süreci başlatılır. Bu diller mantıksal gerçeklemeler için özelleşmişlerdir. Böylece elektronik bileşenlerin davranışsal ve yapısal tanımlamaları kolayca yapılarak, esnek ve hızlı tasarımlar oluşturulabilir.
2. Sentez: Sentez araçları ile devre için gerekli olan mantıksal öğeler ve bu öğeler arasındaki bağlantılar oluşturulur.
3. İşlevsel benzetim: Sentezlenen devre bu aşamada işlevsel doğruluğu açısından incelenir. Hata durumunda tasarımda değişiklik yapılarak sentez ve benzetim basamaklarının tekrarlanması gerekir.
4. Yerleştirme: Yerleştirme araçları aracılığıyla, bağlantı listesinde tanımlanmış mantıksal öğelerin FPGA yongası içerisindeki gerçek mantıksal öğelere yerleşimi yapılır. Mantıksal öğeler arasındaki gerekli bağlantılar için yönlendirme hatları belirlenir.
5. Zaman analizi ve benzetimi: Yerleştirilmiş devrenin çeşitli yolları arasındaki yayılım gecikmeleri, zaman analizi birimi ile analiz edilir. Zaman benzetiminde ise; yerleştirilmiş devre, zamanlama ve doğruluk açılarından sınanır. İstenilen zaman gereksinimleri sağlanamaz ise tasarım sürecinde en başa dönülür ve tasarımda belirlenen hata ile ilgili gerekli geliştirme yapılır. İşlemler benzetim sonuçları yeterli oluncaya kadar tekrarlanır.
6. Programlandırma ve yapılandırma: Son aşamada FPGA yongası arzu edilen devreyi gerçeklemek için belirli donanımlar aracılığı ile programlanır. Yapılandırma ile mantıksal öğeler istenen şekilde yapılandırılır ve içerikleri belirlenir.



Şekil 2.6. FPGA tasarımı akış şeması [22]

2.1.6. FPGA'in kullanım alanları

Son yıllarda kullanımı yaygınlaşan FPGA donanımı, günümüzde iletişim araçlarında, radar sistemlerinde, sayısal sinyal işleme ve görüntü işleme uygulamaları gibi birçok alanda kullanılmaktadır. Ayrıca tıbbi görüntüleme sistemlerinde, kriptoloji uygulamalarında ve endüstriyel sektörde de popülaritesi her geçen gün artmaktadır.

2.2. Sayısal Görüntü İşleme

2.2.1. Görüntü işleme

Görüntü, gerçek yaşamdaki üç boyutlu nesnelerden oluşan bir sahnenin, basit iki değişkenli bir fonksiyon olarak tanımlanmasıdır. Görüntü işleme; ölçülmüş veya kaydedilmiş olan dijital görüntü verilerini, bilgisayar ve yazılımlar yardımı ile amaca uygun şekilde değiştirmeye yönelik yapılan çalışmanın adıdır. Diğer bir ifadeyle görüntü işleme; gerçek yaşamdaki görüntüleri sayısal bir resim haline getirme ve resmin özelliklerinin değiştirilmesi sonucunda yeni bir resim oluşturma işlemidir. Sayısal resim ifadesi, analog bir sinyalin sayısal bir sinyale dönüştürülmüş olduğunu ifade eder. Sayısal görüntü işlemenin mantığının anlaşılması için görme sistemimizin altında yatan temel mekanizmaların bilinmesi oldukça önemlidir. Göz bir fotoğraf makinesi gibi, beynin görme bölümleri de karmaşık bir sayısal görüntü işleme sistemi olarak düşünülebilir [23].

Dijital görüntü sayısal değerlerden oluşur. Lojik olarak 1 ve 0'lardan oluşan sayısal görüntü yapısını temsil eden a[M,N] ifadesi, 2 boyutlu dünyadan elde edilen a(x,y) fonksiyonundan örnekleme tekniği kullanılarak oluşturulur. M ve N değerleri görüntü matrisinde bulunan satır ve sütunları ifade eder. Satır ve sütunların kesiştiği bölgelere piksel denir. Pikseldeki değer; derinlik (z), renk (λ) ve zamanın (t) bir fonksiyonudur. Görüntü işleme, kaydedilmiş veya oluşturulmuş olan mevcut görüntüleri işleyerek resimleri ve grafikleri değiştirmek, yabancılaştırmak ya da iyileştirmek için kullanılır. Ölçeklendirme, döndürme, yansıtma, renk düzeltmesi, devirme, işaretleme, dönüştürme, sihirli değnek, katman örtüleme ve resim filtresi tanımları, görüntü işleme ile ilgili tanımlardan bazılarıdır [24]. Yaygın olarak kullanılan görüntü işleme teknikleri şunlardır:

- Kenar bulma
- Filtreleme
- Yapısal düzenleme
- Bölümlendirme
- Eşikleme
- Görüntü iyileştirme

2.2.2. Görüntü işlemenin kullanım alanları

Günümüzde dijital teknolojilerin hızlı gelişimi neticesinde, görüntü işleme uygulamaları hayatın birçok alanında yaygınlaşmaya başlamıştır. Görüntülerin yakalanması, yakalanan görüntülerin işlenmesi ve daha sonra bu görüntülerin saklanması uzun zamandır üzerinde çalışılmakta olan bir alandır. Görüntülerin işlenmesinin ve kaydedilmesinin kişisel amaçlı olarak yapılmasının yanısıra; güvenlik uygulamalarında, savunma sistemlerinde, simülasyon programlarında ve sağlık sektörü gibi birçok alanda da kullanılıyor oluşu, bu çalışmaların önemini daha da artırmıştır [25].

Görüntü işlemenin kullanım alanlarından bazıları şunlardır;

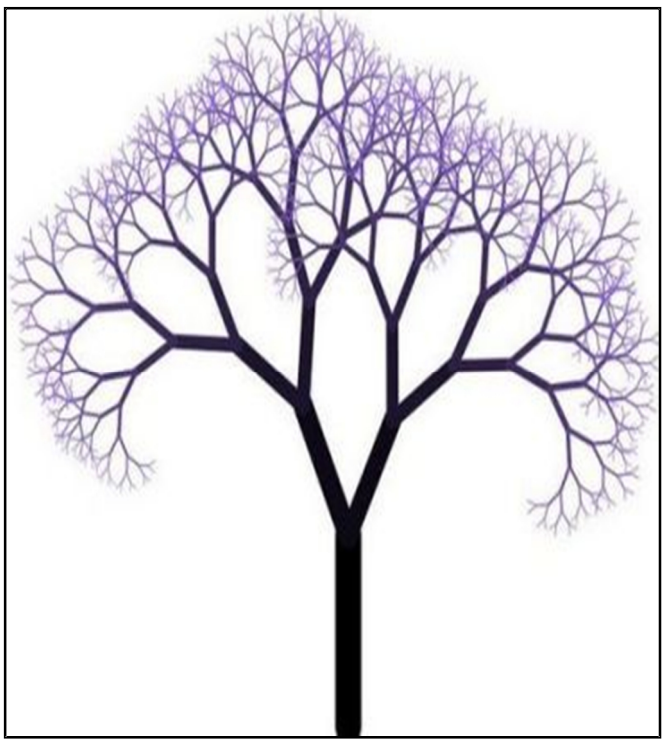
- Askeri endüstri
- Güvenlik, kriminal laboratuarlar
- Tıp
- Robotik uygulamalar
- Uzay bilimleri
- Radar ve uydu sistemleri
- Coğrafi haritaların çıkarılması
- Uçak simülasyonu ve bilgisayar oyunları için fraktal resim oluşturulması

2.3. Fraktal Geometri

Matematiğin önemli bir kolu olan geometri, insanoğlunun doğayı nasıl algıladığı ile yakından ilişkilidir. Algılama biçimleri geliştikçe, daha ileri geometrik yaklaşımlar ortaya çıkmıştır. Yunan matematikçi Euklid'in geometri anlayışına göre, doğada karşımıza çıkan şekiller; doğrular, düzlemler, daireler, küreler, üçgenler ve koniklerdenden ibarettir. Fakat yakından incelendiğinde, doğadaki nesnelerin Euklid geometrisindeki şekillere tam olarak benzemediği görülecektir. Yeryüzünde tam küre şeklinde olan bir elma veya tam koni şeklinde olan bir dağ bulmak oldukça zordur. Bu duruma dikkat çeken Benoit Mandelbrot 'bulutlar küre değildir, dağlar koni değildir, kıyılar çember değildir, ışık da doğru boyunca hareket etmez' tespitini yapmış ve 'fraktal' kavramını ortaya atmıştır [26]. Fraktal kavramı, yalnızca matematik ve geometri alanında değil; fizik, kimya, fizyoloji ve akışkanlar mekaniği gibi değişik alanlar üzerinde de önemli etkiler yaratan yeni bir geometri sisteminin doğmasına yol açmıştır. Bir fraktal şekil kendi kendine benzer parçalardan oluşmuş şekildir.

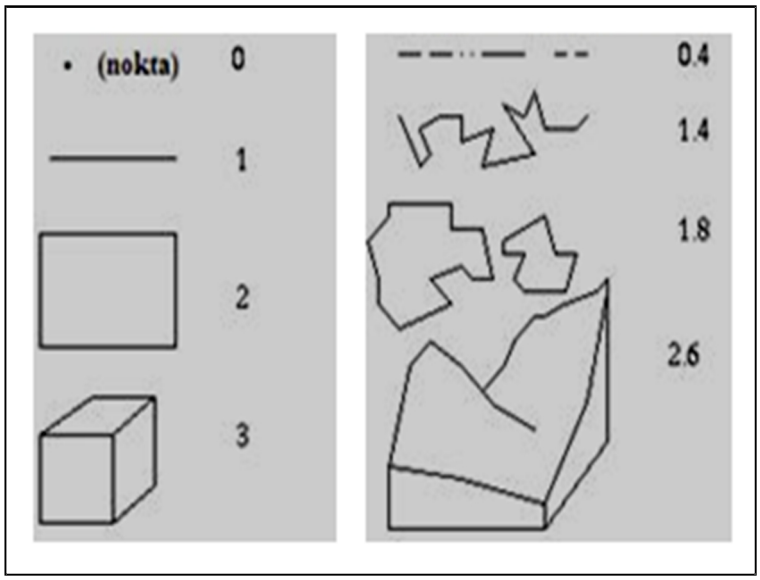
Kendine benzer bir cisimde, cismi oluşturan parçalar ya da bileşenler cismin bütününe benzer. Bu fraktal olgusu, Şekil 2.7'deki kar tanesi ve Şekil 2.8'deki ağaç dallarında kolayca görülebilir.

Şekil 2.7. Kar tanesindeki fraktal sistem [27]



Şekil 2.8. Ağaç dallarındaki fraktal sistem [28]

Fraktalların bir başka önemli özelliği de, fraktal boyut olarak adlandırılan matematiksel parametredir. Fraktalların bu özelliğine göre, cisim ne kadar büyütülürse büyütülsün ya da bakış açısı ne kadar değiştirilirse değiştirilsin fraktal cismin görünümü hep aynı kalır. Euklid geometrisinde ifade edilen boyutun tersine, fraktal boyut genellikle tam sayı olmayan bir sayıyla, yani bir kesir ile ifade edilir. Euklid geometrisindeki boyutlar ile fraktal geometrideki boyutların karşılaştırılması Şekil 2.9'da yapılmıştır.



Şekil 2.9. Euklid ve Fraktal boyutların karşılaştırılması [29]

Kendine benzerlik özelliği ve tamsayı olmayan boyut kavramlarıyla birlikte fraktal geometri, istatistiksel mekanikte, fiziksel sistemlerin incelenmesinde, sinyal sıkıştırma uygulamalarında, bilgisayar oyunlarının oluşturulmasında ve bilgisayarda grafik elde etme uygulamalarında giderek daha yaygın olarak kullanılmaya başlanmıştır. Örneğin, gökada kümelerinin evrendeki dağılımının saptanmasında ve akışkan burgaçlanmalarına ilişkin problemlerin çözülmesinde fraktal benzetimlerden yararlanılmaktadır. Fraktal geometri bilgisayar oyunlarında ve animasyon filmlerinde de sıklıkla kullanılmaktadır. Fraktal algoritma, engebeli dağlık araziler ya da ağaçların karışık dal sistemleri gibi karmaşık ve düzensiz doğal cisimlerin, gerçektekine benzer görüntülerinin oluşturulabilmesini olanaklı kılmıştır [30].

Doğayı daha iyi anlayabilmek ve modelleyebilmek için fraktal geometriye gereksinim vardır. Doğanın her hangi bir parçası, adaların dağılımı, dağlar, bir havzadaki ana akarsu ve kollarının oluşturduğu şekil, buzullar ve bitkiler gibi doğada var olan coğrafik şekiller hep fraktal özelliktedirler.

Şekil 2.10 ve Şekil 2.11 yakından incelenirse fraktal yaklaşımın dağlardaki ve ağaçlardaki varlığı açık bir şekilde görülecektir.

Şekil 2.10. Gerçek bir dağ resminde görülen fraktal düzen [31]

Şekil 2.11. Gerçek bir ağaç resminde görülen fraktal düzen [32]

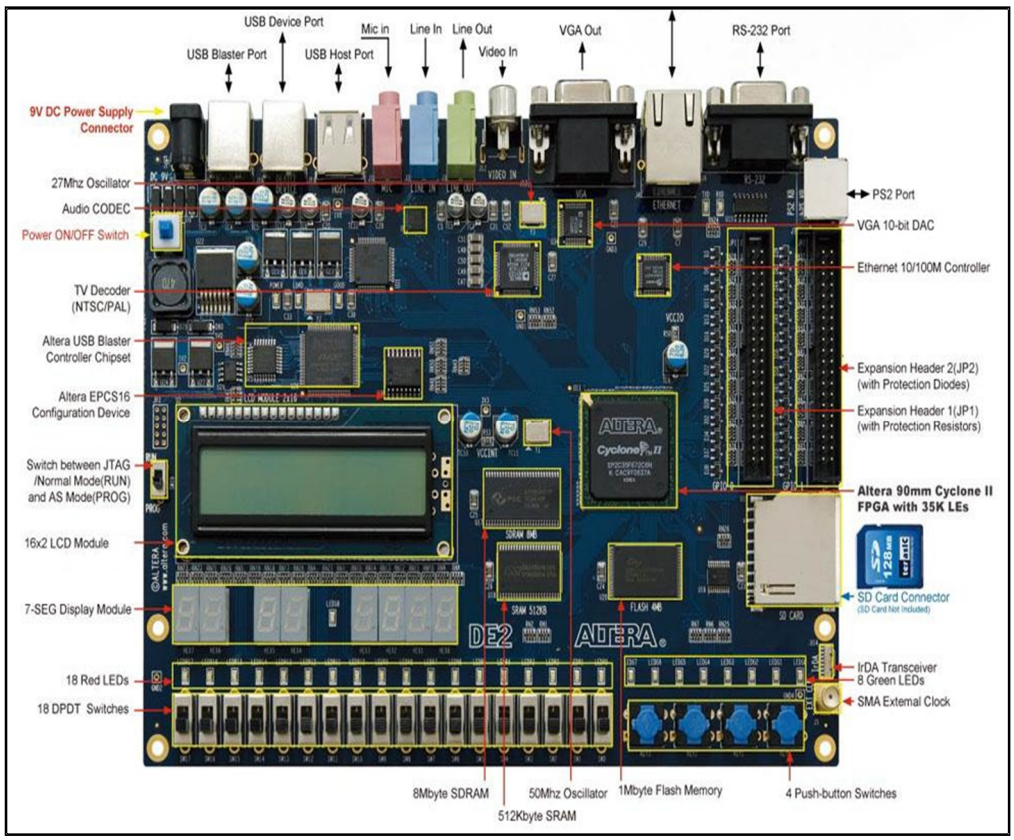
3. SİSTEM DONANIMI VE YAZILIMI

3.1. Kullanılan Donanımlar

Bu tezde gerçekleştirilen uygulamalarda Altera firmasının üretmiş olduğu DE2 serisi FPGA donanımı, uygulama sonuçlarının görüldüğü bir ekran ve ekran girişi ile FPGA'in VGA çıkışını birbirine bağlayan VGA ara kablosu kullanılmıştır.

3.1.1. Altera DE2 FPGA

FPGA üretiminde söz sahibi olan firmalar Xilinx ve Altera firmalarıdır. Bu iki firmanın ürettiği çok çeşitli FPGA platformları vardır. Platformların farklılığını belirleyen temel etmenler; hız, mantıksal blok sayısı, giriş-çıkış pinlerinin sayısı gibi donanımın sahip olduğu özelliklerdir. Altera firmasının ürettiği FPGA serileri Starix, Aria ve Cyclone olarak adlandırılırken, Xilinx firmasının ürettiği FPGA serileri Artix, Kintex, Virtex ve Spartan olarak adlandırılır. Bu tez çalışmasında, Altera Cyclone II 2C70 serisi bir FPGA geliştirme kartına sahip olan DE2 model FPGA donanımı kulanılmıştır. Donanımın görünümü ve giriş- çıkış portları ile ilgili bilgiler Şekil 3.1'de görülmektedir.



Şekil 3.1. Altera DE2 FPGA donanımı [33]

Altera DE2 FPGA donanımında bulunan giriş ve çıkış birimleri şunlardır:

- Dahili FPGA için USB Blaster yapılandırması
- Mikrofon girişi
- Video çıkışı (VGA 10-bit DAC)
- Video girişi (NTSC / PAL / Multi-format)
- RS-232
- Kızılötesi portu
- Ethernet bağlantısı
- USB 2.0 (A tipi ve B tipi)
- PS/2 mouse veya klavye bağlantı noktası
- Genişletme başlıkları (76 sinyal pin)

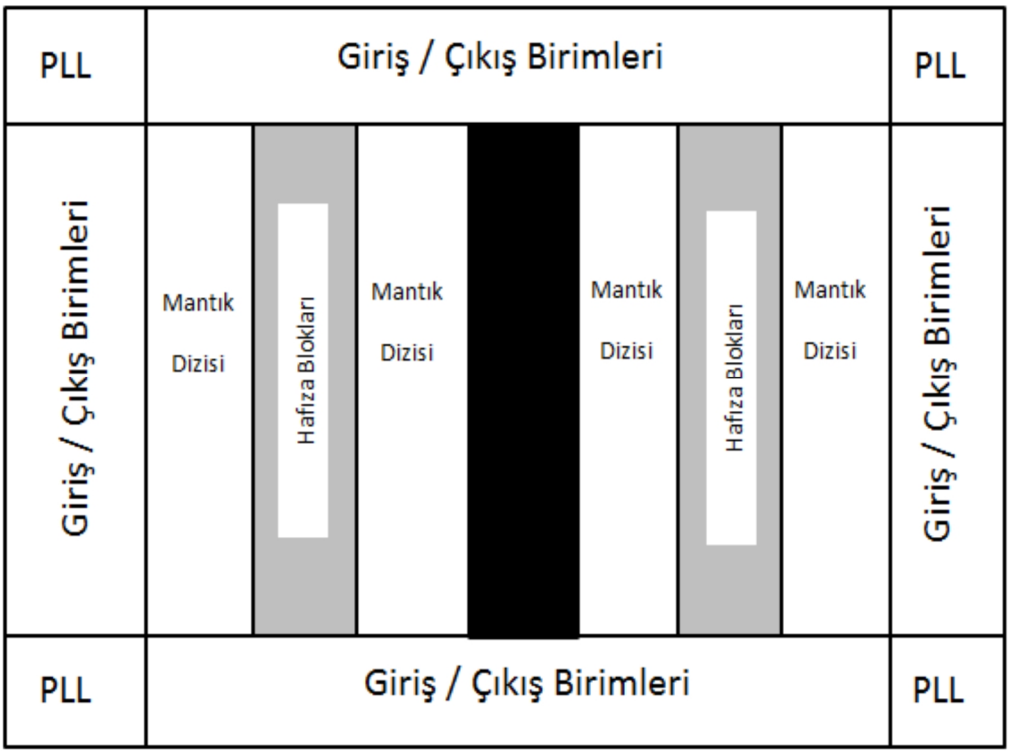
DE2 donanımında hafıza birimleri olarak aşağıdaki hafıza kartları bulunur:

- 8 MB SDRAM
- 512 KB SRAM
- 4 MB Flash

DE2 donanımda bulunan diğer bağlantılar ise şu şekildedir:

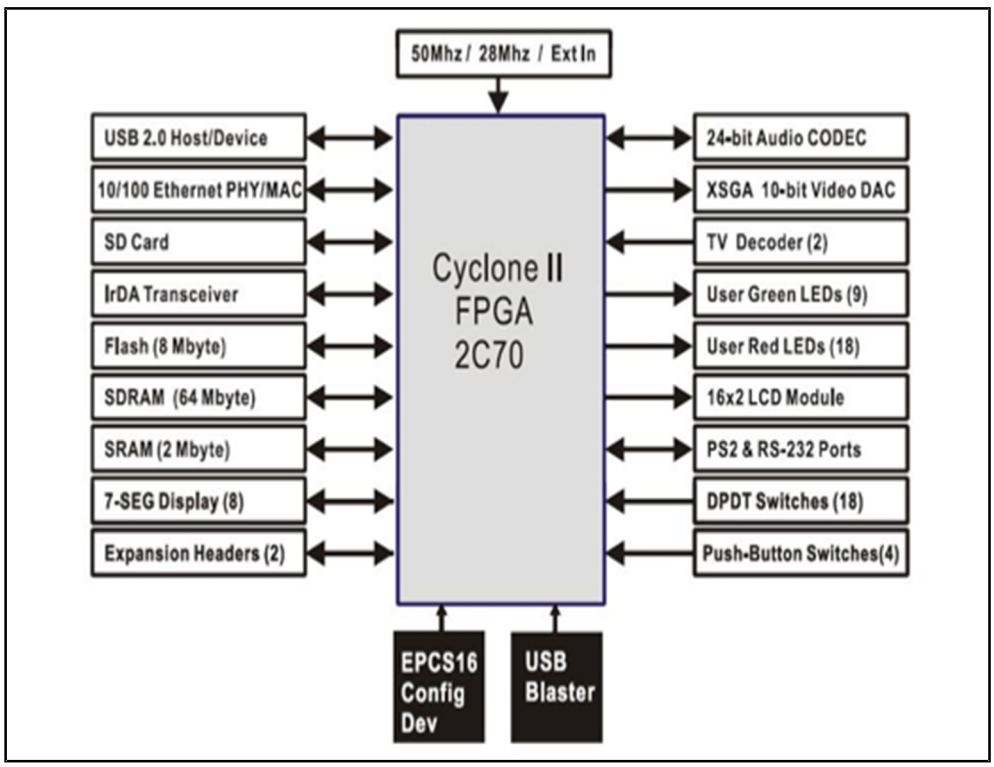
- Anahtarlar
- Saatler
- 18 adet geçiş anahtarı
- 4 adet buton anahtarı
- 18 adet kırmızı, 9 adet yeşil LED
- Sekiz adet 7-segment gösterge
- 16 x 2 LCD ekran
- 27 MHz ve 50 MHz osilatör
- Dış saat girişi

DE2 donanımı üzerinde Cyclone II 2C70 serisi bir FPGA geliştirme kartı vardır. Kart üzerinde 20000 lojik işlem birimi (LES) bulunmaktadır. Bu lojik birimler dışında 238 Kbit blok ram, 26 tane 18 x 18 çarpıcı ve 4 tane PLL bulunmaktadır. Altera Cyclone II'nin içerisinde barındırdığı ana birimler Şekil 3.2'de görüldüğü üzere; giriş-çıkış birimleri, lojik işlem birimleri, sistem saat yöneticisi ve hafıza bloklarıdır.



Şekil 3.2. Cyclone II kartı ana birimleri [34]

Tümdevre üzerinde kullanıcıya ait toplam 315 giriş ve çıkış pini bulunmaktadır. Kullanılan FPGA kartının giriş-çıkış birimleri Şekil 3.3'de verilmiştir.



Şekil 3.3. Cyclone II kartı giriş-çıkış birimleri [35]

3.1.2. VGA çıkış birimi ve PLL modülü

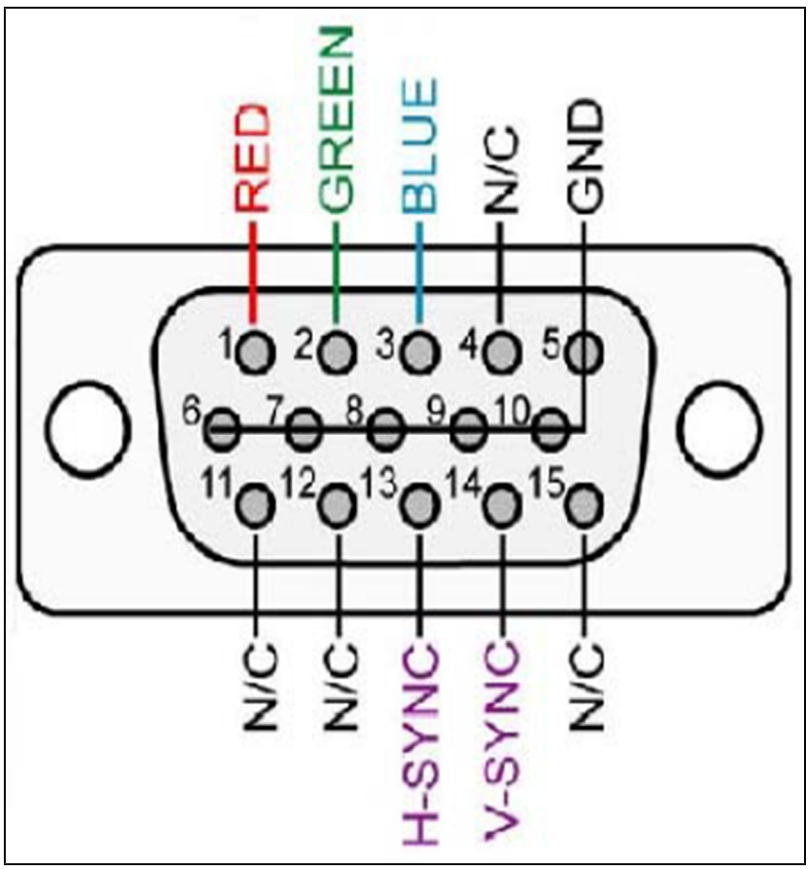
Video grafik dizisi, bilgisayarlardaki analog görüntü standartı ile 15 pin D-sub konektörü veya 640x480 çözünürlüğü ifade eder. 1988 yılında IBM tarafından piyasaya sürülmüştür [36]. VGA birimi; piksel haritasından, renk planından ve yatay-düşey senkronizasyondan oluşur. Ekranda görüntü gösterimlerinde yaygın olarak kullanılır. VGA biriminin daha iyi anlaşılabilmesi için öncelikle 'katot ışın tüpü (CRT)' monitörünün çalışma prensibi incelenmelidir. Bir monitörün en önemli parçası elektronik devreler ile birlikte CRT adı verilen havası boşaltılmış ve ön yüzeyi binlerce fosfor noktasından oluşan bir tüptür. Tüpün dar arka kısmında elektron tabancası bulunur. Tabanca içinde bulunan katot levhaları tel flaman ile ısıtılır ve tüp içerisinde serbestçe dolaşan elektron bulutunu oluşturur. Negatif kutuplandırılan katotlar ile pozitif kutuplandırılan ekranın iç yüzeyi arasına büyük bir gerilim farkı uygulandığında katotlarda oluşan elektronlar dış yüzeye doğru fırlar. Sabit olarak yerleştirilen odaklama elemanları, bu elektronları bir araya getirerek bir ışın halinde ekran orta yüzeyine odaklar. Bu ışını ekranın istenilen taraflarına yönlendirmek için elektron tabancasının etrafında yatay ve dikey saptırma bobinleri bulunur. Bu ısının ön yüzeyde gezdirilmesi sonucunda görüntüler ortaya çıkar [37]. Ekran kartından sinyal geldiği sürece ışın monitörün sol üst köşesinden başlayarak fosfor ile kaplı yüzeyi satır ve sütun halinde tarar.

Bu tez çalışmasında ekran kartı yerine sinyaller FPGA kartından monitöre gönderilmektedir. Bir satırın taranması tamamlandıktan sonra ışının bir sonraki satırın en sol koşesine gitmesi için yatay saptırma bobininin uygun şekilde kontrol edilmesi gerekir. Aynı şekilde tüm ekranın taranma işlemi tamamlandıktan sonra ışının tekrar (0,0) başlangıç noktasına gitmesi için de düşey saptırma bobininin uygun şekilde kontrol edilmesi gerekir. Bu geçiş işlemine eşleme adı verilmektedir. Monitörün içerisinde, yatay ve dikey saptırma bobinlerinin kontrolünün sağlanması amacıyla, osilatorler ve kuvvetlendiriciler tarafından oluşturuluran testere dişi dalgalar vardır. Bu dalgalar taramanın tüm ekran boyunca gerçekleşmesini sağlar [38]. VGA portu aracılığıyla kırmızı, yeşil ve mavi renklerin oluşturduğu kombinasyonların ekran piksellerine gönderilmesiyle görüntü oluşur. Bu çalışmada her bir renk değeri 4 bit ile ifade edilmiştir. Buna göre piksele atanan renk değerleri 12 bit ile ifade edilir. 12 bit ile belirtilen renk değerleri kullanılarak, 4096 farklı renk tonu ekran üzerinde elde edilebilir.

VGA iletiminde 5 farklı sinyal kullanılır. Bunlar;

- horizontal sync: Satır eşlemesini sağlayan dijital sinyal
- vertical sync: Kare eşlemesini sağlayan dijital sinyal
- red(R): Kırmızı renk bilgisini taşıyan analog sinyal
- green(G): Yeşil renk bilgisini taşıyan analog sinyal
- blue(B): Mavi renk bilgisini taşıyan analog sinyaldir.

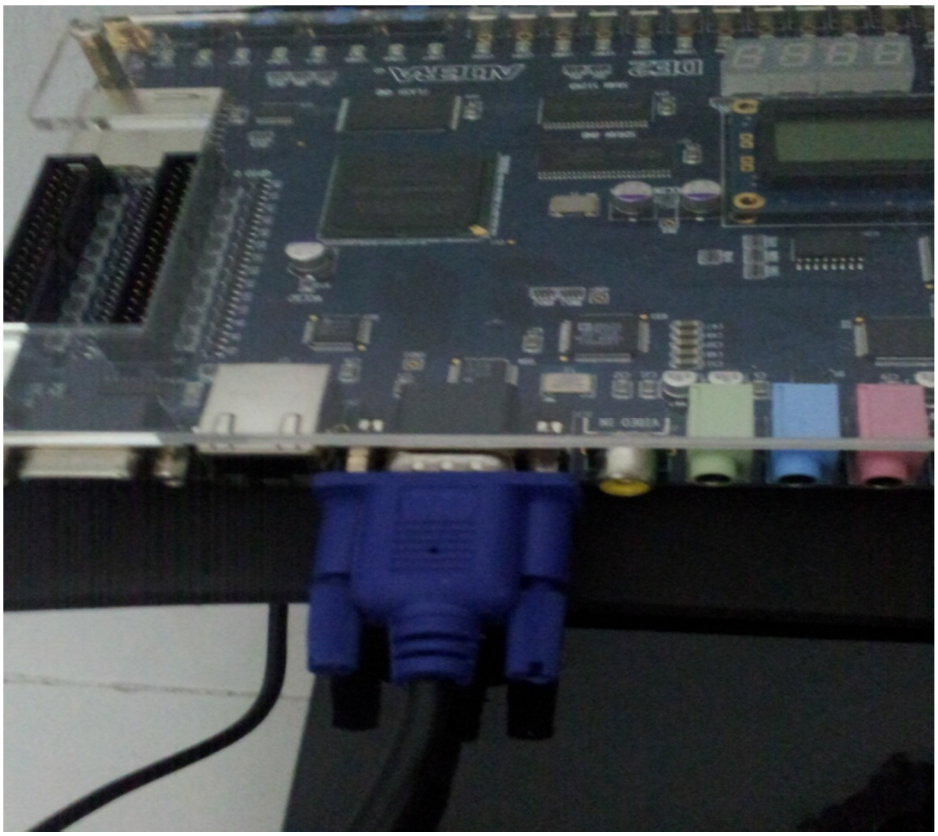
Şekil 3.4'de FPGA donanımında bulunan VGA soketinin veri yolları görülmektedir.



Şekil 3.4. VGA çıkış biriminde veri yolları

VGA çıkışını kullanabilmek için piksel zaman hesaplamalarının yapılması, tetikleyicinin düzenlenmesi ve renk tonlama ayarlamalarının yapılması önemlidir. Ekran üzerinde tarama işlemi yukarıdan aşağıya doğru satır satır yapılır. Bu satırlar ile, ekran soldan sağa doğru piksel piksel olacak şekilde taranmış olur. Görüntünün hareketli olarak algılanması için ekranın bir saniyede onlarca kez taranması gerekmektedir. Bir saniyede yapılan tarama sayısına tazeleme oranı denir ve bir bilgisayar ekranında bu oran 60 veya daha fazla olmaktadır.

Bu çalışmada kullanılan DE2 FPGA donanımı üzerinde VGA çıkış portu vardır. Bu çıkış portu Şekil 3.5'de görülmektedir. Şekil 3.6'da görülen VGA bağlantı kablosunun bir ucu FPGA'de bulunan çıkış portuna, diğer ucu ekranda bulunan giriş portuna bağlanarak, oluşturulan görüntülerin ekranda görüntülenmesi sağlanmıştır.



Şekil 3.5. FPGA donanımı VGA çıkış birimi

Şekil 3.6. VGA bağlantı kablosu

Verilog programlama dilinde, ekran kontrolü için VGA_Kontrolü isimli bir modül oluşturulur. Bu modülün içinde belirtilen değerler ile VGA senkronizasyonu sağlanır. Uygulamalarda kullanılan Verilog kodunun, bu senkronizasyonu sağlamak için gerekli olan bölümü aşağıdaki gibidir:

```
// Yatay Senkronizasyon Parametreleri
parameter H_SYNC_CYC   = 95;
parameter H_SYNC_BACK  = 65;
parameter H_SYNC_ACT   = 640; //  Yatay olarak 640 piksel
parameter H_SYNC_FRONT = 0;
parameter H_SYNC_MAX   = 799;
//  Düşey Senkronizasyon Parametreleri
parameter V_SYNC_CYC   = 2;
parameter V_SYNC_BACK  = 33;
parameter V_SYNC_ACT   = 480; //  Düşey olarak 480 piksel
parameter V_SYNC_FRONT = 10;
parameter V_SYNC_MAX   = 524; //
// Başlangıç ofseti
parameter X_START      = H_SYNC_CYC + H_SYNC_BACK;
parameter Y_START      = V_SYNC_CYC + V_SYNC_BACK;
```

Yatay ve düşey senkronizasyonda bulunan bazı terimlerin anlamları şu şekildedir:

Yatay geri kayma: Piksel işaretçisinin tekrar yatay olarak sol tarafa geldiği alan.

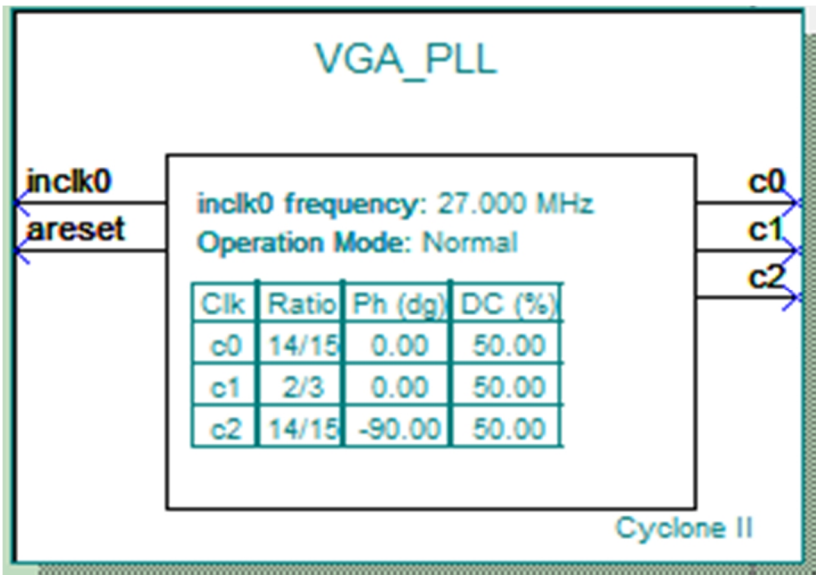
Düşey geri kayma: Piksel işaretçisinin tekrar düşey olarak ekranın üst kısmına geldiği alan.

Front porch: Ekranın sağ sınır alanı olarak tanımlanır. Video sinyalinin bu kısımda ekrana basılmaması gerekir.

Back porch: Ekranın sol sınır alanı olarak tanımlanır. Video sinyalinin bu kısımda ekrana basılmaması gerekir.

Ekran modülünün çalışması için saat sinyaline ihtiyaç duyulur. DE2 FPGA üzerinde 27 MHz ve 50 MHz frekansa sahip osilatör bulunur.

Kullanıcı bu iki frekans değeri dışında bir sinyal üretmek isterse, FPGA platformunda bulunan PLL bloklarından birini veya birkaçını kullanabilir. PLL bloklarını kullanmak için, Quartus II programında Tools sekmesi altında bulunan Mega Wizard Function aracılığı ile alt_pll modülü oluşturulur. Uygulamalarda kullanılan PLL bloğu Şekil 3.7'deki gibidir.



Şekil 3.7. VGA PLL modülü

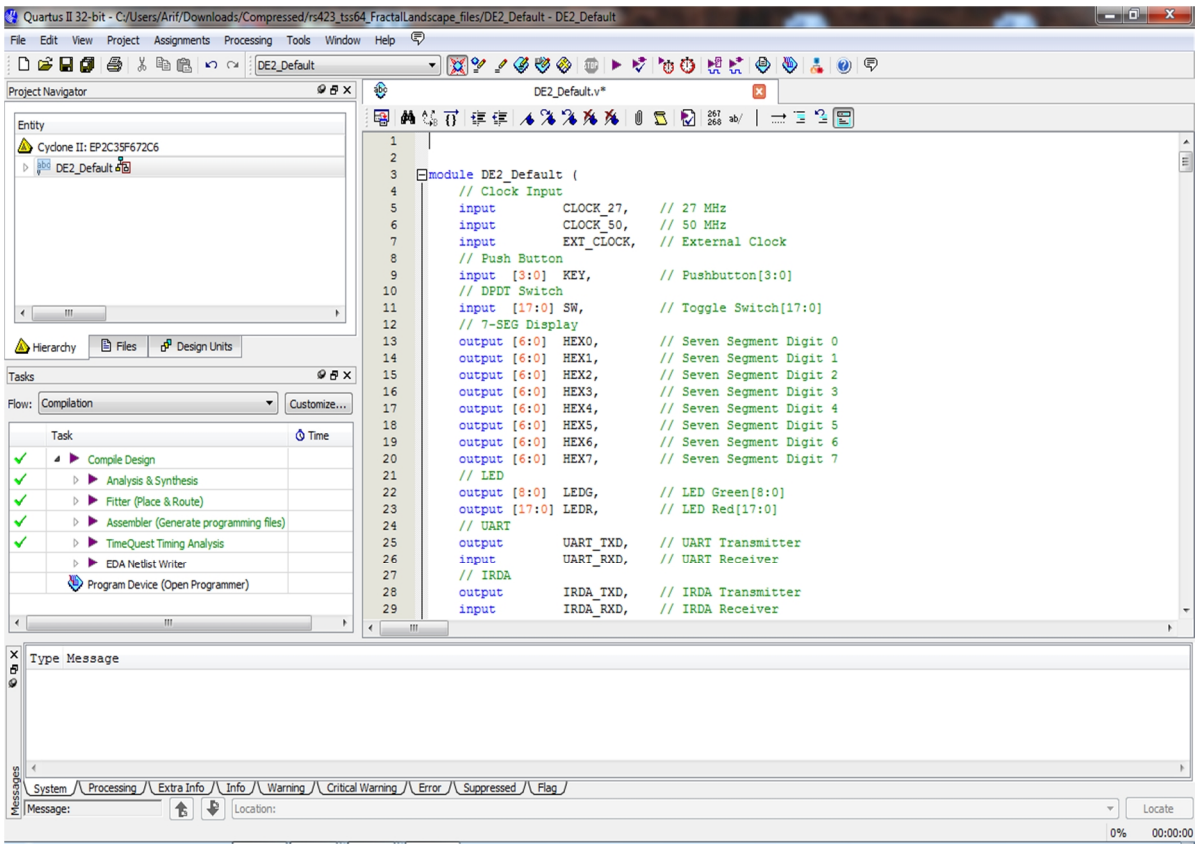
3.2. Kullanılan Yazılımlar

FPGA platformu ile uygulama gerçekleştirmek için, Altera firmasının hazırlamış olduğu Quartus II yazılımı kullanılır. Uygulanacak olan algoritmalar Quartus II arayüzü üzerinde Verilog veya VHDL programlama dillerinden biri kullanılarak gerçeklenir. Hazırlanan programın çalışma durumunun test edilmesi için Modelsim programı kullanılır.

3.2.1. Quartus II yazılımı

Quartus II yazılımı tamamen entegre bir yazılım geliştirme ortamıdır. Mantık kapıları tasarımı, devre dizaynı ve simulasyonu, HDL kodunun derlenmesi gibi işlemlerin gerçekleştirilmesini sağlar. FPGA programlama ile uğraşan kullanıcıların ihtiyaç duyabileceği tüm birimleri içinde barındırmaktadır. Quartus II kullanılarak şematik ve HDL dilleri ile tasarım oluşturulur, oluşturulan tasarım sentezlenir ve bu tasarımın FPGA'e yerleştirilmesi gerçekleştirilir. Ayrıca projenin doğruluğu Quartus yazılımı üzerinde simülasyon yapılarak kontrol edilebilir.

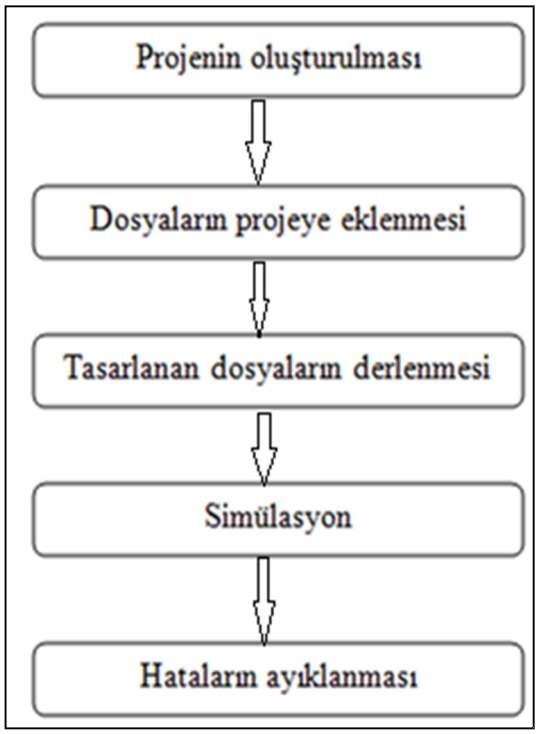
Şekil 3.8'te Quartus II programının ana ekranı görülmektedir. Ekranda görülen 4 farklı pencere vardır. Gerçekleştirilecek olan proje ile ilgili program kodu ana pencereye yazılır. 'Project navigator' isimli pencerede, yazılmış olan ana modül ve alt modüller görülebilir. 'Task' penceresinde; analiz, sentez, yerleştirme, zaman analizi ve benzetimi gibi tasarım için gerekli olan görevlerin başarılı bir şekilde gerçekleşip gerçekleşmediği incelenir. Yazılan programda hata olması durumunda, 'Type message' penceresinde hata uyarısı görülür.



Şekil 3.8. Quartus II yazılımı anasayfa görüntüsü

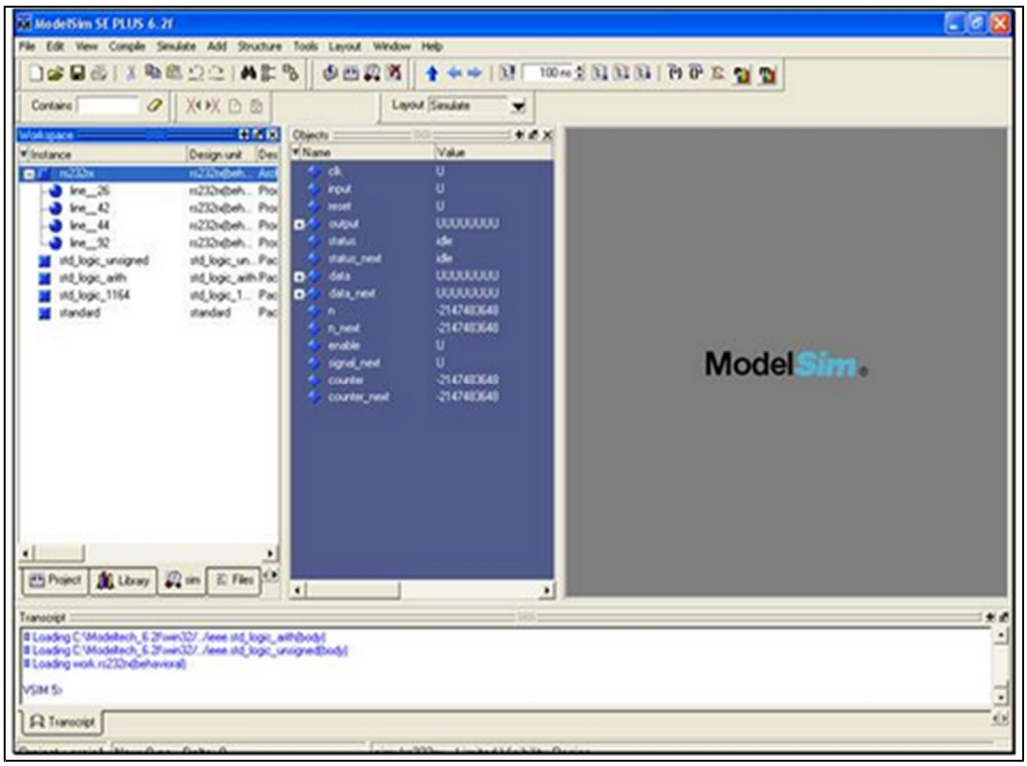
3.2.2. Modelsim Programı

Tasarımcı, hazırlamış olduğu programdan istediği sonuçları alıp alamayacağını anlayabilmek için test ve simülasyon programlarından yararlanır. Modelsim programı, VHDL ve Verilog HDL programlama dilleri ile oluşturulan sistemlerin simülasyonunda kullanılır. Tasarlanan sistemin ve sistemin alt modüllerinin gerçeklenmesi halinde donanımın çalışma durumunun nasıl olacağı hakkındaki bilgi, simülasyon neticesinde elde edilir. Quartus II ve Modelsim programlarının ortak kullanımı ile transfer seviyesinde (RTL) ve kapı seviyesinde (gate level) simülasyonlar yapılabilir. Modelsim'de proje oluşturmak için gerekli olan aşamalar Şekil 3.9'daki gibidir.



Şekil 3.9. Modelsim'de proje oluşturulması

Proje oluşturulduktan sonra simülasyonu yapılacak olan modüllerin ve dosyaların projeye eklenme işlemi gerçekleştirilir. Örnek bir Modelsim ana ekranı Şekil 3.10'da görülmektedir.



Şekil 3.10. Modelsim programında anasayfa görüntüsü

3.2.3. Verilog HDL donanım tanımlama dili

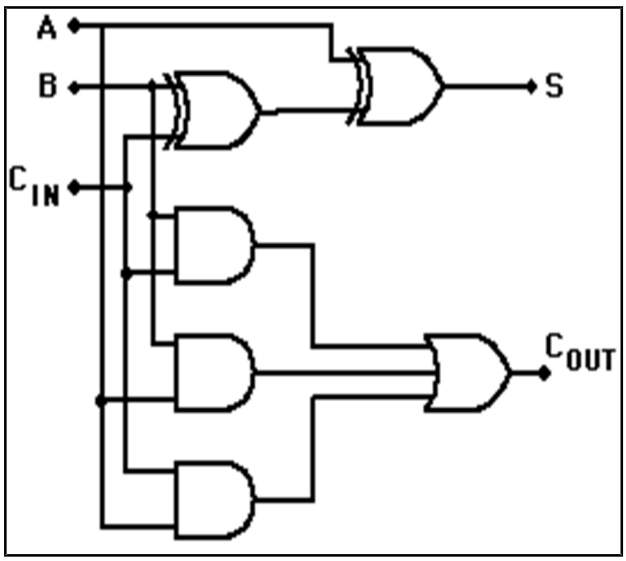
FPGA'lerde algoritmaları oluşturmak için kullanılan programlama dillerine genel olarak 'donanım tanımlama dili' denir. Donanım tanımlama dilleri, FPGA'de bulunan mantıksal blokların ve ara bağlantı anahtarlarının uygun şekilde programlanmasında kullanılır. En sık kullanılan diller VHDL ve Verilog programlama dilleridir. VHDL, 'Çok yüksek hızda donanım tanımlama dili' olarak tanımlanır ve donanımlarda kullanılan mantık elemanlarının davranışlarını tanımlar. Verilog dili de elektronik sistemleri modellemek için kullanılan diğer bir donanım tanımlama dilidir. Verilog; analog, sayısal ve karışık işaretli devrelerin tasarımını, doğrulanmasını ve yürütülmesini değişik düzeylerde desteklemektedir. Verilog donanım tanımlama dili, C programlama diline yakın bir söz dizimine sahiptir. Verilog tasarımında modüller vardır ve bu modüller arasında bir hiyerarşi bulunur. Modüller bir takım giriş, çıkış ve çift yönlü portlar şeklinde tanımlanır. Bir modül içinde yazmaç ve kablo listesi bulunur. Eş zamanlı ve ardışık ifadeler modülün davranışını; portların, kabloların ve yazmaçların arasındaki ilişki ile tanımlar. Ardışık ifadeler bir begin/end bloğuna konur ve blokla beraber ardışık olarak yürütülür. Tüm eş zamanlı ifadeler ve begin/end blokları koşut olarak yürütülür. Tasarımdaki modüller sadece sentezlenebilir ifadeler içeriyorsa, tasarımın temel bileşenlerini ve bağlantılarını içeren bağlantı listesi, yazılım ile sentezlenir. Elde edilen bu bağlantı listesi bir FPGA'i tanımlamak amacıyla kullanılır [39].

Verilog dili, kullanıcıların davranışsal seviyede, yazmaç transfer seviyesinde ve kapı seviyesinde sayısal tasarım yapabilmelerini sağlar. Davranışsal seviye, bir sistemi koşut zamanlı yani aynı zamanlı algoritmalar ile tanımlar. 'Fonksiyonlar' (functions) 'görevler' (tasks) ve 'her zaman' (always) blokları bu seviyenin temel elemanlarıdır. Yazmaç transfer seviyesi kullanılarak yapılan tasarımda, bir devrenin işlemleri ve yazmaçları arasındaki verilerin transfer karakteristiği belirtilir. Bu sistemde harici bir saat kullanılır. Kapı seviyesi kullanılan tasarımlarda ise lojik seviyesi içerisinde bir sistemin karakteristiği mantıksal bağlantılarla ve onların zamanlama özellikleriyle tanımlanmıştır. Bu tasarımlardaki tüm sinyaller ayrık sinyallerdir. Sadece belirli mantıksal değerlere sahip olabilirler. Kullanılabilir işlemler mantık kapılarıyla (and,or,not) önceden tanımlanmıştır [40]. Bu tezde gerçekleştirilen uygulamalarda, C programlama diline benzerlik göstermesi ve Altera firmasının bilgilendirme ve tanıtım dökümanlarında kullanılan dil olması nedeniyle Verilog HDL dili tercih edilmiştir. Verilog dilinde 'modül' kavramı çok önemlidir.

Kullanıcı, uygulamada gerçekleştirmek istediği tüm algoritmaları ayrı ayrı alt modüller şeklinde oluşturur. Algoritmada kullanılacak olan giriş-çıkış sinyallerinin yönleri ve boyutları modüllerin içinde belirtilir. Oluşturulan alt modüllerin birleşmesiyle 'ana modül' meydana gelir. Modül oluşturulurken şu yapı kulanılır;

```
module module_name (port names);
  Port Declaration (giriş-çıkış bildirimi)
  Data Type Declaration (Data'nın wire veya register oluşu)
  Module Structure (algoritmanın bildirimi)
endmodule
```

Örneğin, bir 'tam toplayıcı (full-adder)' tasarımında giriş elemanları A, B ve Cin olsun. Çıkış elemanları ise S ve Cout olsun. Tasarımda, Cin elde girişini (carry in), Cout elde çıkışını (carry out), S ise toplamı (sum) ifade etmektedir. Bu elemanlardan oluşan tam toplayıcı devresi Şekil 3.11'de verilmiştir.



Şekil 3.11. Tam toplayıcı devresi

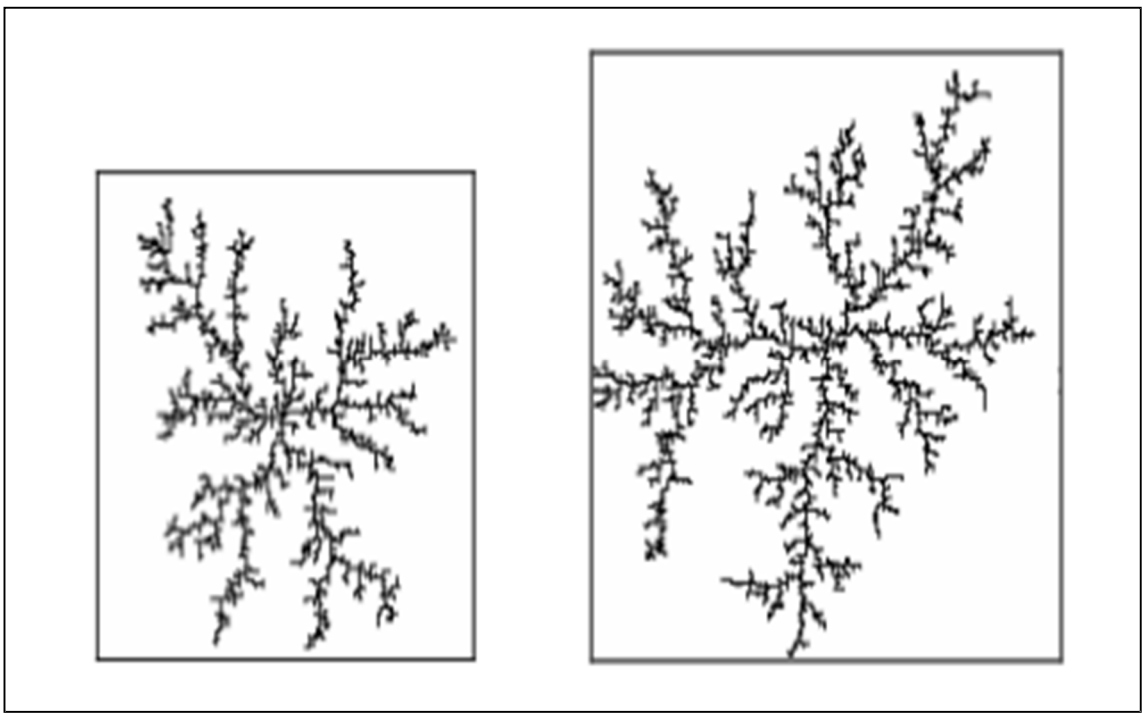
Şekil 3.11'de görülen devrenin, Verilog dili ile bir modül içinde oluşturulması şu şekilde olur:

```
module full_adder (A, B, Cin, S, Cout);
    input A, B, Cin ;
    output  S, Cout ;
    wire    S, Cout ;
    assign  S = A ^ B ^ Cin ;
    assign  Cout = (A & B) | (A & Cin) | (B & Cin);
endmodule
```

4. GERÇEKLEŞTİRİLEN GÖRÜNTÜ İŞLEME UYGULAMALARI

4.1. Fraktal Şekil Oluşturulması

Doğadaki yapılar farklı geometrik şekillere sahiptir. Her bir yapı, ortamdaki temel yapı taşlarına göre farklı bir tanecikler kümesidir. Heterojen çevre şartları ve düzensizlik olmasına rağmen kümelerin düzenli olabilmesi ve simetrik özellikler taşıması bilimsel araştırmalara konu teşkil etmiştir [41]. Simetrik özelliğe sahip ve kendi kendine benzer parçalardan oluşmuş şekillere fraktal şekil adı verilir. Bilgisayar ortamında fraktal şekil elde etme yöntemlerinden biri 'difüzyon ile sınırlı tanecik kümeleşme modeli (DLA)' kullanarak şekil oluşturmaktır. DLA modeli (T.Witten 1981) tarafından oluşturulan bir modeldir. Bu modele göre; öncelikle bir kapalı kare örgünün boyutları belirlenir. Örgünün merkezine yani orta noktasına bir çekirdek tanecik yerleştirilir. Bu tanecik rastgele olarak sağ, sol, yukarı veya aşağı yönlerde ilerlemeye başlar. Bu hareket örgü sınırlarına ulaşıncaya kadar devam eder. Difüzyon ile sınırlı kümeleşme modeli ile oluşturulan tanecik kümeleri, kapalı-kare örgü içerisinde merkezi taneciğe bağlı ana dallar ve bu dallara rastgele noktalardan eklenmiş alt dallardan meydana gelen seyrek yapılı kümelerdir [42]. 160x160 pikselden ve 200x200 pikselden oluşan örgüler için bilgisayar kullanılarak elde edilen küme simülasyonları Şekil 4.1'de gösterilmiştir.



Şekil 4.1. DLA modeli ile oluşturulan fraktal şekil simülasyonları [43]

Sinyal işleme, bilgisayar grafikleri, ordu kamuflajı, teknik analiz gibi alanlar genel olarak fraktal şekillerin uygulama alanlarıdır. Sinyal işlemeye örnek olarak fraktal görüntü sıkıştırma ve filtreleme uygulamaları; teknik analize örnek olarak ise hava durumu tahmini ve depremlerin tahmin edilmesi gösterilebilir. DLA modeli kullanılarak oluşturulan fraktallar; bakteri kolonilerinin büyümesi, elektrolitten ayrışan maddenin bir elektrotta birikerek büyümesi, kristallerin büyümesi, yalıtkan ortamda elektrik boşalması, bitkilerin dallanması, akışkanların gözenekli ortamlarda yer değiştirmesi, canlılarda damarların ve sinirlerin dallanmasının gözlemlenmesi gibi farklı alanlarda kullanılmaktadır [44].

Örneğin (Thomas C. Halsey) fizik, kimya, biyoloji ve matematik bilim dallarında sık kullanılan 'pattern formation' sisteminde DLA modeli ile oluşturulan fraktalleri kullanmıştır [45]. (Bayırlı, M., 2004) 'mangan dentritleri ve difüzyonla sınırlı kümeleşme modeli' isimli çalışmasında; manyezit cevheri yüzeylerindeki mangan kümelerinin nasıl büyüyerek oluştuğunu, DLA modeline göre simülasyonla elde edilmesi mümkün olan kümelerle karşılaştırılarak incelemiştir. Bu amaçla manyezit cevheri nümunelerinden oluşmuş birbirinden farklı mangan kümelerinin resimlerini, bir görüntü tarayıcı elde ederek bilgisayar ortamına aktarmış; aktarılan bu görüntüleri kullanarak her bir kümenin yoğunluk korelasyon fonksiyon üsleri ve fraktal boyutlarını hesaplamıştır [46]. (Bülbül, M. 2010) ise yüksek lisans çalışmasında, iki boyutlu Ising modelinin simülasyonunu gerçekleştirmek için, DLA modeli ile elde edilen fraktal sistemden yararlanmıştır [47].

Örneklerden anlaşıldığı üzere, DLA modeli ile oluşturulan fraktal şekillerin birçok farklı amaçta kullanımı bulunmaktadır. Bu çalışmada FPGA ile gerçekleştirilen uygulamada, 320x240 pikselden oluşan bir örgü kullanılmış ve elde edilen fraktal şekil üzerinde renklendirme yapılmıştır. Bu uygulama için gerekli olan rastgele sayı üretme algoritmasının gerçeklenmesi ve rastgele oluşturulan fraktal görüntünün farklı renk tonlarıyla renklendirilmesi Verilog donanım tanımlama dili ile sağlanmıştır. Uygulamada rastgele hareketi sağlamak için bir rastgele sayı üreticiye ihtiyaç vardır. Bu uygulamada, rastgele sayı üretiminde sıkça tercih edilen 'doğrusal geri beslemeli kaydırma yazmacı' kullanılmıştır. Bu yöntem genellikle ikili tabandaki sayılar üzerinden çalışır ve sistem iki adımdan oluşur. İlk olarak, yazmaç keyfi olarak seçilen değerler ile yüklenerek başlangıç değeri elde edilir. İkinci aşamada ise, sayıların kaydırılması ve açılan boşluğa bir önceki adımda elde edilen fonksiyon sonucunun yerleştirilmesi yapılır. Bu sistemde bir veya daha fazla XOR kapısı kullanılır [48].

Doğrusal geri beslemeli kaydırma yazmacına örnek olarak 4 bitten oluşan bir sayı düşünülsün. Sayının değerleri aşağıdaki gibi olsun:

A	B	C	D

Kullanılacak olan fonksiyon F= C ⊕ D olarak tanımlı olsun. Bu durumda her adımda aşağıdaki şekilde bir kaydırma işlemi gerçekleşir:

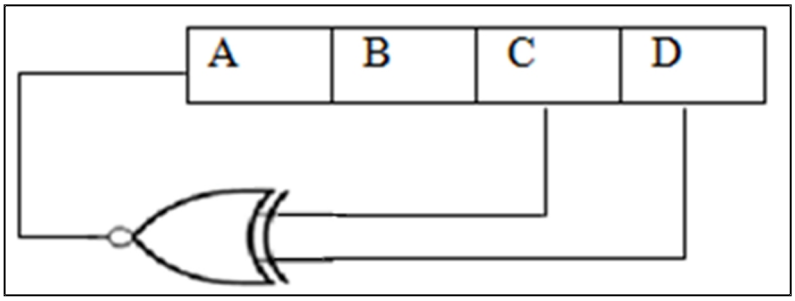
D = C

C = B

B = A

A = F

Yeni durumda, 4 bitlik rastgele bir sayı elde edilmiştir. Bu işlemi gerçekleştiren sistem Şekil 4.2'deki gibidir.



Şekil 4.2. XOR kapısı ile oluşturulan kaydırma yazmacı

Bu örnekte A, B, C ve D sayıları başlangıç değeri olarak sırasıyla 1,1,1,1 olursa, sistemin çalışması şu şekilde olacaktır;

1111 (son iki bit 1⊕1 = 0)

0111 (başa 0 eklendikten sonra son iki bit 1⊕1 = 0)

0011 (başa 0 eklendikten sonra son iki bit 1⊕1 = 0)

0001 (başa 0 eklendikten sonra son iki bit 0⊕1 = 1)

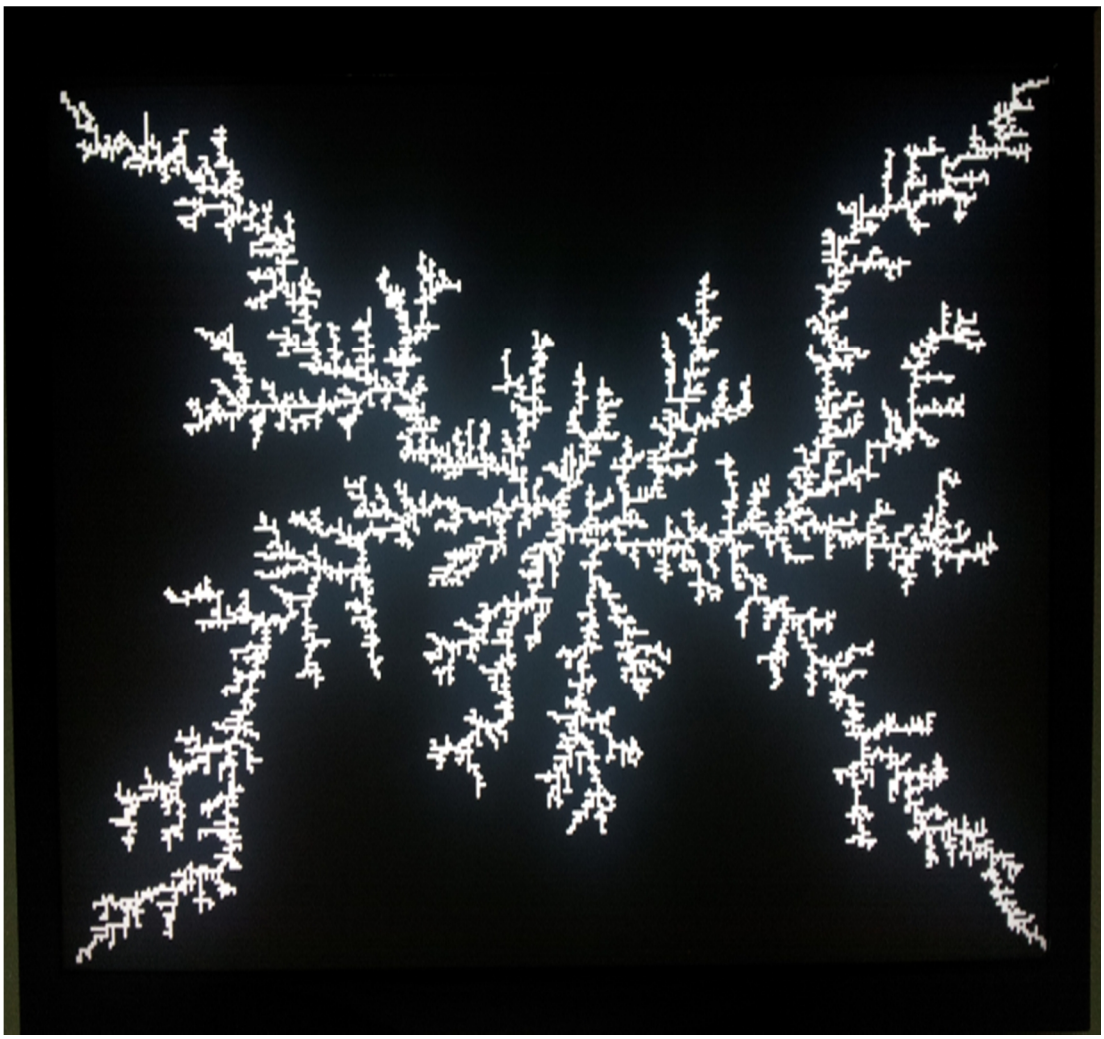
1000 (başa 1 eklendikten sonra son iki bit 0⊕0 = 0)

0100 -- 0010 – 1001 – 1100 – 0110 – 1011 – 0101 – 1010 – 1101 – 1110 – 1111

Döngü sisteminden anlaşılacağı üzere; fonksiyonun sonucunun başa eklenmesiyle devam eden döngü, yeniden 1111 durumuna gelinceye kadar devam eder. Döngü başlayıp son bulana kadar 15 farklı sayı rastgele olarak üretilmiştir.

Bir adet XOR kapısı ve 4 bitlik sayı kullanılarak oluşturulan bu sistem, doğrusal geri beslemeli kaydırma yazmacı ile oluşturulabilecek en basit sistemdir. Bu tez çalışmasında ise 31 bitlik sayı kullanılmıştır. 31 bitlik sayının başlangıç değeri olarak 1010101010101010101010101010101 sayısı atanmıştır. 2'lik sistemdeki bu sayı, Verilog kodu içinde 16'lık sayı sistemine göre 31'h55555555 olarak ifade edilir. Bu sayının 27. ve 30. bitleri XOR kapısının giriş elemanları kabül edilmiştir. XOR kapısının çıkış değeri ise, sayının en düşük bitine yani sıfırıncı bitine atanmıştır. Bu işlemin neticesinde rastgele sayılar elde edilmiştir. Elde edilen bu sayılar, difüzyonla kümeleşme modeli için gerçekleştirilen hareketin rastgele yönlerde olmasını sağlar.

FPGA donanımı üzerinde, Verilog dili kullanılarak gerçekleştirilen bu uygulamanın görüntüleri Şekil 4.3'de görülmektedir.



Şekil 4.3. Oluşturulan fraktal şekil

Oluşturulan fraktal görüntünün kırmızı, yeşil ve mavi renkler ile renklendirilmesi neticesinde oluşan yeni fraktal şekil Şekil 4.4'de gösterilmiştir.

Şekil 4.4. Renkli fraktal şekil

Gerçekleştirilen bu uygulama için hazırlanmış olan Verilog yazılımının, Quartus II programında derlenmesi sonucunda oluşan derleme raporu Şekil 4.5'deki gibidir. Bu rapordan da anlaşılacağı üzere, FPGA donanımında bulunan mantık kapılarının çok az bir bölümü kullanılarak fraktal şekil oluşturulabilir.

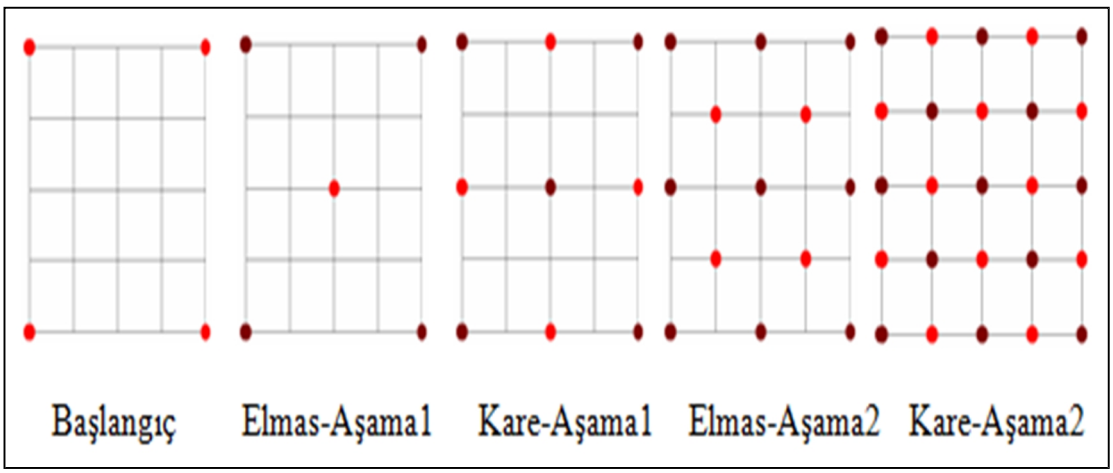
Flow Summary	
Flow Status	Successful - Tue Jun 11 17:06:25 2013
Quartus II 32-bit Version	11.1 Build 259 01/25/2012 SP 2 SJ Web Edition
Revision Name	DE2_Default
Top-level Entity Name	DE2_Default
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	354 / 33,216 (1 %)
Total combinational functions	343 / 33,216 (1 %)
Dedicated logic registers	190 / 33,216 (< 1 %)
Total registers	190
Total pins	425 / 475 (89 %)
Total virtual pins	0
Total memory bits	0 / 483,840 (0 %)
Embedded Multiplier 9-bit elements	0 / 70 (0 %)
Total PLLs	1 / 4 (25 %)

Şekil 4.5. Fraktal şekil oluşturma uygulaması ile ilgili derleme raporu

4.2. Elmas-Kare Algoritması ile Coğrafik Şekillerin Elde Edilmesi

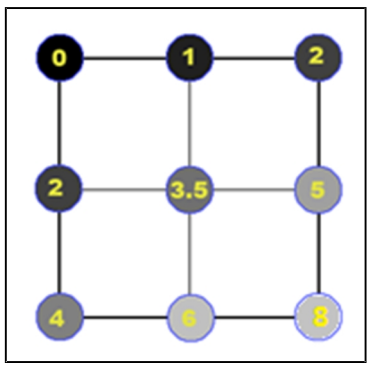
Uçuş simülasyonlarında, bilgisayar grafik ve oyunlarının oluşturulmasında, animasyon filmlerinde ve daha birçok alanda, doğada var olan şekillerin gerçektekine benzer görüntülerinin oluşturulmasına ihtiyaç duyulur. Bu görüntülerin elde edilmesi için, yükseklik ve doku eşlemelerinin gerçekleştirilmesi gereklidir. Yükseklik eşlemeleri, yükseklik ile ilgili verileri içerir. Doku eşlemeleri ise; bir, iki veya üç boyutlu matris olarak düzenlenen renk değerlerini içerir. Coğrafik şekillerden oluşan bir yeryüzü zemin görüntüsünün, diğer bir ifadeyle arazi resminin oluşturulması için, yükseklik ve renk değerleri belirlenir. Daha sonra çeşitli algoritmalar kullanılarak, elde edilmek istenilen resmin görüntüsü oluşturulur. Kullanılabilecek bu algoritmalardan biri 'elmas-kare' algoritmasıdır.

Elmas-kare algoritması 1986 yılında geliştirilmiş olup, bilgisayar grafiklerinin oluşturulmasında ve arazi resmi oluşturma uygulamalarında yaygın olarak kullanılan bir tekniktir [49]. Elmas-kare algoritmasının işleyişinin anlaşılabilmesi için; 5x5 bölmeden oluşan yani içinde 16 küçük kare bulunan bir büyük kare düşünelim. Başlangıçta bu karenin dört köşesine aynı olacak şekilde yükseklik değerleri atanır. Bu dört değerin ortalaması alınır ve bulunan sonuca rastgele bir değer eklenerek yeni bir değer elde edilir. Elde edilen bu değer, büyük karenin köşegenlerinin kesiştiği noktaya yani merkeze yerleştirilir. Bu işlem neticesinde; dört noktaya sahip karenin yerini, beş noktaya sahip bir elmas şekli alır. Bu aşamaya elmas oluşturma aşaması denir. İkinci aşamada ise elmas şeklin köşelerindeki noktaların ortalamasına rastgele bir değer eklenerek 4 küçük karenin köşeleri elde edilir. Bu aşamaya ise kare oluşturma aşaması adı verilir. Bu aşamalar tekrarlanarak devam eder. Algoritmanın diyagramı Şekil 3.7'de gösterilmiştir.



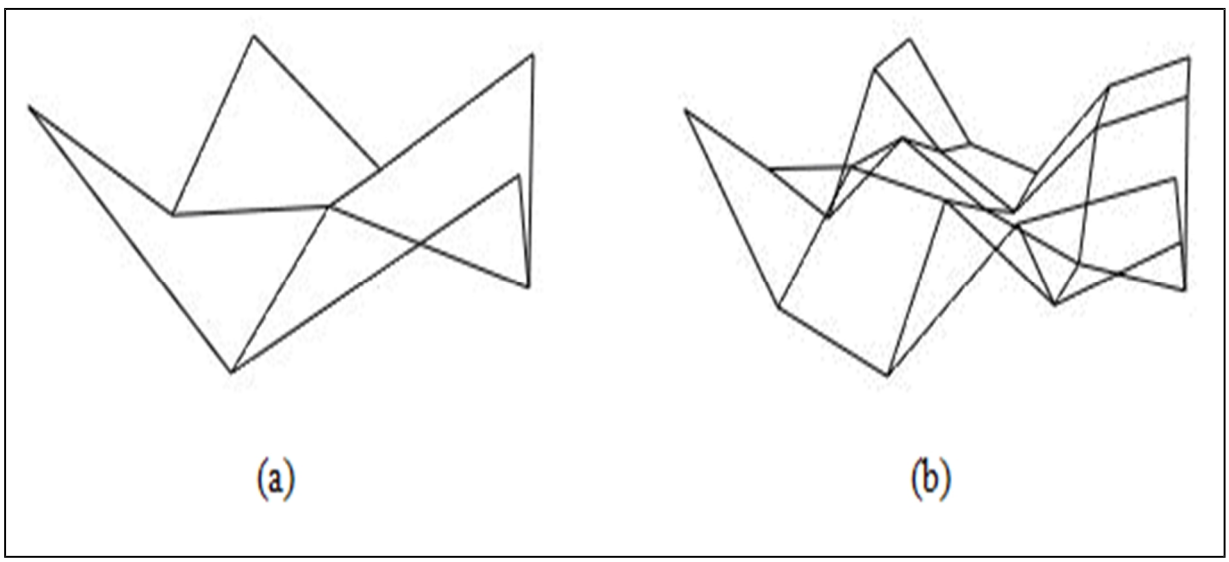
Şekil 4.6. Elmas-kare algoritmasının ilk iki aşaması

Algoritmanın işleyişine göre; ilk elmas-kare dönüşümünden sonra 4 adet küçük karenin köşe noktaları oluşmuştur. Bu işlemin ikinci tekrarında 16 adet küçük kare elde edilmiştir. Tekrarlama işlemi devam ettiği sürece küçük karelerin miktarı artacaktır. Algoirtmadaki en önemli işlem, yükseklik değerlerinin ortalama değerine rastgele (random) bir değer eklenmesidir. Örneğin; başlangıçta 4 köşesine yerleştirilen yükseklik değerleri aşağıdaki gibi olan bir karenin, önce elmas şekline dönüşmesi ve daha sonra dört adet küçük kareye dönüşmesi neticesinde meydana gelen yeni değerler Şekil 4.7'de verilmiştir. Bu işlemin bir sonraki tekrarı için, merkezde oluşan 3.5 birim miktarındaki yüksekliğe rastgele bir değer eklenir.



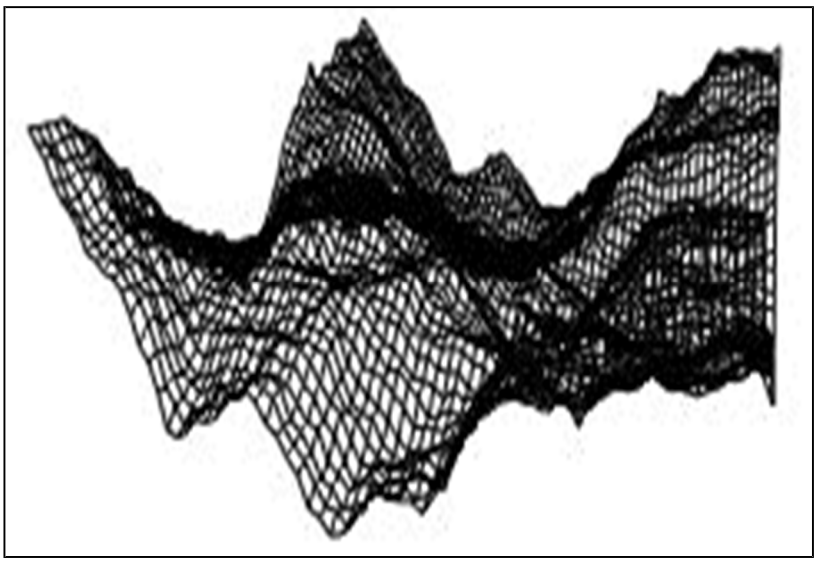
Şekil 4.7. Kare aşamasında sayı değerlerinin atanması

Birinci elmas-kare dönüşümünden sonra elde edilen 9 noktanın birbirlerine bağlanması ile Şekil 4.8.a'daki gibi bir yüzey şekli elde edilebilir. İkinci tekrarlamadan sonra gerçekleşen elmas-kare dönüşümü ile 16 küçük kare ve 25 nokta oluşur. Bu noktaların birleştirilmesiyle elde edilebilecek yüzey şekli ise Şekil 4.8.b'de görülmektedir.



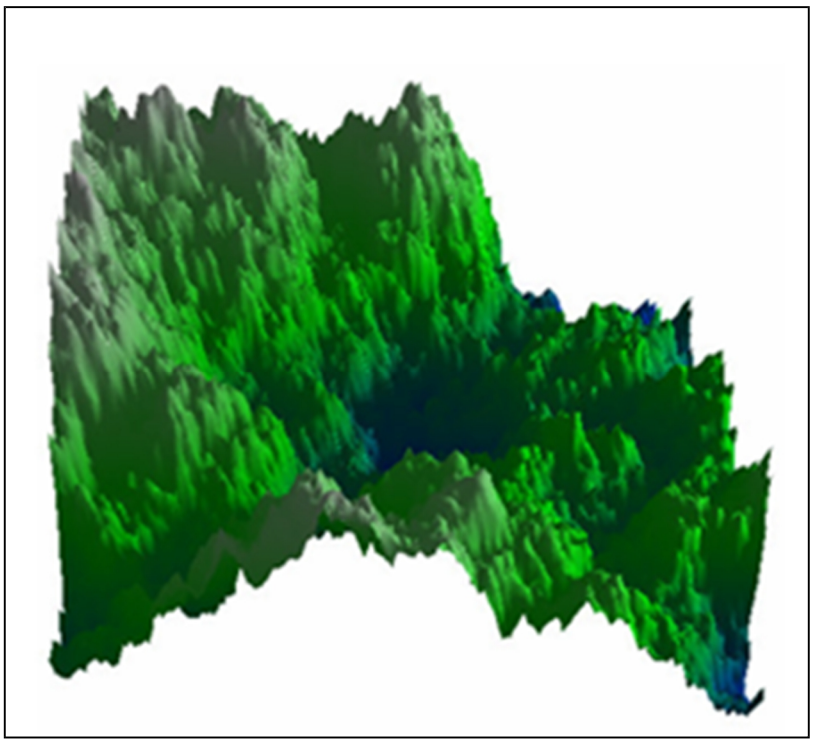
Şekil 4.8. Elmas-kare algoritması ile oluşan noktaların birbirlerine bağlanması

Bu işlemler tekrar edilmeye devam edilirse elde edilen şeklin yoğunluğu artmış olur. Örneğin, 5. tekrarlamadan sonra Şekil 4.9'daki gibi bir yüzey elde edilebilir. Sonuç olarak yükseklik değerleri atanmış bir şekil elde edilmiştir.



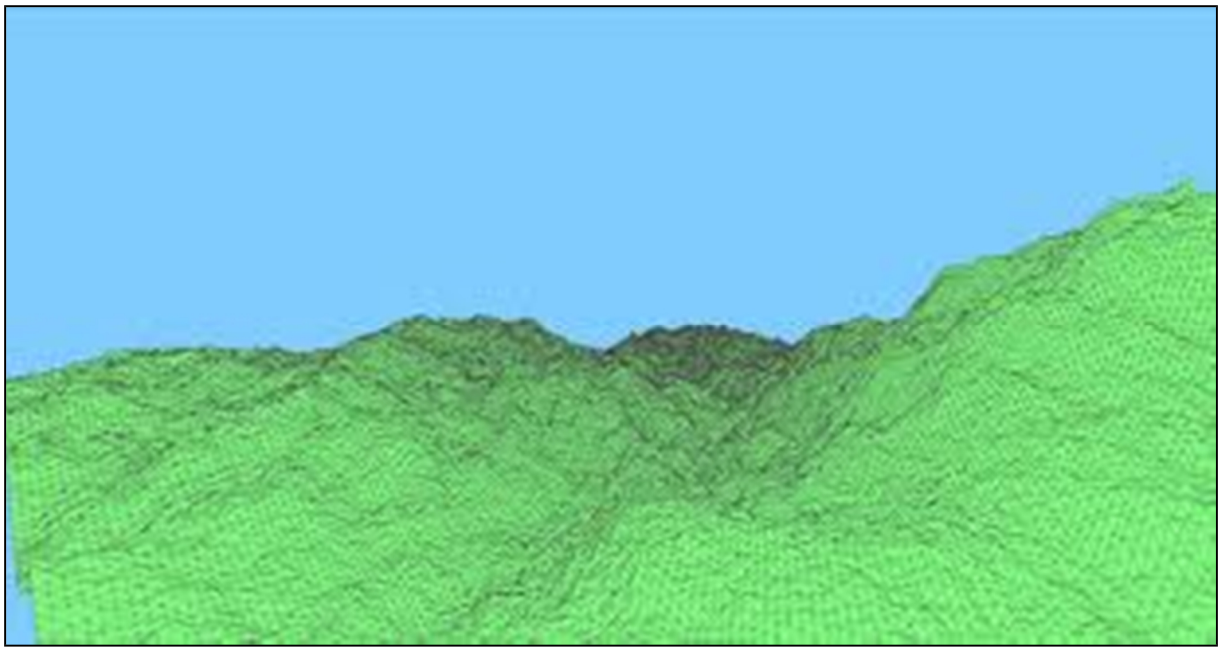
Şekil 4.9. Elmas-kare algoritmasındaki beşinci tekrarlama sonucu oluşan şekil [50]

Yüzey şekli oluşturulduktan sonra doku eşlemi gerçekleştirilir. Oluşturulan şeklin yeşil renkile renklendirilmesi yapıldıktan sonra, Şekil 4.10'da verilen görüntüler meydan gelir.

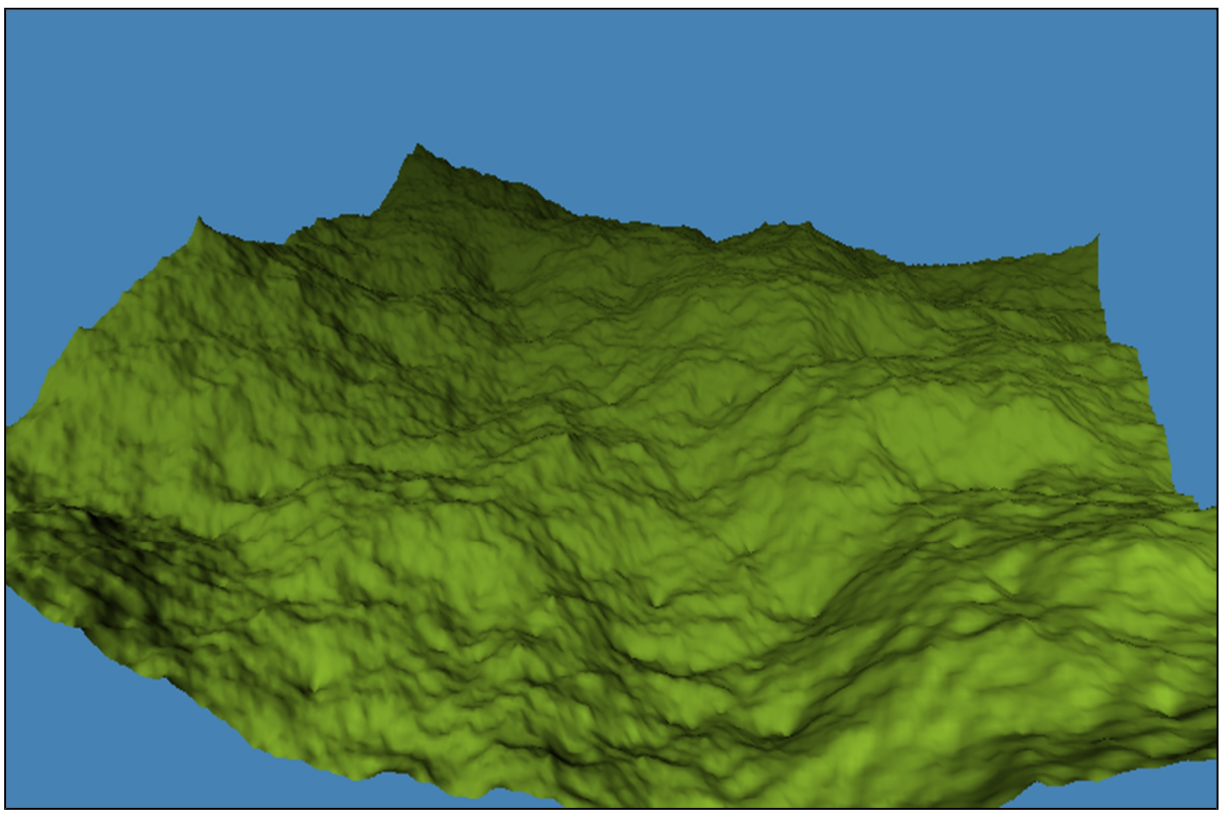


Şekil 4.10. Beşinci tekrarlama sonucu oluşan şeklin yeşil ile renklendirilmesi [51]

Bir manzara resminin oluşması için; dağ, ova, göl vb. yeryüzü şekillerinin bir veya birkaçının görüntüsünün oluşturulması gerekir. Bilgisayar ortamında, simülasyon programları aracılığı ile oluşturulan ve dağ görüntüsüne benzeyen bazı resimler Şekil 4.11 ve 4.12'de gösterilmiştir.

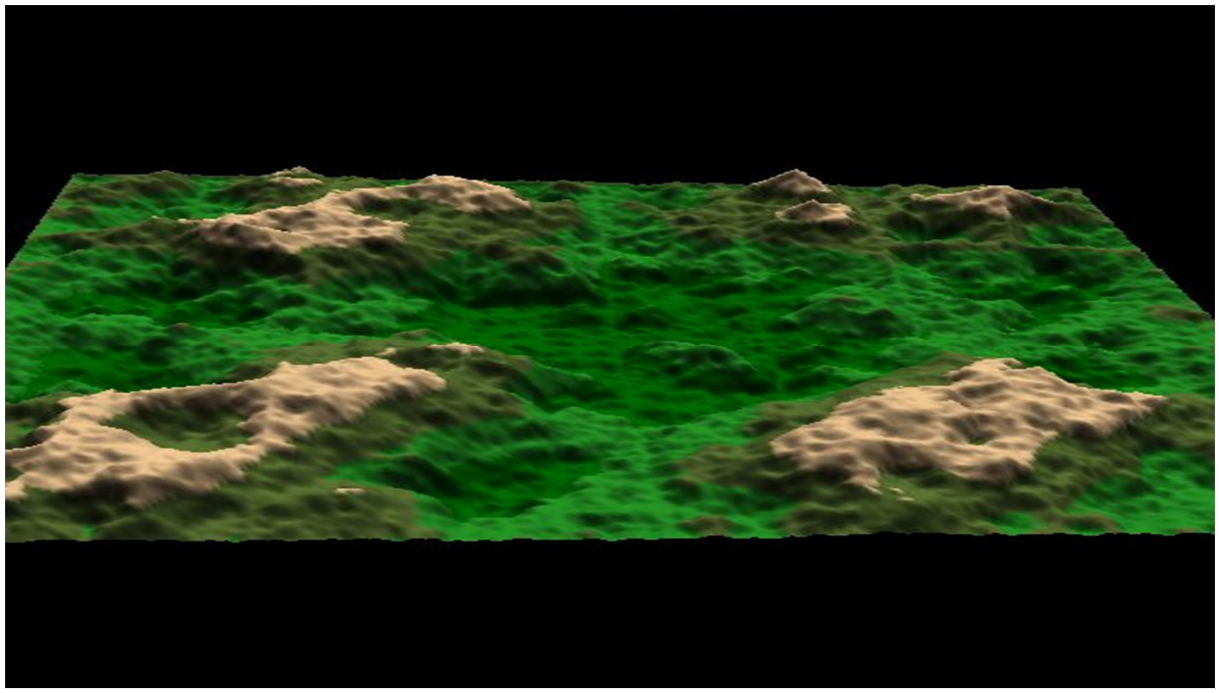


Şekil 4.11. Bilgisayar ortamında oluşturulan sembolik dağ resmi [52]



Şekil 4.12. Bilgisayar ortamında oluşturulan sembolik dağ resmi [53]

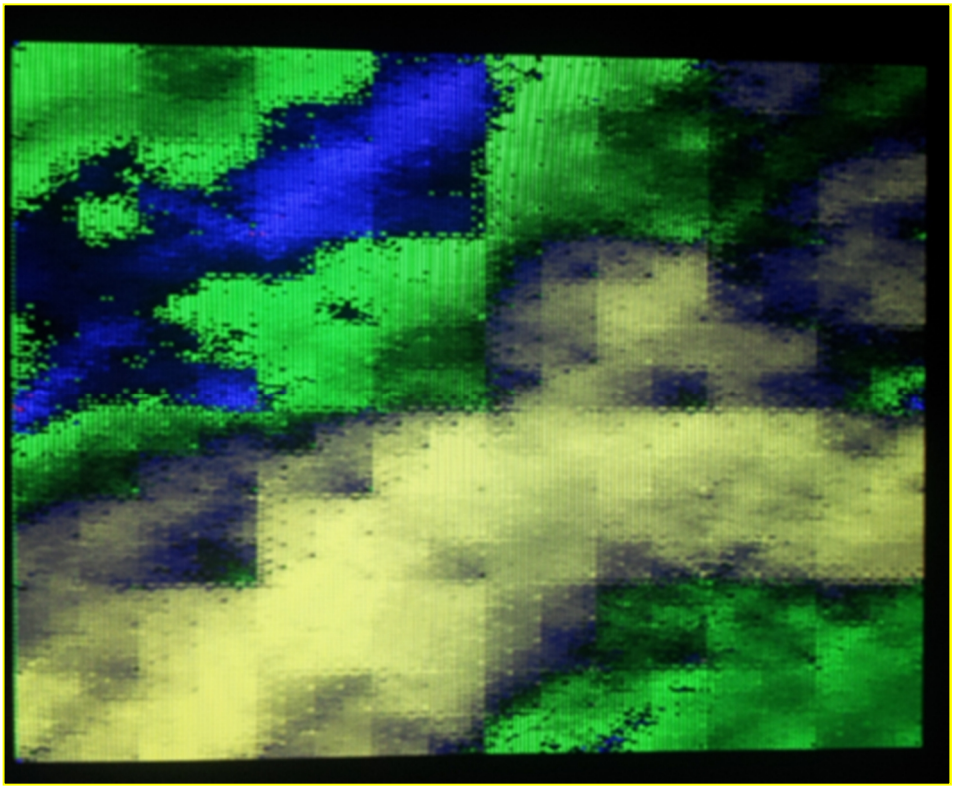
Yüksekliği fazla olan dağ görüntülerinin sarı renk, dağlardan daha alçakta bulunan ve vadileri temsil eden bölümlerin ise yeşil renk ile gösterildiği bir şekil ise Şekil 4.13'deki gibidir.



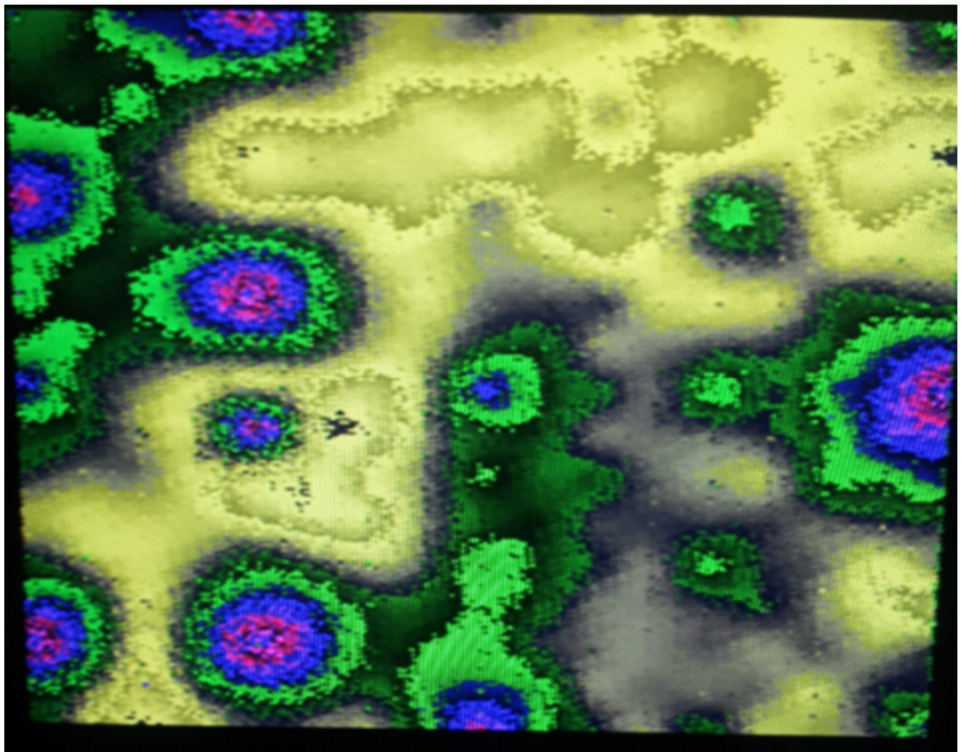
Şekil 4.13. Dağ ve vadi görünümüne sahip resim [54]

Bu tez çalışmasındaki uygulamada; dağ, vadi ve göl gibi farklı coğrafik şekiller, aynı resim içinde gerçek zamanlı olarak üretilmiştir. Elmas-kare algoritmasının kullanıldığı bu uygulamada, farklı yükseklik ve renk değerleri kullanılarak şekillerin gerçektekine benzer görüntülerinin oluşturulması sağlanmıştır.

Yukarıda bahsedildiği üzere, görüntü oluşturulurken yükseklik değerlerine rastgele değerler atanmaktadır. Bu durumda her bir uygulamada birbirine benzeyen fakat birbirinden farklı resimler elde edilmiştir. Bu resimlerden birinin görüntüsü Şekil 4.14'deki gibidir. Sarı renkten oluşan bölümler dağları, yeşil bölümler vadileri, mavi bölümler ise göl ve nehirleri temsil etmektedir. Şekil 4.15'de ise hem kırmızı renk hem de yeşil ile sarı renklerin farklı tonları resime eklenmiştir. Böylece 6 farklı renk değerine sahip görüntü elde edilmiştir.



Şekil 4.14. FPGA kullanılarak elde edilen ve 3 renkten oluşan resmin kuşbakışı görüntüsü



Şekil 4.15. FPGA kullanılarak elde edilen ve 6 renkten oluşan resmin kuşbakışı görüntüsü

4.3. Gauss Filtresi ile Görüntünün Netleştirilmesi

Görüntüdeki gürültünün yani bozucu etkinin yok edilmesi veya indirgenmesinde görüntü yumuşatma ve iyileştirme operatörleri kullanılır. Bir resim üzerinde düzleştirme ve iyileştirme işleminin yapılmasında kullanılan en önemli teknik filtreleme tekniğidir. Filtreler, çekirdek matris formunda gösterilirler. Boyutları 3x3, 5x5, 7x7, 9x9, 11x11 şeklinde olabilir. Adından da anlaşılacağı üzere, filtreler görüntüde belirli ayrıntıların filtrelenip ayıklanması ya da daha belirgin hale getirilmesi operasyonlarını gerçekleştiren operatörlerdir. Highpass filtresi, Gauss filtresi, Sinc filtresi, Sobel operatörü filtresi, Laplace filtresi gibi filtre çeşitleri vardır. Bu filtrelerden Gauss filtresi blur efektini artırır ve görüntüyü yumuşatır [55]. Filtre matrisi tanımlandığı amaca yönelik olarak görüntüde işleme dahil edilir. Gauss filtre ile sonsuz bir transfer fonksiyonuna karşılık mekansal alanda sonlu bir pencerede (tarama penceresi) filtreleme yapılabilmektedir. Bu da filtrelemenin temel problemini daha kolay çözülebilir hale getirir [56].

Bir önceki bölümde elde edilmiş olan resimler Gauss filtresi aracılığıyla yumuşatılarak pürüzsüz hale getirilebilir. 3 satır ve 3 sütundan oluşan Gauss Kernel filtresi Şekil 4.16'daki matris gibidir.

$$\begin{bmatrix} \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \\ \frac{2}{16} & \frac{4}{16} & \frac{2}{16} \\ \frac{1}{16} & \frac{2}{16} & \frac{1}{16} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} \\ \frac{2}{4} \\ \frac{1}{4} \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{4} & \frac{2}{4} & \frac{1}{4} \end{bmatrix}$$

Şekil 4.16. Gauss Kernel filtresi

Görüntü netleştirme işlemini FPGA kullanarak gerçekleştrimek için bu matrisin Verilog programlama dili ile ifade edilmesi gerekir. Bu ifadenin yazılımı aşağıdaki gibidir;

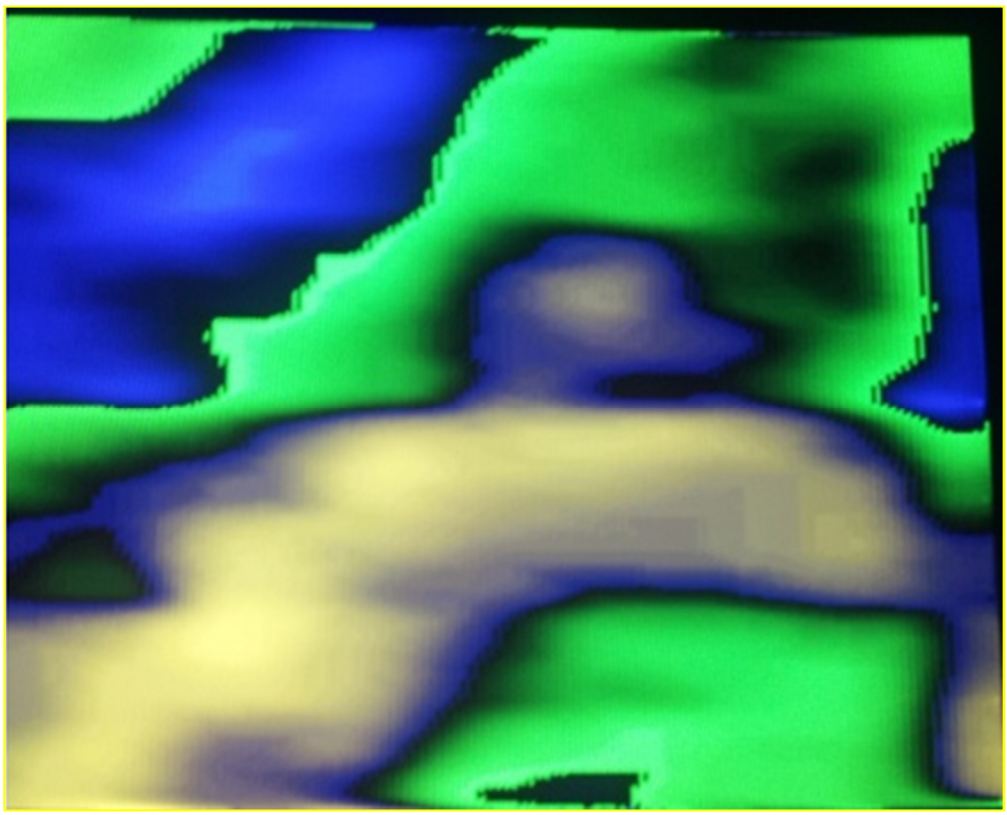
```
assign type1  = data0 + data2 + data6 + data8;
assign type2  = (data1 + data3 + data5 + data7) << 1;
assign type3  = data4 << 2;
assign result = (type1 + type2 + type3) >> 4;
```

Yazılan kodda belirtilen dataların matris içindeki konumları şu şekildedir;

$$\begin{bmatrix} Data0 & Data1 & Data2 \\ Data3 & Data4 & Data5 \\ Data6 & Data7 & Data8 \end{bmatrix}$$

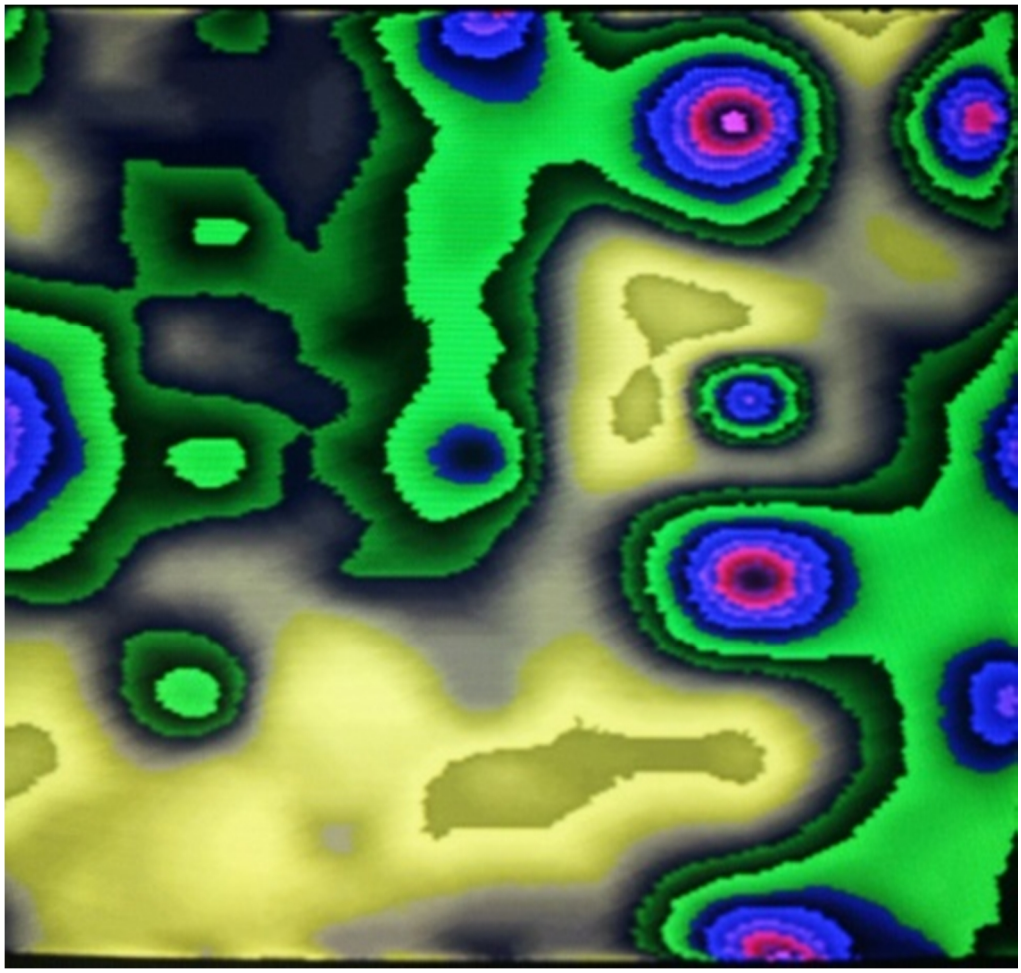
Kodda yer alan << 1 ifadesi 2 ile çarpım işlemini, <<2 ifadesi 4 ile çarpım işlemini belirtir. >> 4 ifadesi ise 16'ya bölme işlemini gösteririr.

Bu filtrenin Şekil 4.14'deki manzara resmine uygulanması ile meydana gelen yeni görüntü Şekil 4.17'deki gibidir.



Şekil 4.17. Üç renkten oluşan resmin netleştirilmiş görüntüsü

Filtrenin Şekil 4.15'deki resime uygulanması ile oluşan görüntü ise Şekil 4.18'deki gibidir.



Şekil 4.18. Altı renkten oluşan resmin netleştirilmiş görüntüsü

4.4. Üç Boyutlu Görüntü Oluşturulması

İkinci uygulamada elde edilmiş olan ve üçüncü uygulamada Gauss filtresi ile pürüzsüz hale getirilen resimler 2 boyutlu olarak görülmektedir. Bu resimlerin 3 boyutlu olarak görülmesi için x ekseni etrafında döndürülmeleri gerekir. Başka bir ifadeyle; Şekil 4.17 ve 4.18'deki görüntüler, manzara resimlerinin üstten görüntüleridir. Resim sadece x-y ekseni olarak yani en ve boy olarak görülmektedir. Yükseklik miktarı yani z ekseni görülmemektedir. Resmin x ekseni etrafında belli açı değerleri ile döndürülmesi neticesinde, görüntünün en-boy-yükseklik özellikleri net bir şekilde görülür. Bir görüntünün x ekseni, y ekseni ve z ekseni etrafında döndürülmesi için gerekli olan rotasyon matrisleri sırasıyla Şekil 4.19'daki gibidir.

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

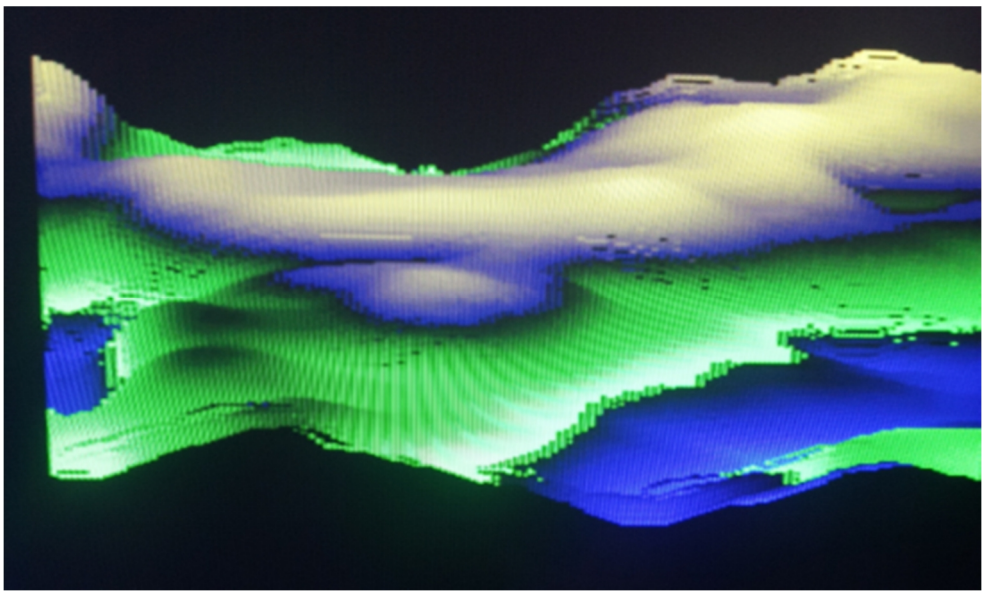
$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Şekil 4.19. Rotasyon matrisleri

Uygulama 2'de elde edilip, Uygulama 3'de netleştirilmiş olan manzara resimlerinde 'x' ve 'y' piksel değerini, 'z' ise yükseklik değerini gösteriyor olsun. Bu değerleri ifade eden matris A matrisi olarak isimlendirilir ise; matrisin değeri aşağıdaki gibi ifade edilebilir.

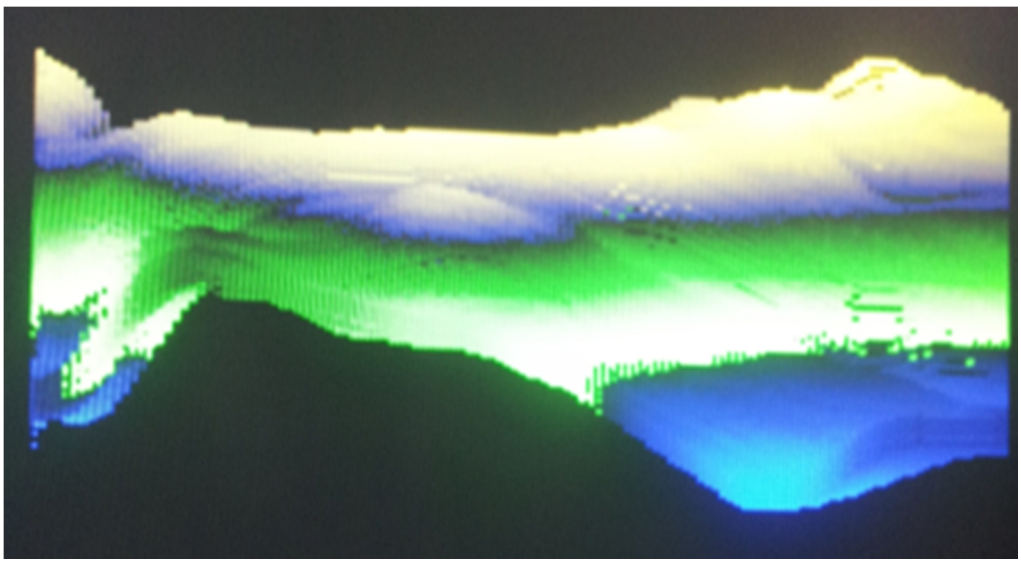
$$A = \begin{vmatrix} x \\ y \\ z \end{vmatrix}$$

Resmi x ekseni etrafında döndürmek istediğimiz için $R_x(\theta)$ matrisi ile A matrisi çarpılır. Bu işlem neticesinde x sabit kalırken, y ve z değişime uğrar. Fakat z değerinin yani yükseklik değerinin değişmesine gerek duyulmadığından, işlemlere z değeri değiştirilmeden devam edilir. Değişim neticesinde yeni y değeri; $y_{yeni} = y*\cos(\theta) - z*\sin(\theta)$ olur. Açı değerlerine göre sinüs ve kosinüs fonksiyonlarının değerinin ne olacağı, yazılan programda alt modüller içerisinde belirtilmiştir. Örneğin Şekil 4.17'deki resim 45 derecelik bir açıyla döndürülürse Şekil 4.19'daki görüntü elde edilir. Sarı renk ile belirtilen dağların yükseklik değeri bu açı ile bakıldığında daha net anlaşılmaktadır. Aynı şekilde vadiler dağın yamacında, göller ise zemin hizasında görülmektedir.



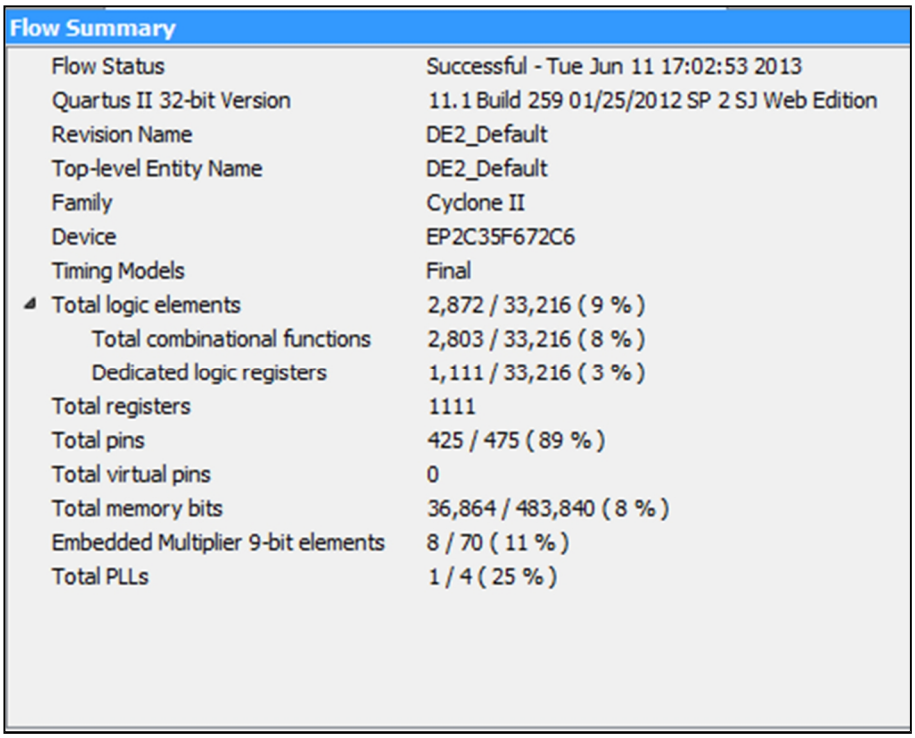
Şekil 4.20. Kuşbakışı görülen resmin 45 derece döndürülmüş görüntüsü

Resim 90 derecelik bir açıyla döndürülür ise Şekil 4.20'deki görüntü elde edilir. Dikkat edilirse bu görüntüde dağların arkasında bulunan yeşil bölge artık görülmemektedir.



Şekil 4.21. Kuşbakışı görülen resmin 90 derece döndürülmüş görüntüsü

Farklı renk ve yüksekliklerden oluşan görüntünün oluşturulması, oluşan görüntünün netleştirilmesi ve netleştirilen görüntünün belirli açılarla döndürülmesi işlemleri ile ilgili rapor Şekil 4.21'de verilmiştir. Rapordan görüleceği üzere kullanılan FPGA platformu 33216 adet lojik elementten oluşmaktadır. Yukarıda bahsedilen üç uygulamanın gerçekleşmesi için tasarlanan sistem ise 2872 lojik elementi kullanmıştır.



Flow Summary

Flow Status	Successful - Tue Jun 11 17:02:53 2013
Quartus II 32-bit Version	11.1 Build 259 01/25/2012 SP 2 SJ Web Edition
Revision Name	DE2_Default
Top-level Entity Name	DE2_Default
Family	Cyclone II
Device	EP2C35F672C6
Timing Models	Final
Total logic elements	2,872 / 33,216 (9 %)
Total combinational functions	2,803 / 33,216 (8 %)
Dedicated logic registers	1,111 / 33,216 (3 %)
Total registers	1111
Total pins	425 / 475 (89 %)
Total virtual pins	0
Total memory bits	36,864 / 483,840 (8 %)
Embedded Multiplier 9-bit elements	8 / 70 (11 %)
Total PLLs	1 / 4 (25 %)

Şekil 4.22. Uygulama 2, 3 ve 4 ile ilgili derleme raporu

4. SONUÇLAR

Bu tez çalışmasında gömülü bir sistem olan FPGA donanımı üzerinde bazı görüntü işleme algoritmalarının gerçeklenmesi hedeflenmiştir. İlk uygulamada, difüzyon ile sınırlı tanecik kümeleşme modelinin uygulanması neticesinde kendine benzeyen parçalardan oluşan fraktal şekil elde edilmiştir. İkinci uygulamada ise, farklı renk ve yükseklik değerlerinden oluşan yeryüzü şekillerinin gerçektekine benzer görüntülerinin elde edilmesi için elmas-kare algoritmasından yararlanılmıştır. Üçüncü uygulamada görüntü filtrelenmiştir. Görüntü iyileştirme algoritması olarak Gauss filtreleme tekniği kullanılmış ve görüntü netleştirme işlemi gerçekleştirilmiştir. Son uygulamada ise resmin farklı açılardaki görüntülerinin elde edilmesi için rotasyon matrisi kullanılmış ve yeni görüntüler elde edilmiştir.

Gömülü sistem donanımı olarak Altera firmasının üretmiş olduğu DE2 çalışma ve geliştirme platformu tercih edilmiştir. Algoritmalar donanım tanımlama dillerinden biri olan Verilog dili kullanılarak gerçeklenmiştir. Uygulamalarda kullanılan modüllerin çalışması için gerekli olan saat sinyalleri, FPGA üzerinde bulunan osilatörler ve Quartus II programında bulunan PLL birimleri aracılığı ile elde edilmişlerdir.

FPGA kullanılarak gerçekleştirilen görüntü işleme uygulamalarının yüksek hız ve performansa sahip olacağı bu çalışma neticesinde anlaşılmıştır. Fraktal şekil elde edilen birinci uygulamada FPGA'de bulunan kaynakların sadece yüzde 1'lik bölümü kullanılmıştır. Elmas-kare algoritması ile resim elde edilmesi, görüntü üzerinde filtreleme işleminin gerçekleştirilmesi ve görüntünün farklı açılarda döndürülmesi işlemlerinin toplamında ise FPGA mantık elementlerinin yüzde 9'luk bölümü kullanılmıştır. Bu yüzdelik oranları, yapılabilecek daha karmaşık görüntü işleme çalışmalarında FPGA kullanımının isabetli olacağını göstermektedir.

Tez çalışması süresince yapılan araştırmalar sayesinde gömülü sistemler ve donanım tanımlama dilleri hakkında bilgi edinilmiştir. Gerçekleştirilen uygulamalar neticesinde; görüntü işleme ve iyileştirme algoritmaları, FPGA donanımının yapısı ve özellikleri, uygulamalar için gerekli olan işlevsel birimler ve Verilog programlama dili konularında kişisel kazanım sağlanmıştır.

KAYNAKLAR

[1] Özçelik, F., Görüntü İşleme Algoritmalarının FPGA Üzerinde Gerçeklenmesi. Yüksek Lisans Tezi. Gazi Üniversitesi Bilişim Enstitüsü. Ankara. 48s (2012).

[2] Rodriguez, M., Sanchez-Perez, J., Gomez-Pulido, J.A. An FPGA Based Implementation for Median Filter Meeting the Real-Time Requirements of Automated Visual Inspection Systems, Proceedings of the 10th Mediterranean Conference on Control, 9-12 July 2002, Lisbon, Portugal. s.7 (2002).

[3] He, C., Papakonstantinou A., Chen, D., A Novel SoC Architecture on FPGA for Ultra Fast Face Detection, Proceedings of the 2009 IEEE International Conference on Computer Design, 2009, Nj, USA. s.412 (2009).

[4] Christe, S.A., Vignesh, M., Kandaswamy, A. An Efficient FPGA Implementation of MRI image filtering tumour characterization using Xilinx System Generator, *International Journal of VLSI Desing & Communication Systems*. 2 (4) : 95-109 (2011).

[5] Caner, H., FPGA Donanımı Üzerinde Araç Plakası Tanıma Sistemi. Yüksek Lisans Tezi. Hacettepe Üniversitesi Fen Bilimleri Enstitüsü. Ankara. 89s (2006).

[6] Sarı, V., Görüntü İşleme Sistemi Tasarımı ve Uygulaması. Yüksek Lisans Tezi. Hacettepe Üniversitesi Fen Bilimleri Enstitüsü. Ankara. 66s (2006).

[7] Yeniçeri, R., CMOS Görüntü Sensörü ve FPGA ile Sayısal Fotoğraf Makinesi Gerçeklenmesi. Lisans Tezi. İstanbul Teknik Üniversitesi Elektrik-Elektronik Mühendisliği. İstanbul. 41s (2007).

[8] Özcan, A.R., Gerçek Zamanlı Lineer Görüntü İşleme Algoritmalarının FPGA ile Gerçeklenmesi. Lisans Tezi. Yıldız Teknik Üniversitesi Elektrik-Elektronik Mühendisliği. İstanbul. 45s (2009).

[9] Taşcı, M., FPGA Kontrollü Robotik Göz. Yüksek Lisans Tezi. Balıkesir Üniversitesi Fen Bilimleri Enstitüsü. Balıkesir. 93s (2011).

[10] Özçelik, F., Görüntü İşleme Algoritmalarının FPGA Üzerinde Gerçeklenmesi. Yüksek Lisans Tezi. Gazi Üniversitesi Bilişim Enstitüsü. Ankara. 48s (2012).

[11] Gacar, A., FPGA Tabanlı Görüntü İşleme Arabirimi. Yüksek Lisans Tezi. Ege Üniversitesi Fen Bilimleri Enstitüsü. İzmir. 76s (2009).

[12] Tekdur, O., Sayısal Görüntü İyileştirme Algoritmalarının Geliştirilmesi ve Bu Algoritmaların Gerçek Zamanlı Gömülü Sistemlerde Gerçeklenmesi. Yüksek Lisans Tezi. Ege Üniversitesi Fen Bilimleri Enstitüsü. İzmir. 86s. (2012)

[13] Brown, S., Rose, J., 1992. Field Programmable Gate Arrays. Kluwer Academic Publishers. ISBN: 0-7923-9248-5, USA, 206s. (1992).

[14] Maxfield, C., 2008. FPGAs : Instant Access. Elseiver. ISBN: 978-0-7506-8974-8, UK, 216s. (2008).

[15] Brown, S., Rose, J., 1992. Field Programmable Gate Arrays. Kluwer Academic Publishers. ISBN: 0-7923-9248-5, USA, 206s. (1992).

[16] Wayne, W., 2004. FPGA-Based System Design. Pearson Education. ISBN: 8-1317-2465-4, USA, 544s. (2004).

[17] Parnell, K., Bryner, R., Comparing and Contrasting FPGA and Microporcessor System Design and Development. Xilinx White Paper Series. No.213. 32S. (2004).

[18] Maxfield, C., 2004. The Design Warrior's Guide to FPGAs. Elseiver. ISBN: 0-7506-7604-3, USA, 500s. (2004).

[19] Maxfield, C., 2004. The Design Warrior's Guide to FPGAs. Elseiver. ISBN: 0-7506-7604-3, USA, 500s. (2004).

[20] Maxfield, C., 2004. The Design Warrior's Guide to FPGAs. Elseiver. ISBN: 0-7506-7604-3, USA, 500s. (2004).

[21] Maxfield, C., 2004. The Design Warrior's Guide to FPGAs. Elseiver. ISBN: 0-7506-7604-3, USA, 500s. (2004).

[22] Taşcı,M., FPGA Kontrollü Robotik Göz. Yüksek Lisans Tezi. Balıkesir Üniversitesi Fen Bilimleri Enstitüsü. Balıkesir. 93s (2011).

[23] Bernd, J., 1997. Practical Handbook on Image Processing for Scientific Applications. CRS Press. ISBN: 0849389062, USA, 608s. (1997).

[24] Gonzales, R., 2004. Digital Image Processing Using Matlab. Dorling Kindersley. ISBN: 8177588982, USA, 620s. (2004).

[25] Sarı, V., Görüntü İşleme Sistemi Tasarımı ve Uygulaması. Yüksek Lisans Tezi. Hacettepe Üniversitesi Fen Bilimleri Enstitüsü. Ankara. 66s (2006).

[26] Mandelbrot, B., 1983. The Fractal Geometry of Nature. Henry Holt and Company. ISBN: 0716711869, USA, 468s. (1983).

[27] Fraktallar. URL (erişim tarihi: 05.04.2013) http://simurgistan.tr.gg/FRAKTALLAR.htm (2013)

[28] Fraktallar. URL (erişim tarihi: 05.04.2013) http://simurgistan.tr.gg/FRAKTALLAR.htm (2013)

[29] Koçak, K., Doğanın Geometrisi: Fraktal Geometri. URL (erişim tarihi: 01.04.2013) http://web.itu.edu.tr/~kkocak/fraktal_yazi.htm (2013)

[30] Fractal. URL (erişim tarihi: 03.04.2013) http://en.wikipedia.org/wiki/Fractal (2013)

[31] Fraktallar. URL (erişim tarihi: 05.04.2013) http://simurgistan.tr.gg/FRAKTALLAR.htm (2013)

[32] Fraktallar. URL (erişim tarihi: 05.04.2013) http://simurgistan.tr.gg/FRAKTALLAR.htm (2013)

[33] Altera Corp, Terasic Inc.

[34] Özçelik, F., Görüntü İşleme Algoritmalarının FPGA Üzerinde Gerçeklenmesi. Yüksek Lisans Tezi. Gazi Üniversitesi Bilişim Enstitüsü. Ankara. 48s (2012).

[35] Altera Corp, Terasic Inc.

[36] Özcan, A.R., Gerçek Zamanlı Lineer Görüntü İşleme Algoritmalarının FPGA ile Gerçeklenmesi. Lisans Tezi. Yıldız Teknik Üniversitesi Elektrik-Elektronik Mühendisliği. İstanbul. 45s (2009).

[37] Bilgisayar Monitörü. URL (erişim tarihi: 03.02.2013)

http://tr.wikipedia.org/wiki/Bilgisayar_monitörü (2013)

[38] Atay, Melike., FPGA Üzerinde Güneş Paneli ve Sıcaklık Kontrolü. Proje Tezi. Elektrik Mühendisleri Odası İstanbul Şubesi Proje Yarışması. İstanbul. 50s (2010).

[39] Verilog. URL (erişim tarihi: 10.12.2012) http://tr.wikipedia.org/wiki/Verilog (2012)

[40] Thomas, D., Moorby, P., 2002. The Verilog Hardware Description Language. Kluwer Academic Publishers. ISBN: 1-4020-7089-6, USA, 372s. (2002).

[41] Bayırlı, M., Mangan Dentritleri ve Difüzyon ile Sınırlı Kümeleşme Modeli. *Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi,* Balıkesir. Sayı 5.2, s.47-54.

[42] Bayırlı, M., İki Boyutlu Uzayda Tanecik Kümeleşmesinin İncelenmesi. Doktora Tezi. Gazi Üniversitesi Fen Bilimleri Enstitüsü. Ankara. (2003)

[43] Bülbül, M., Difüzyon ile Sınırlı Tanecik Kümeleşme Modeli Kullanılarak Elde Edilen Fraktallar İçin İki Boyutlu Ising Modelinin Creutze Cellular Automaton'ında Simülasyonu. Yüksek Lisans Tezi. Gazi Üniversitesi Fen Bilimleri Enstitüsü. Ankara. 85s. (2010)

[44] Bülbül, M., Difüzyon ile Sınırlı Tanecik Kümeleşme Modeli Kullanılarak Elde Edilen Fraktallar İçin İki Boyutlu Ising Modelinin Creutze Cellular Automaton'ında Simülasyonu. Yüksek Lisans Tezi. Gazi Üniversitesi Fen Bilimleri Enstitüsü. Ankara. 85s. (2010)

[45] Halsey, T., Diffusion-Limited Aggregation: A Model for Pattern Formation. *Physicstoday,* (36) 0031-9228. Ekim 2000.

[46] Bayırlı, M., Mangan Dentritleri ve Difüzyon ile Sınırlı Kümeleşme Modeli. *Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi,* Balıkesir. Sayı 5.2, s.47-54.

[47] Bülbül, M., Difüzyon ile Sınırlı Tanecik Kümeleşme Modeli Kullanılarak Elde Edilen Fraktallar İçin İki Boyutlu Ising Modelinin Creutze Cellular Automaton'ında Simülasyonu. Yüksek Lisans Tezi. Gazi Üniversitesi Fen Bilimleri Enstitüsü. Ankara. 85s. (2010)

[48] Şeker, S.E., Linear Feedback Shift Register URL (erişim tarihi: 01.05.2013) http://www.bilgisayarkavramlari.com/2010/05/19/lfsr-linear-feedback-shift-register/ (2013)

[49] Miller, G.S., The Definition and Rengering of Terrain Maps, Proceedings Siggraph '86 Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques, 1986, NY, USA, s.39-48, (1986)

[50] Martz, P., Generating Random Fractal Terrain. URL (erişim tarihi: 08.02.2013) http://www.gameprogrammer.com/fractal.html (2013)

[51] Martz, P., Generating Random Fractal Terrain. URL (erişim tarihi: 08.02.2013) http://www.gameprogrammer.com/fractal.html (2013)

[52] Andrews, T., Computer Graphichs and Fractal Terrain. URL (erişim tarihi: 10.03.2013) http://www.mtholyoke.edu/~andrews/cs331/assignments/assignment6.html (2013)

[53] Ogas, E., Diamond Square. URL (erişim tarihi: 05.03.2013) https://github.com/eogas/DiamondSquare/blob/master/README.md (2013)

[54] Ogas, E., Diamond Square. URL (erişim tarihi: 05.03.2013) https://github.com/eogas/DiamondSquare/blob/master/README.md (2013)

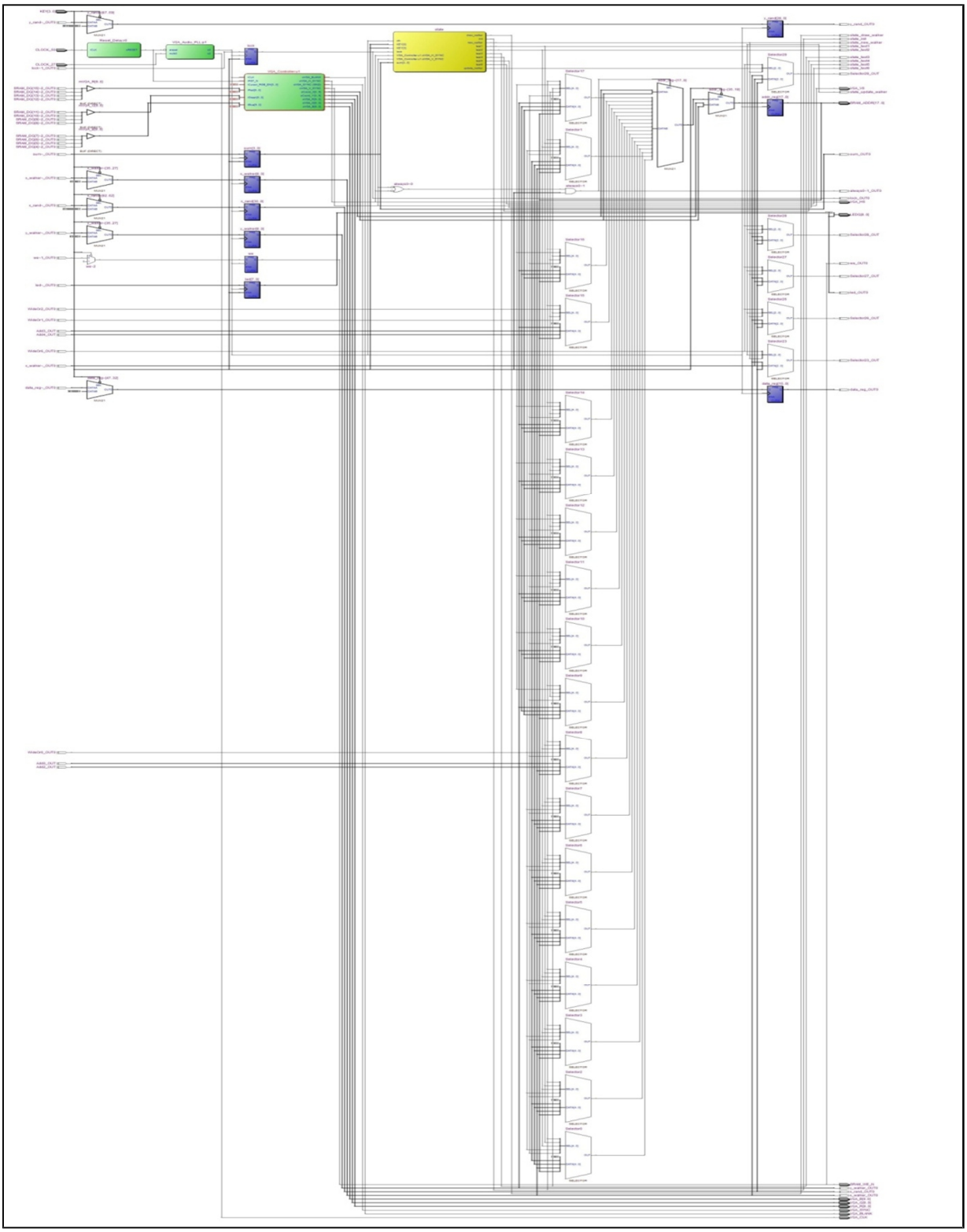
[55] Taşcı, M., FPGA Kontrollü Robotik Göz. Yüksek Lisans Tezi. Balıkesir Üniversitesi Fen Bilimleri Enstitüsü. Balıkesir. 93s (2011).

[56] Kuşcu, Ç., Antalya Aksu Bölgesi Tarım Alanlarında Ekspert Sınıflandırma Yöntemi ile Arazi Kullanımlarının Belirlenmesi. Yüksek Lisans Tezi. Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü. İstanbul. 93s (2005).

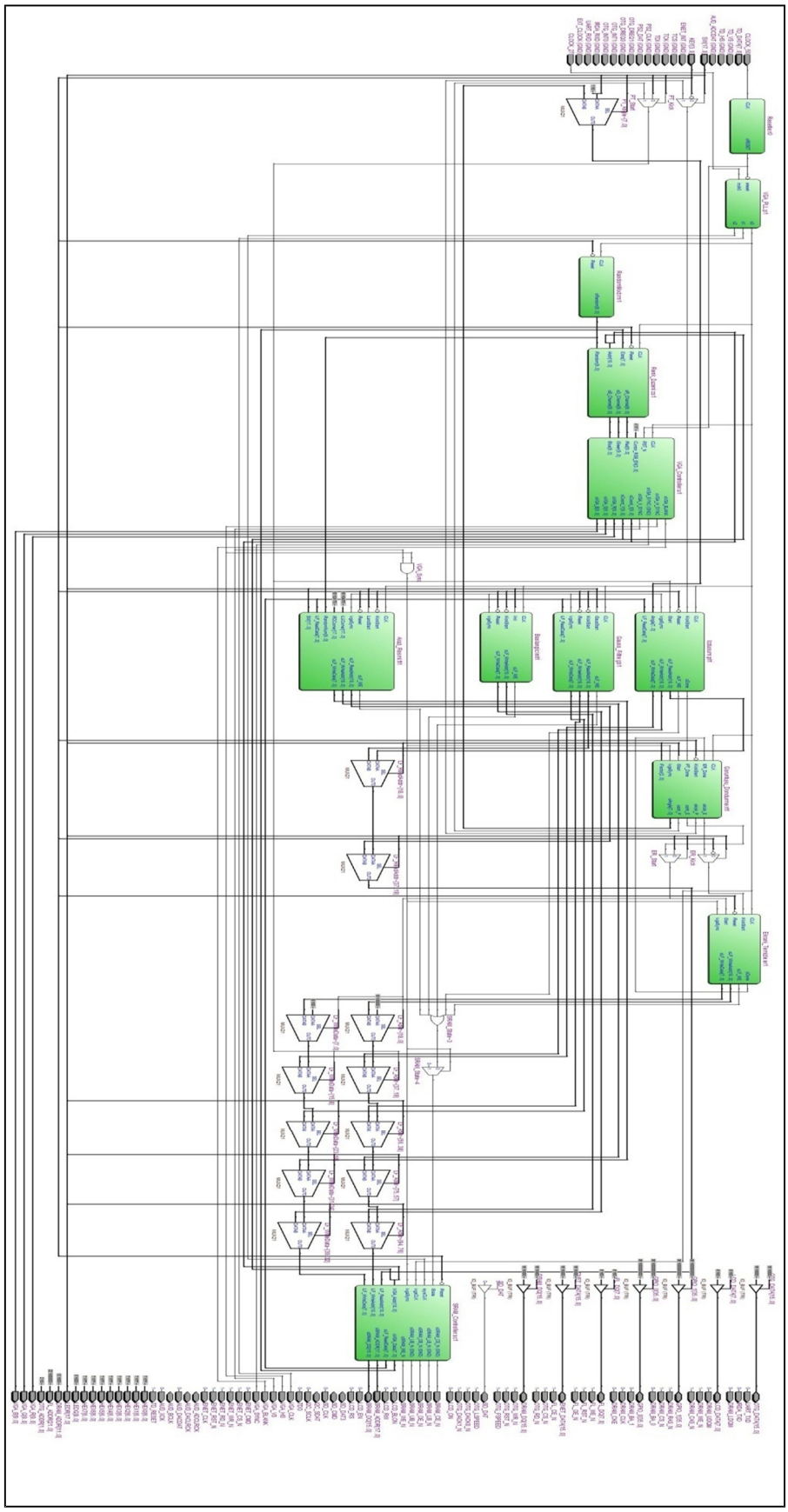
EKLER

EK-1 : Uygulamaların genel blok şemaları

Fraktal şekil oluşturma uygulamasının blok şeması;



Arazi resmi oluşturma uygulamasının blok şeması;



ÖZGEÇMİŞ

Kişisel Bilgiler

Adı, soyadı : ALİ RECAİ ÇELİK

Uyruğu : T.C.

Doğum tarihi ve yeri : 30424466212 - Diyarbakır

Medeni hali : Evli

Telefon : 05075478918

e-posta : eeealicelik@gmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek Lisans	KSÜ, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Müh.	2013
Lisans	Gaziantep Üniversitesi / Elektrik-Elektronikr Müh.	2011
Lise	Diyarbakır Anadolu Lisesi	2006

İş Deneyimi

Yıl	Yer	Görev
2012-2013	Zirve Üniversitesi	Araştırma Görevlisi

Yabancı Dil

İngilizce

Hobiler

Futbol, Yüzme