

İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ

DCT TANIM KÜMESİNDEKİ DAMGALAMA İŞLEMLERİ İÇİN
YAZILIM/DONANIM ORTAKLI SİSTEM TASARIMI

BİTİRME ÖDEVİ
AHMET TURAN EROZAN
040080341

Bölümü: Elektronik ve Haberleşme Mühendisliği Bölümü

Programı: Elektronik Mühendisliği

Danışmanı: Doç. Dr. Sıddıka Berna ÖRS YALÇIN

MAYIS 2013

ÖNSÖZ

Bitirme ödevim süresince yardımını esirgemeyen, değerli vaktini ayıran, bilgi ve tecrübesiyle yol gösteren ve destek veren değerli hocam Doç. Dr. Sıddıka Berna ÖRS Yalçın'a, Gömülü Sistem Tasarımı Laboratuvarı çalışanlarına ve arkadaşlarıma teşekkürlerimi sunarım.

Ayrıca, hayatım boyunca destekleriyle hep yanımda olan aileme teşekkürlerimi sunarım.

Mayıs 2013

Ahmet Turan EROZAN

İÇİNDEKİLER

KISALTMALAR.....	v
ŞEKİL LİSTESİ.....	vi
ÖZET.....	vii
SUMMARY.....	viii
1. GİRİŞ.....	1
2. Önbilgiler.....	3
2.1.Damgalama.....	3
2.2.Görüntü Damgalama.....	3
2.3.Sahada Programlanabilir Kapı Dizileri.....	4
2.4.Xilinx Spartan-6 LX45 FPGA.....	7
2.5.DCT ve IDCT.....	7
2.6.Xilinx ISE Programı.....	9
2.7.Xilinx EDK Programı.....	10
2.8.Xilinx SDK Programı.....	12
2.9.Microblaze İşlemcisi	14
3. FPGA ÜZERİNDE SİSTEM TASARIMI.....	16
3.1.Kayıpsız Tasarım.....	16
3.2.Kayıplı Tasarım.....	16
4. DONANIM TASARIMI.....	18
4.1.DCT 1 Donanımı.....	18
4.2.DCT 2 Donanımı.....	19
4.3.IDCT 1 Donanımı.....	20
4.4.IDCT 2 Donanımı.....	21
4.5.Donanımların Zaman Analizi.....	23
5. YAZILIM.....	24
6. DAMGALAMA YÖNTEMİ.....	25
7. SONUÇLAR.....	27
KAYNAKLAR.....	28
ÖZGEÇMİŞ.....	29

KISALTMALAR

FPGA	: Field Programmable Gate Array
DCT	: Discrete Cosine Transform
IDCT	: Inverse Discrete Cosine Transform
EDK	: Embedded Development Kit
ISE	: Integrated Synthesis Environment
XPS	: Xilinx Platform Studio
RISC	: Reduced Instruction Set Computer

ŞEKİL LİSTESİ

Şekil 2.1 : Damgalama İşleminin Genel Yapısı	3
Şekil 2.2 : Görüntü Damgalama Yöntemleri	4
Şekil 2.3 : Mantık Hücresi Yapısı	5
Şekil 2.4 : FPGA'nın İç Yapısı	6
Şekil 2.5 : Kullanılan FPGA Kartı	7
Şekil 2.6 : Kullanılan DCT Matrisi	8
Şekil 2.7 : Kullanılan DCT Matrisinin Transpozu	9
Şekil 2.8 : DCT ve IDCT Uygulanışı	9
Şekil 2.9 : Xilinx ISE Programının Görünümü	10
Şekil 2.10 : EDK sistem geliştirme araçları	11
Şekil 2.11 : Xilinx EDK Programının Görünümü	12
Şekil 2.12 : Sistem tasarımı akışı	13
Şekil 2.13 : Xilinx SDK Programının Görünümü.....	14
Şekil 2.14 : Microblaze mimarisi	15
Şekil 3.1 : Sistemin genel yapısı.....	17
Şekil 4.1 : DCT 1 donanımının iç yapısı.....	18
Şekil 4.2 : DCT 1 donanımının dış yapısı.....	19
Şekil 4.3 : DCT 1 donanımının benzetimi.....	19
Şekil 4.4 : DCT 2 donanımının dış yapısı.....	20
Şekil 4.5: DCT 2 donanımının benzetimi.....	20
Şekil 4.6: IDCT 1 donanımının dış yapısı.....	21
Şekil 4.7: IDCT 1 donanımının benzetimi.....	21
Şekil 4.8: IDCT 2 donanımının dış yapısı.....	22
Şekil 4.9: IDCT 2 donanımının benzetimi.....	22
Şekil 4.10: Donanımların maksimum çalışma frekansı.....	23
Şekil 5.1: DCT ile damgalama işlemi.....	24
Şekil 6.1: Lena görüntüsünün damgalamadan önceki ve sonraki görünümü.....	25
Şekil 6.2: Airplane görüntüsünün damgalamadan önceki ve sonraki görünümü.....	25
Şekil 6.3: Damga çıkarım başarı yüzdeleri.....	26

DCT TANIM KÜMESİNDEKİ DAMGALAMA İŞLEMLERİ İÇİN YAZILIM/DONANIM ORTAKLI SİSTEM TASARIMI

ÖZET

Elektronik mühendisliği alanı her geçen gün gelişmektedir. Bu gelişim günlük yaşamdaki kullanım alanını da artırmaktadır. Bu kullanım alanlarından biri de sayısal verilerin kullanılması ve paylaşılmasıdır. Görüntü, ses, yazı, video gibi sayısal verilerin paylaşılması ve yayılması izinsiz çoğaltma ve kullanma gibi problemleri ortaya çıkarmıştır.

Sayısal damgalama, sayısal verilerin izinsiz çoğaltılması ve kullanılması problemini çözmek amacıyla kullanılan yöntemlerden biridir. Damgalama işlemi, sayısal verinin içine sadece yetkili kişilerin çıkarabileceği bir verinin gizlenmesi ile yapılır. Sayısal görüntü damgalamada bu veri sayısal bir görüntü içine gizlenerek damgalama işlemi yapılır. Sayısal görüntü damgalama yöntemleri uzaysal tanım kümesinde ve frekans tanım kümesinde olmak üzere ikiye ayrılır.

Sayısal görüntü damgalamanın geliştirilmesi için çeşitli yöntemler geliştirilmektedir. Yöntemler geliştirilirken iki kısıt öne çıkmaktadır. Bunların ilki gizlenen verinin görüntüde gözle görülür bir değişiklik yapmasının istenmemesidir. Diğer kısıt ise görüntü üzerinde yapılacak çeşitli işlemlerin gizlenen veriyi bozmamasıdır.

Ayrık Kosinüs dönüşümü damgalamanın görüntüde gözle görülür değişiklik yapmaması ve görüntü üzerinde yapılacak çeşitli işlemlerden etkilenmemesi için geliştirilen yöntemlerden biridir. Bu tez çalışmasında, damgalama yöntemi olarak ayrık kosinüs dönüşümü kullanılmıştır. Öncelikle MATLAB ortamında DCT işlemi gerçekleştirilmiştir. Sayısal görüntü damgalamadaki kısıtları sağladığı gözlemlendikten sonra Xilinx ISE ortamında DCT ve IDCT işlemlerini gerçekleyen donanımlar tasarlanmıştır. Bu donanımlar Xilinx EDK ortamında Xilinx Microblaze yazılımsal işlemcisi ile birleştirilmiştir. Böylece DCT tanım kümesindeki yapılacak sayısal görüntü damgalama işlemleri için yazılım/donanım ortakli sistem tasarımı gerçekleştirilmiştir. Bu sistem FPGA üzerinde gerçekleştirilerek örnek görüntüler tasarlanan sistemde damgalanmıştır. Böylece sistemin başarısı test edilmiştir.

HARDWARE/SOFTWARE CODESIGN FOR WATERMARKING IN DCT DOMAIN

SUMMARY

Electronics area has developed day by day. This development has increased usage area in daily life. One of this usage areas is digital data usage and share. Sharing and publishing digital data such as image, audio, writing, video has brought out problems like unauthorised reproduction and use.

Digital watermarking is one of the using method to solve the problem of unauthorised reproduction and use of digital data. Watermarking is done by hiding the data that only authorised person can remove. In digital image watermarking, watermarking is done by hiding this data into a digital image. Digital image watermarking methods are seperated two groups in spatial domain and frequency domain.

Several methods have developed for improving digital image watermarking. Two constraint become prominent in developing methods. The first of them is that watermarking do not do noticeable change on image. The other one is that several watermarking

Discrete cosine transform is one of the method developed to not do noticeable change on image and not affected from various operation on image. In this study, discrete cosine transform was used as watermarking method. First of all, DCT was verified in MATLAB platform. After observing that it provides constraints of digital image watermarking, DCT and IDCT hardwares are implemented using Xilinx ISE platform. These hardwares are put together with Xilinx softcore processor Microblaze in Xilinx EDK platform. Thus, hardware/software codesign system for watermarking in DCT domain is implemented. This system is realized on FPGA and sample images are watermarked by designed system. By this way, system success is tested.

1. GİRİŞ

Sayısal görüntü ve ses verilerinin yaygınlaşması ile bu verilerin telif haklarını koruma konusundaki uygulamalara büyük bir ihtiyaç doğmuştur[1]. Bu ihtiyaç için geliştirilen şifreleme uygulamalarında sadece geçerli anahtar sahiplerinin veriye ulaşmasına izin verilir. Ancak verinin şifresinin çözülmesiyle birlikte verinin çoğaltılması veya paylaşılmasını takip etmenin hiçbir yolu bulunmamaktadır. Bu sebeple şifreleme, izinsiz çoğaltmayı önlemeye çalışan yayıncılara yeteri kadar koruma sağlayamaz. Sayısal damgalama kriptografik süreçleri tamamlamayı amaçlar. Veriye gömülen ayırt edici bilgi (damga) herhangi bir şifre çözme işleminden sonra da olduğu gibi veri içinde durmalı ve telif hakkı bu ayırt edici bilgi kullanılarak korunabilmedir[1].

Bu çalışmada sayısal görüntülerdeki telif hakkı koruma işlemleri için sayısal görüntü damgalama işlemi ele alınmaktadır. Literatürde görüntü damgalama işlemi için birçok yöntem geliştirilmiştir[2]. Sayısal damgalama yöntemleri uzamsal ve frekans tanım kümesi olarak iki uygu-lama alanına ayrılmaktadır. Uzamsal tanım kümesinde piksel değerlerinde değişiklikler yapılarak çeşitli yöntemler geliştirilmiştir. Frekans tanım kümesinde ise genellikle DCT, Hızlı Fourier Dönüşümü (Fast Fourier Transform - FFT) ve Ayrık Dalgacık Dönüşümü (Discrete Wavelet Transform - DWT) kullanılarak çeşitli yöntemler geliştirilmiştir[2].

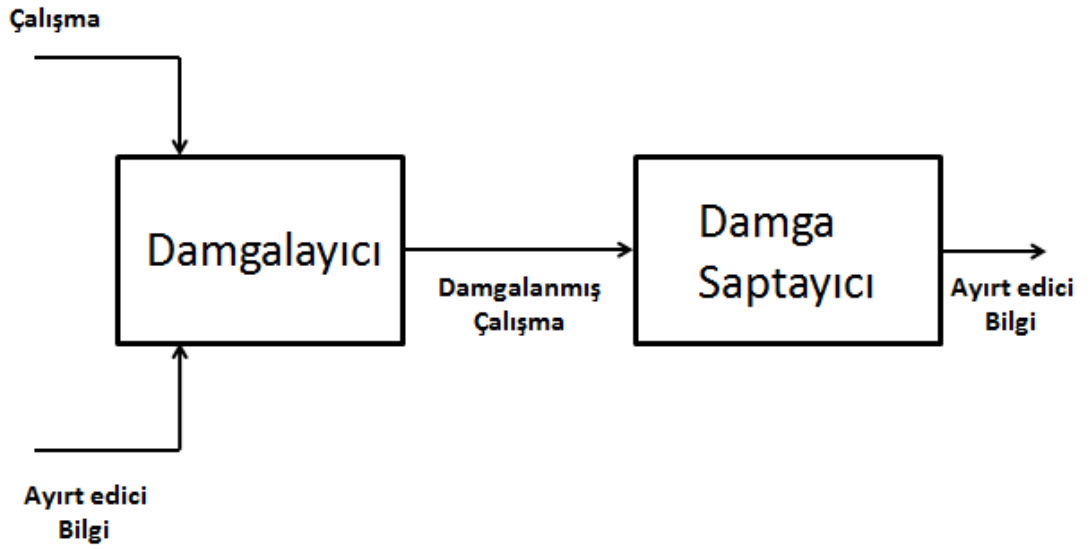
Birleşik Fotoğraf Uzmanları Grubu (Joint Photo-graphic Experts Group – JPEG) görüntü sıkıştırma işleminin damgaya olan etkisinden dolayı görüntü damgalama işlemlerinde sıkıştırmaya karşı dayanıklılık önemli bir çalışma konusu olmuştur [3]. JPEG sıkıştırma işleminde DCT kullanılmaktadır. Bundan dolayı damgalama işleminde DCT kullanılması sıkıştırmaya karşı dayanıklılığı sağlamaktadır[3]. Yapılan çalışmada bu durum göz önünde bulundurularak DCT kullanılarak yapılan damgalama işlemleri için donanım-yazılım ortaklı sistem tasarımı yapılmıştır. Xilinx ISE programı kullanılarak Ayrık Kosinüs Dönüşümü (Discrete Cosine Transform - DCT)ve Ters Ayrık Kosinüs Dönüşümü (Inverse Discrete Cosine Transform - IDCT) donanımları gerçekleştirilmiştir. Kullanılacak algoritmanın gerçekleştirilmesi için Xilinx

MicroBlaze yazılımsal işlemcisi kullanılarak istenilen yöntemin yazılım ile gerçekleştirilmesi sağlanmıştır. Dönüşüm işlemleri tasarlanan donanımlarda paralelleştirilerek hız kazanılmıştır. Damgalama işlemi MicroBlaze işlemcisi üzerinde gerçekleştirilerek istenilen algoritmanın donanım değiştirilmeden, yazılımla kolayca uygulanması sağlanmıştır.

2. ÖNBİLGİLER

2.1. Damgalama

Yazı, ses, görüntü gibi çeşitli çalışmaların telif hakkını koruma amaçlı o çalışmalara ayırt edici bilgi gömülmesine damgalama denir[7].



Şekil 2.1 : Damgalama İşleminin Genel Yapısı

2.2. Görüntü Damgalama

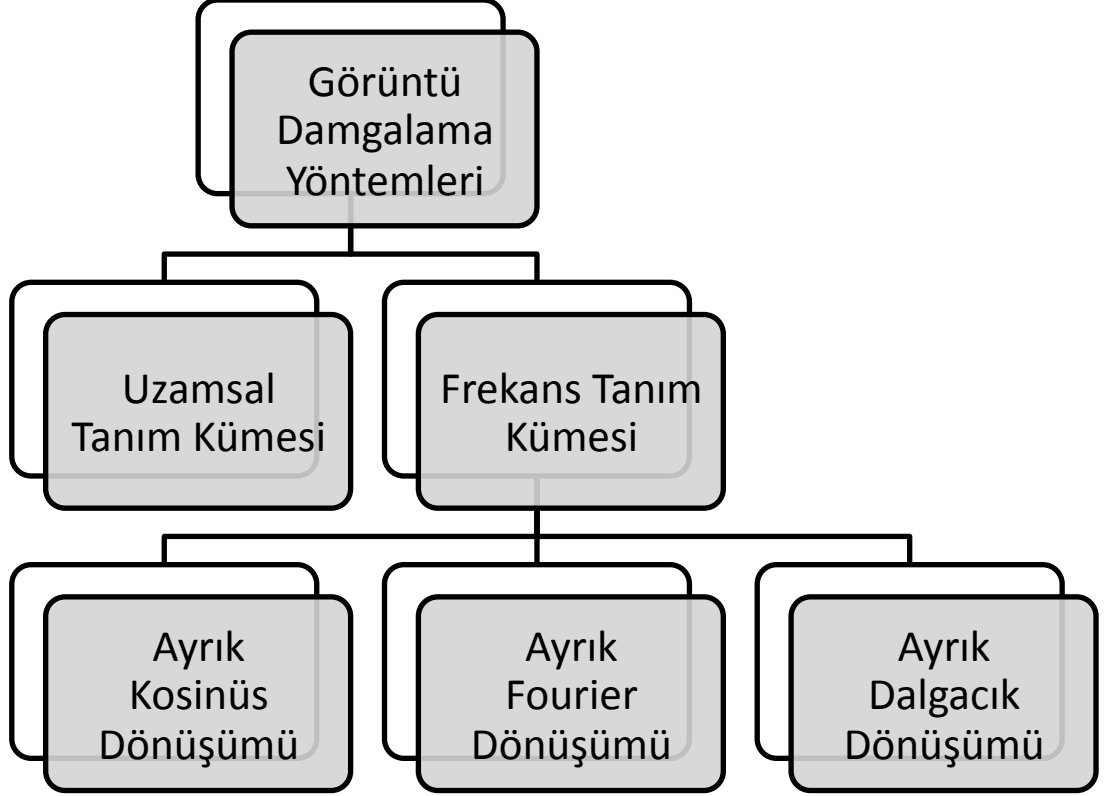
Şekil 2.1'deki damgalayıcıya giren çalışma verisinin görüntü verisi olduğu damgalamadır. Görüntü damgalamanın genel işlem yapısı denklem 1.1'de belirtilmiştir.

$$G'=G + f(G, V) \quad (1.1)$$

Burada G' damgalanmış görüntüyü, G damgalanacak görüntüyü, V gizlenecek veriyi, f damgalama işleminde kullanılan yöntemi göstermektedir.

Görüntü damgalama işleminde kullanılacak yöntemin belirli özellikleri sağlayabilmesi gerekir. Bu özellikler aşağıda belirtilmiştir[4].

- Damgalama işleminin görüntünün görünümünü bozmaması gerekmektedir[4].
- Görüntüye gömülen damganın, filtreleme, kesme, kayıplı sıkıştırma, büyütme, küçültme gibi çeşitli görüntü işleme işlemlerine karşı dayanıklı olması gerekmektedir[4].



Şekil 2.2: Görüntü Damgalama Yöntemleri [7].

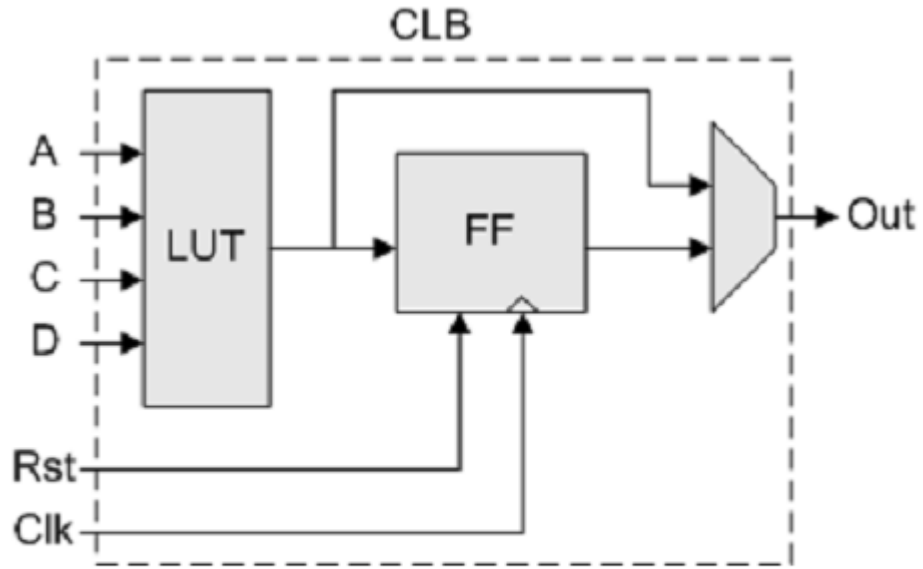
Şekil 2.2’de görüldüğü gibi görüntü damgalama yöntemleri iki farklı tanım kümesinde yapılmaktadır. Bunlar Uzamsal tanım kümesi ve frekans tanım kümesidir. Uzamsal tanım kümesinde yapılan damgalama işlemleri piksel değerlerinin değiştirilmesi ile yapılmaktadır. Frekans tanım bölgesinde ise görüntüye dönüşüm işlemi uygulanmakta ve bunun sonucunda oluşan katsayılarla damgalama işlemi yapılmaktadır. Literatürdeki dönüşüm işlemleri Ayrık Kosinüs Dönüşümü, Ayrık Fourier Dönüşümü ve Ayrık Dalgacık dönüşümüdür.

2.3. Sahada Programlanabilir Kapı Dizileri

Sahada Programlanabilir Kapı Dizileri herhangi bir sayısal fonksiyonu gerçekleştirebilmek için kullanıcı tarafından programlanabilen tümleşik devrelerdir [9]. FPGA yönetilebilir anahtarların ve programlanabilir mantık hücrelerinin iki

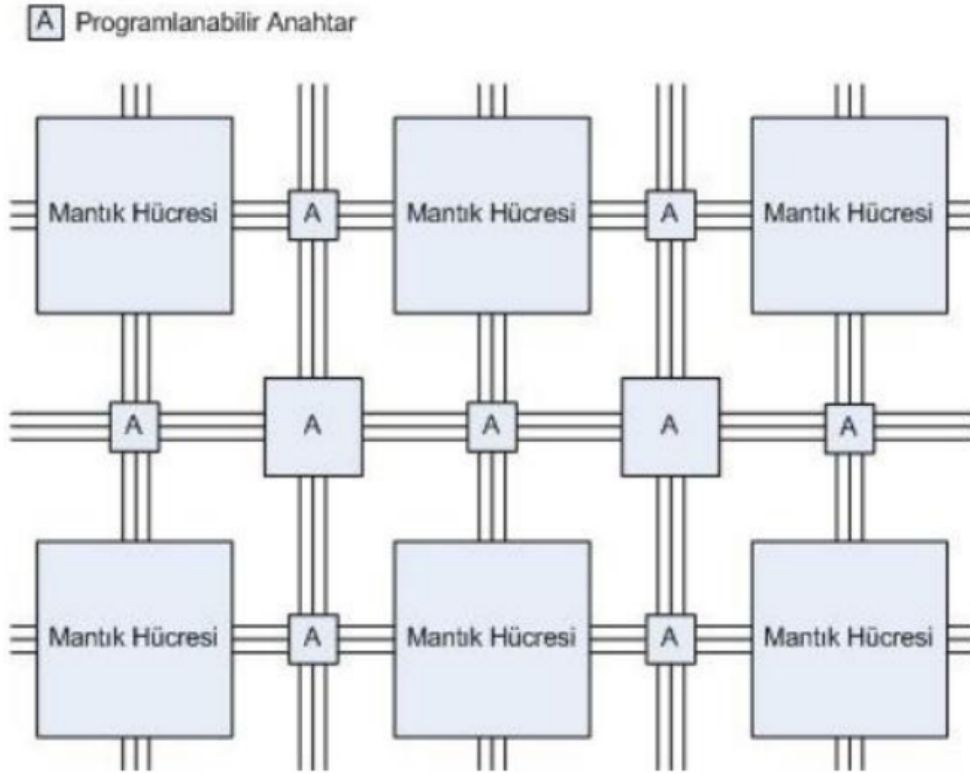
boyutlu olarak dizilmesiyle oluşturulur. Mantık hücreleri basit bir fonksiyonu gerçeklemek üzere yapılandırılabilirdiği gibi programlanabilir anahtarlar ile mantık hücreleri arasında bağlantılar kurulabilir. Bu şekilde mantık hücreleri ve anahtarların programlanmasıyla sayısal donanımlar gerçekleştirilir. Donanım tanımlama dilleri kullanılarak devrenin tasarımı yapıldıktan ve sentezlenmesinin ardından istenilen lojik hücre ve anahtar yapılandırılmasının yer aldığı veri dizisi kablo yardımıyla FPGA'ya gönderilerek devre gerçekleştirilmiş olur [8].

Mantık hücreleri Şekil 2.3'de görüldüğü gibi programlanabilir kombinezonel devre ve bir adet D tipi flip-flop içerir.



Şekil 2.3 : Mantık Hücresi Yapısı [8].

Programlanabilir ara bağlantılardan ve içyapısı Şekil 2.3’de gösterilen mantık hücrelerinden oluşan FPGA’nın genel yapısı Şekil 2.4’de gösterilmektedir



Şekil 2.4 : FPGA'nın İç Yapısı [8].

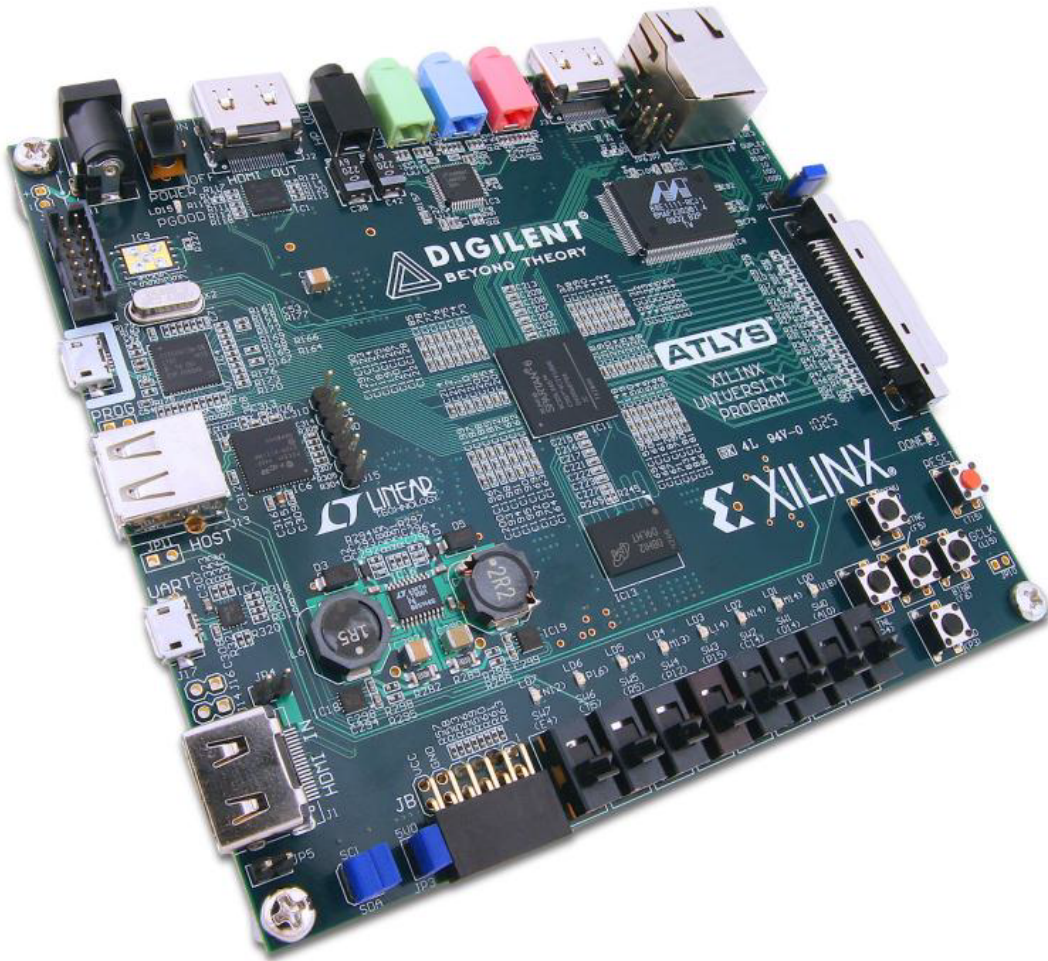
Mantık hücrelerindeki LUT (Look-up Table) yapılandırılabilen kombinezonal devreleri gerçeklemek için kullanılmaktadır. LUT'lar aslında bir mantık işlemini yerine getiren küçük belleklerdir. N girişli bir LUT 2^N boyutlu bellek elemanına karşılık düşmektedir. Binlerce LUT elemanı yan yana getirilerek daha karmaşık kombinezonal fonksiyonlar gerçekleştirilmesine imkân tanır [6].

FPGA'nın paralel işlem yapabilme kapasitesine sahip olması sayesinde çok daha hızlı sistemlerin tasarlanmasına olanak sağlanmaktadır. Bununla birlikte, mikroişlemciler de mantık devrelerinden oluştuğu için FPGA üzerinde kullanılabilirler. Dolayısıyla tek bir tümleşik devre içerisinde kontrol birimi olarak hem işlemci hem de kullanıcıya özgü donanımsal fonksiyonları gerçekleyen fonksiyonlar tanımlamak mümkündür. Tüm sistemin aynı yerde yer alması bağlantılar arası gecikmeler azalacağından FPGA üzerinde daha da hızlı sistemlerin tasarlanmasına da olanak sağlamaktadır. Bütün bu özelliklerin tasarım sırasında

büyük esneklik sağlaması ve ayrıca FPGA'nın paralel işlem yapabilme kapasitesine sahip olduğundan ötürü bu tezin FPGA üzerinde tasarlanması tercih edilmiştir.

2.4. Xilinx Spartan-6 LX45 FPGA

Çalışmada kullanılan kart üzerinde Xilinx firmasının Spartan-6 LX45 FPGA'yi bulunmaktadır. Bu FPGA, her biri 4 adet 6 girişli LUT ve 8 adet flip flop içeren 6822 dilim, 58 DSP dilimi içermektedir. Ayrıca 500 MHz'e kadar saat hızı sunmaktadır [10].



Şekil 2.5 : Kullanılan FPGA Kartı [10].

2.5. Discrete Cosine Transform ve Inverse Discrete Cosine Transform

DCT bir işareti frekans bileşenlerine ayrılmasıdır[15]. IDCT bu işlemin ters dönüşümüdür.

Birçok görüntü sıkıştırma standardında DCT işlemi görüntü 8x8 piksellik parçalara ayrılarak uygulanır. DCT işlemi görüntü 8x8 bloklardan daha büyük parçalara ayrılıp uygulandığında sıkıştırmada kayda değer bir iyileştirme sağlanmamaktadır [5]. Bu sebeple çalışmada görüntü 8x8 piksellik parçalara ayrılarak DCT ve IDCT işlemleri yapılmıştır.

DCT işlemini gerçeklemek için denklem 2.1, IDCT işlemini gerçeklemek için denklem 2.2 kullanılmıştır.

$$Y=C*G*CT \quad (2.1)$$

$$G'=CT*X*C \quad (2.2)$$

G 8x8 piksellik görüntü parçasını, C 8x8 DCT matrisini, CT DCT matrisinin transpozunu, Y DCT işlemi sonucu oluşan 8x8 katsayı değerlerini, G' IDCT işlemi sonucunda oluşan 8x8 piksellik görüntü parçasını belirtmektedir.

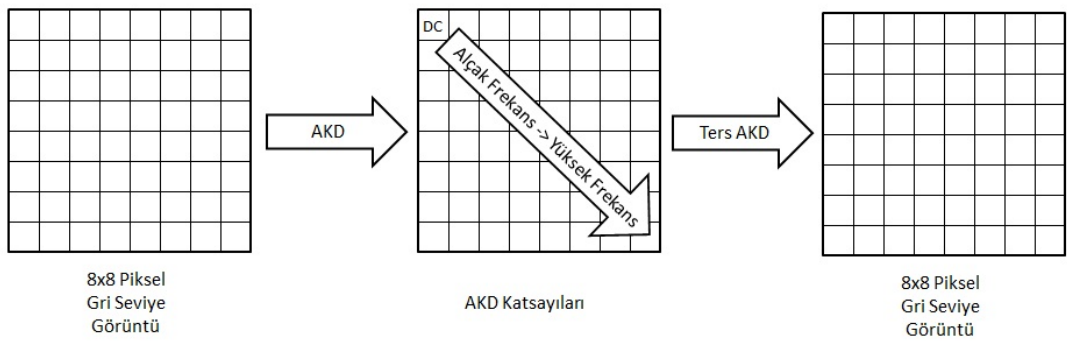
DCT ve IDCT işlemlerinde kullanılacak olan DCT matrisi donanımda gerçekleştirilebilmek için 16 bit kaydırılarak kullanılmıştır.

$$C = \begin{bmatrix} 23170 & 23170 & 23170 & 23170 & 23170 & 23170 & 23170 & 23170 \\ 32138 & 27246 & 18205 & 6393 & -6393 & -18205 & -27246 & -32138 \\ 30274 & 12540 & -12540 & -30274 & -30274 & -12540 & 12540 & 30274 \\ 27246 & -6393 & -32138 & -18205 & 18205 & 32138 & 6393 & -27246 \\ 23170 & -23170 & -23170 & 23170 & 23170 & -23170 & -23170 & 23170 \\ 18205 & -32138 & 6393 & 27246 & -27246 & -6393 & 32138 & -18205 \\ 12540 & -30274 & 30274 & -12540 & -12540 & 30274 & -30274 & 12540 \\ 6393 & -18205 & 27246 & -32138 & 32138 & -27246 & 18205 & -6393 \end{bmatrix}$$

Şekil 2.6: Kullanılan DCT Matrisi [5].

$$C^t = \begin{bmatrix} 23170 & 32138 & 30274 & 27246 & 23170 & 18205 & 12540 & 6393 \\ 23170 & 27246 & 12540 & -6393 & -23170 & -32138 & -30274 & -18205 \\ 23170 & 18205 & -12540 & -32138 & -23170 & 6393 & 30274 & 27246 \\ 23170 & 6393 & -30274 & -18205 & 23170 & 27246 & -12540 & -32138 \\ 23170 & -6393 & -30274 & 18205 & 23170 & -27246 & -12540 & 32138 \\ 23170 & -18205 & -12540 & 32138 & -23170 & -6393 & 30274 & -27246 \\ 23170 & -27246 & 12540 & 6393 & -23170 & 32138 & -30274 & 18205 \\ 23170 & -32138 & 30274 & -27246 & 23170 & -18205 & 12540 & -6393 \end{bmatrix}$$

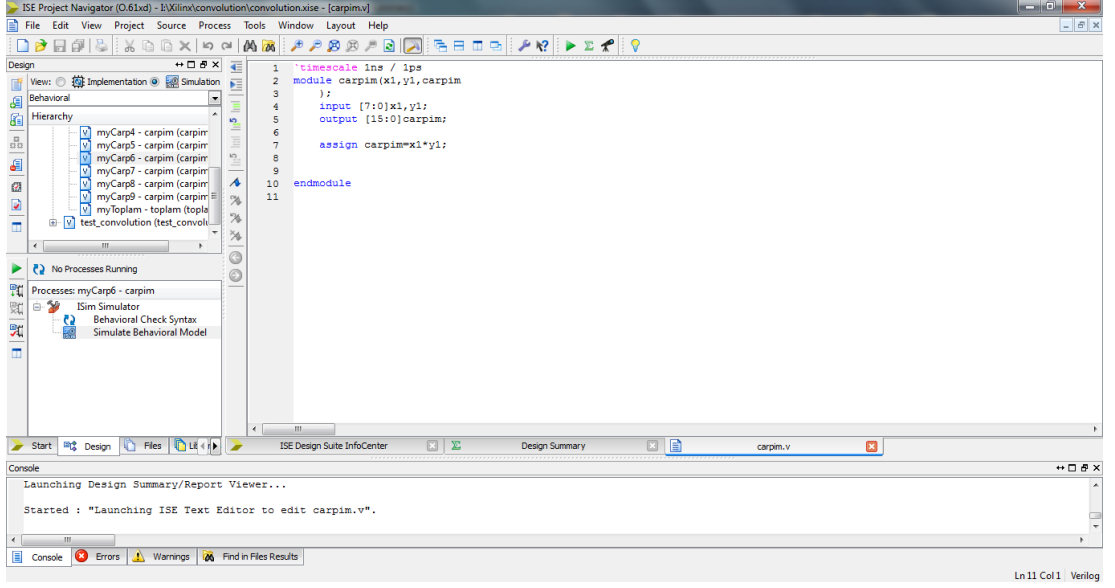
Şekil 2.7: Kullanılan IDCT Matrisinin Transpozu [5].



Şekil 2.8: Ayrık Kosinüs Dönüşümü ve Ters Ayrık Kosinüs Dönüşümünün Uygulanışı.

2.6. Xilinx ISE Programı

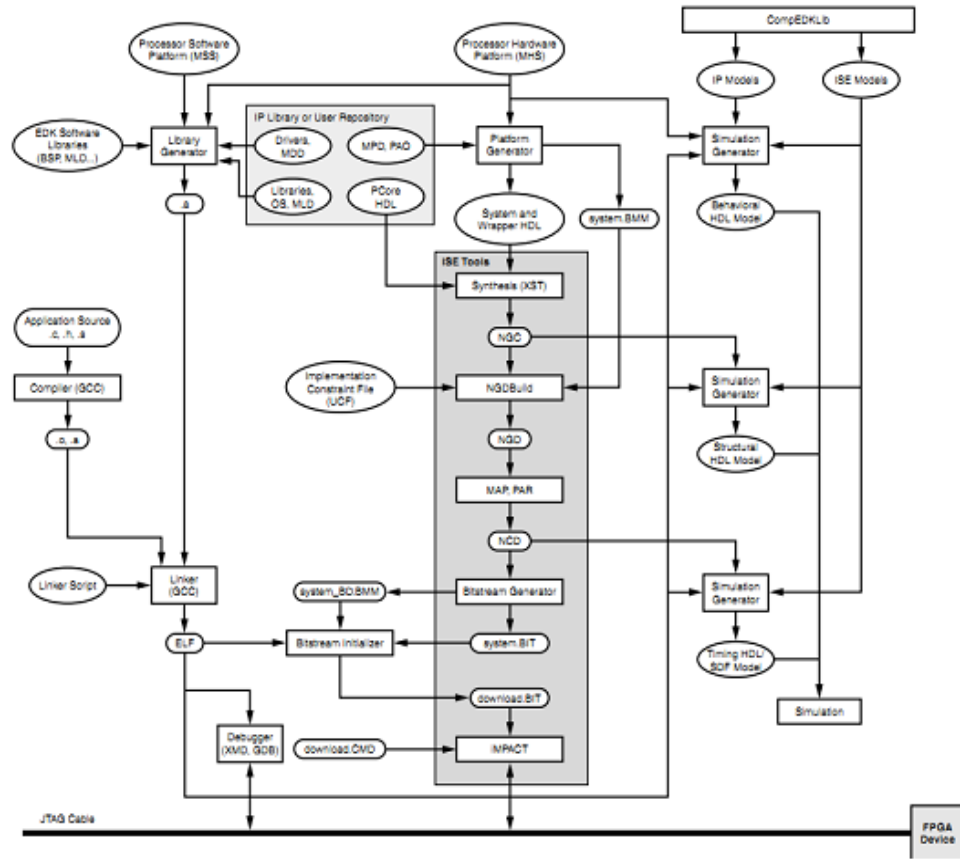
Tümleşik Yazılım Ortamı (Integrated Software Environment, ISE) FPGA'ları programlamak için kullanılmak üzere Xilinx firmasının geliştirdiği bir ara yüz yazılımıdır. ISE ortamı, donanım tanımlama dilleri ya da şematik çizimler sayesinde FPGA'da çalıştırmak üzere sistemler tasarlamaya olanak sağlar. ISE ortamında Verilog, VHDL gibi donanım tanımlama dilleriyle oluşturulan tasarım sentezleme ve gerçekleştirme aşamalarından geçirek kablo aracılığıyla FPGA içine yerleştirilebilir. Ayrıca, donanım üzerinde çalıştırılmadan hata ayıklamak ve hatanın nereden kaynaklandığını görmek üzere ISE ortamında test yapmaya da olanak sağlamaktadır. Ayrıca ISE kullanıcıya tasarladığı sistemlerin ne kadar hat gecikmesine sahip olduğunu, donanım üzerinde ne kadar yer kapladığını ve yapılan tasarımın FPGA'nın hangi bölgesine yerleştirileceğine kadar detaylı bilgiler sunmaktadır.



Şekil 2.9: Xilinx ISE Programının Görünümü.

2.7. Xilinx EDK Programı

EDK (Embedded Development Kit) ortamı Xilinx firmasının ürettiği FPGA'lar üzerinde mikroişlemci tabanlı sayısal sistemler geliştirmek üzere kullanıma sunulmaktadır. EDK çevre birimlerinin ve FPGA donanımlarının bağlanması, sistemin adreslenmesi, haberleşme protokollerinin yazılması gibi işlerle uğraşmak yerine sadece donanım ve yazılım tasarımına odaklanmayı sağlar [11]. Şekil 2.10'da FPGA içerisindeki mikroişlemciyi kullanarak tasarlanan bir sistemin tasarım akış diyagramı görülmektedir. EDK ortamı bu geliştirme aşamalarının hepsini tek bir ara yüz programı ile kullanıcıya sunarak çok zahmetli ve karmaşık sistemleri daha kolay tasarlanabilir hale getirir ve proje süresini önemli ölçüde kısaltır.



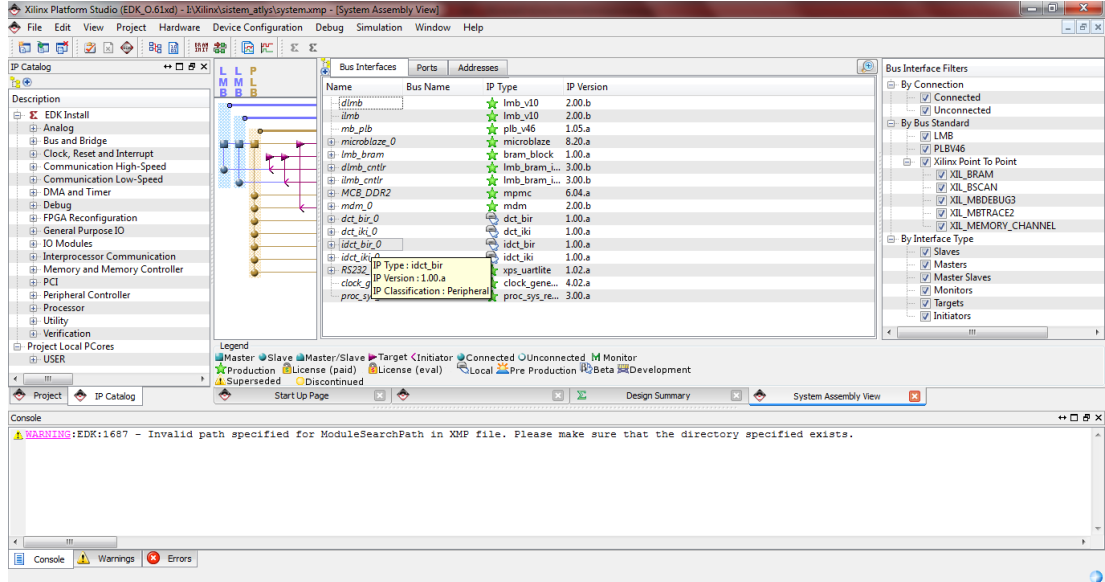
Şekil 2.10: EDK sistem geliştirme araçları [11].

EDK geliştirme ortamında “soft core” (Microblaze) veya “hard core” (PowerPC) gibi mikroişlemci temelli donanım projelerine FPGA kartı üzerinde bulunan çevre birimleri ve giriş-çıkış birimleri eklenebileceği gibi Xilinx tarafından geliştirilmiş donanımlar ve kullanıcının ISE aracılığıyla oluşturduğu kendi donanımları eklenebilmektedir [11].

EDK ortamı Temel Sistem Oluşturucu (Base System Builder, BSB) ile tasarımcıya kendi tasarım sisteminin tabanını oluşturma imkânı sağlar. BSB yardımıyla tasarımcı kolaylıkla FPGA kartı üzerindeki istediği donanımların hazır İnternet Protokollerini (İnternet Protocol, IP) seçerek tasarımına ekleyebilmektedir. Tasarımcı kendi donanımları ve eklediği bu IP’leri EDK’nın sunmuş olduğu veri yolları ile kolayca bağlayarak sistemi oluşturmuş olur.

EDK donanım projesinin yapılandırılmasında XPS (Xilinx Platform Studio) programı kullanılmaktadır. XPS’de Microblaze mikroişlemcisine bağlanan çevre

birimleri Microblaze tarafından adreslenmektedir. Sistemin adres haritası üretildikten sonra XPS ortamında ya da ISE ortamında tasarlanan projeye sentezleme ve gerçekleştirme aşamaları uygulanmaktadır. Son olarak bu aşamadan sonra donanım tasarımı bitirilerek bu donanımı kontrol etmek için kullanılan MicroBlaze ya da PowerPC gibi mikroişlemcilerin yazılımının tasarlanması aşamasına geçilmektedir.



Şekil 2.11: Xilinx EDK Programının Görünümü.

2.8. Xilinx SDK Programı

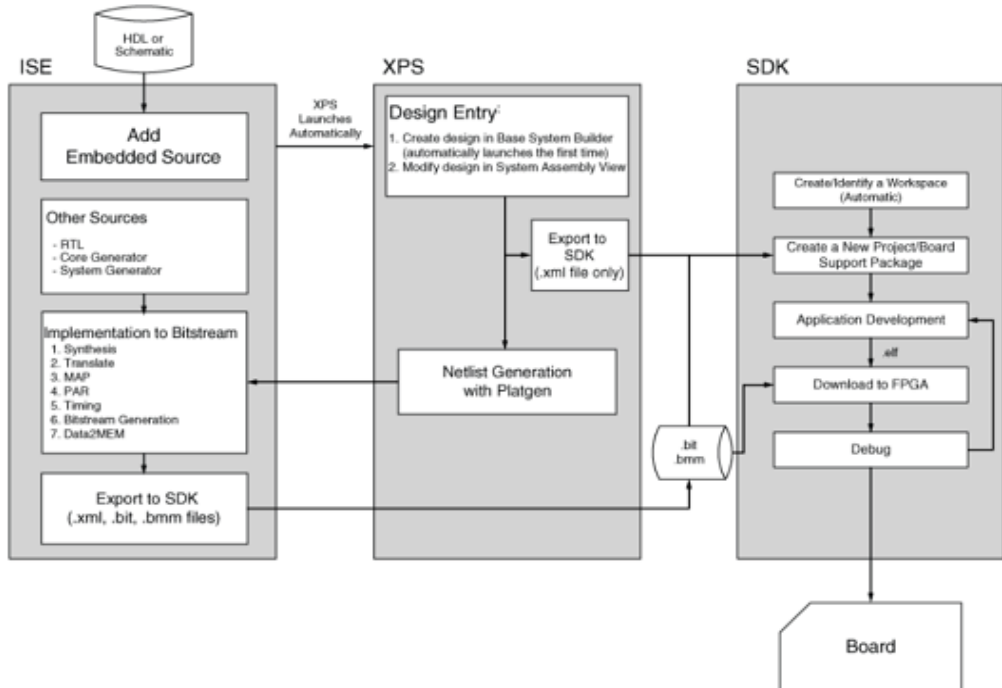
Yazılım Geliştirme Kiti (Software Development Kit, SDK) Xilinx firması tarafından EDK ortamında tasarlanan mikroişlemci merkezli sayısal sistem tasarımlarının yazılım tasarımını gerçekleştirmek için geliştirilen ara yüz ortamıdır. Xilinx'in tasarım ortamlarının eski sürümlerinde SDK, XPS geliştirme ortamı içerisinde yer almaktaydı. ISE13 sürümünden sonra Xilinx firması SDK'yı XPS ortamından ayırarak XPS'yi sadece donanım tasarlama ortamına dönüştürmüştür. SDK ise sadece tasarlanan donanımlara yazılım tasarımı yapmak amacıyla kullanılmaktadır. EDK ortamında tasarlanan sisteme ait kullanıcı donanımları ve çevre birimlerinin kütüphaneleri üretilerek yazılım tasarımına ilk adımın atılması sağlamaktadır. Aynı zamanda, SDK tarafından üretilen kütüphanelerin söz konusu yazılım projesine eklenmesiyle kullanıcıya mikroişlemciyi kolayca kontrol etme olanağı sağlanmaktadır.

- Zengin özellikli C/C++ kod editörü ve derleme ortamı

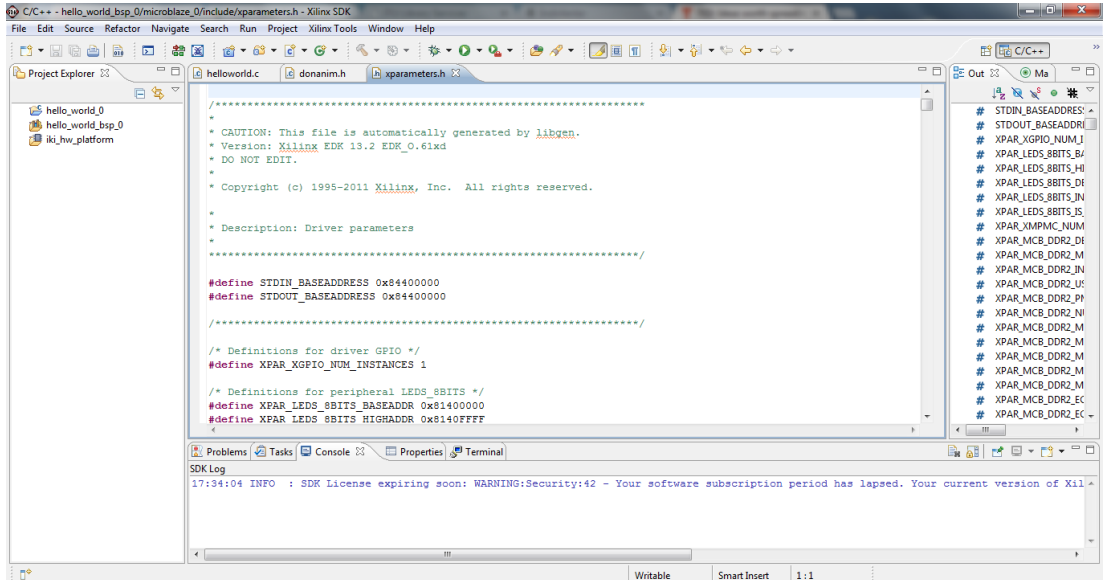
- Proje yönetimi
- Tasarım yapılandırması uygulaması ve otomatik Makefile üretimi
- Hata navigasyonu
- Kaynak düzeyinde hata ayıklama ve gömülü hedeflerin görünüşü için iyi tümleştirilmiş ortam
- Kaynak kodu sürümü kontrolü

SDK tarafından kullanıcılarına sunulmuş başlıca özelliklerdir [12].

ISE, EDK ve SDK ortamları kullanılarak sıfırdan bir sistem tasarımının akışı Şekil 2.12’de gösterilmektedir. Tasarım akışından bahsedilecek olursa öncelikle ISE ortamında donanım tanımlama dilleri ya da şematik çizimlerle tasarlanan donanımlar EDK ortamında kullanıcı donanımı olarak tanımlanır. EDK ortamında kullanıcı donanımına IP verilerek mikroişlemci merkezli sayısal sistem tasarımına eklenmek istenen diğer hazır IP’lerle birlikte eklenmektedir. EDK ortamında donanım yapısı tamamlanan sistem SDK ortamına gönderilerek bu aşamada otomatik olarak kütüphaneleri üretildikten sonra yazılım tasarımı yapılmaktadır. Son olarak, donanım ve bu donanımları kontrol etmek için yapılan yazılım da tamamlandıktan sonra SDK aracılığıyla, donanım bilgilerini içeren “bit” uzantılı donanım dosyası ve “elf” uzantılı yazılım dosyası birleştirilerek FPGA’ya gönderilir.



Şekil 2.12: Sistem tasarımı akışı [13].



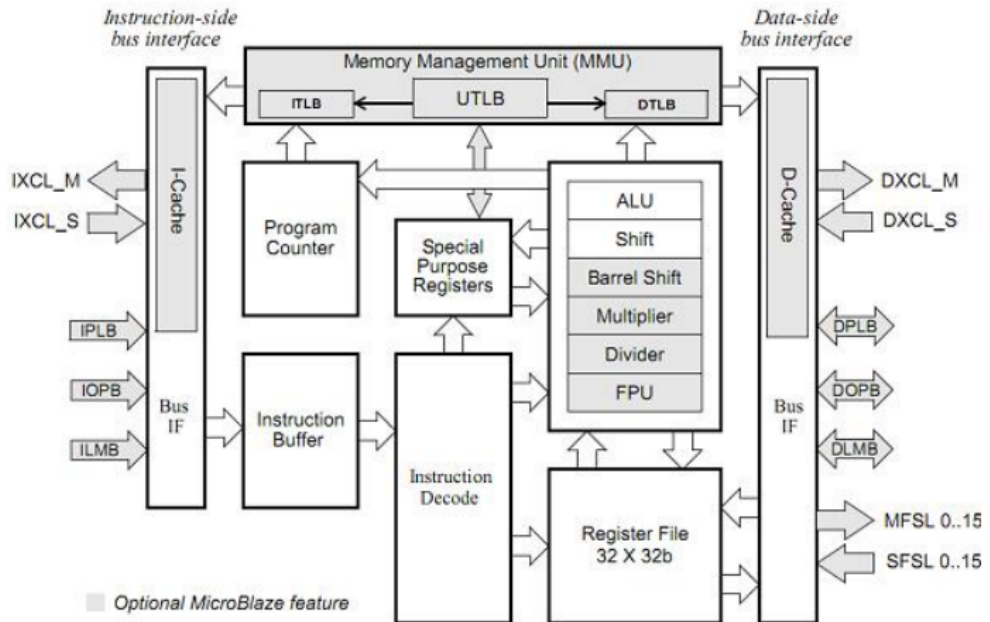
Şekil 2.13: Xilinx SDK Programının Görünümü.

2.9. Microblaze İşlemcisi

Mimari yapısı Şekil 2.14'te gösterilen MicroBlaze mikroişlemcisi, FPGA üzerinde yazılım ile kontrol edilebilir gömülü sistemler tasarlamaya olanak sağlamak üzere FPGA bloklarının uygun şekilde programlanması ile oluşturulur. MicroBlaze, tek FPGA üzerinde kullanılacak çevre birimleri, hafıza ve ara yüz özelliklerinin seçiminde esneklik sağlayarak kullanıcının isteğine tam olarak cevap veren gömülü sistemler tasarlamaya olanak sağlar.

MicroBlaze 32-bit İndirgenmiş Komut Takımı Bilgisayarı (Reduced Instruction Set Computing, RISC) Harvard bellek mimarisine sahiptir. Program ve veri erişimi ayrı bellek alanlarından sağlanır. Her bir adres alanı 32 bit ile adreslenir [14].

32 bitlik 32 adet genel amaçlı kaydedicileri ve 32 bit adres yolu gibi özellikleri sabit iken, iş hattı (pipeline) derinliği, veri yolu sayısı ve türleri, kayan noktalı sayı birimi (Floating Point Unit, FPU) ve bellek idare birimi (Memory Management Unit, MMU) gibi özellikleri ile FPGA için optimize edilmiş bir mikroişlemcidir [14].



Şekil 2.14: MicroBlaze mimarisi [14].

3. FPGA ÜZERİNDE SİSTEM TASARIMI

3.1. Kayıpsız Tasarım

Proje ilk olarak görüntü verilerinde kayıp olmayacak şekilde tasarlanmıştır. DCT katsayı matrisindeki 16 bit kaydırma işlemi başlangıçta yapılmış, görüntü sırasıyla DCT 1, DCT 2, IDCT 1, IDCT 2 donanımlarından geçmiştir. IDCT 2 donanımında son işlem olarak 16 bit kaydırma işleminden gelen fazla bitler kaydırma yöntemiyle silinmiştir.

Bu tasarım MicroBlaze ile Spartan 6 LX45 FPGA'ine sığmamıştır. Bu nedenle belirli bir oranda kayıp kabul edilerek kayıplı tasarım gerçekleştirilmiştir.

3.2. Kayıplı Tasarım

DCT işleminin yapılabilmesi için piksel değerlerinin sıfır etrafında salınması gerekir. Bu yüzden DCT işlemi uygulanmadan önce işlemci aracılığıyla tüm piksel değerlerinden yüz yirmi sekiz çıkarılır. Böylece işaretli 8 bitlik verilerden oluşan 8x8 boyutunda bir matris elde edilir. Bu matrisin tüm satırları işlemci aracılığıyla ilk satırından başlanarak sırayla DCT 1 donanımına verilir.

DCT işleminin ilk kısmı olan $X * C^T$ matris çarpım işlemi DCT 1 donanımında gerçekleşir. Tüm matris DCT 1 donanımında işleme tabi tutulduktan sonra işaretli 27 bitlik verilerden oluşan 8x8 boyutunda bir matris elde edilir.

Elde edilen bu matrisin ilk sütunundan başlanarak tüm sütunları işlemci aracılığıyla sırayla DCT 2 donanımına verilir. DCT işleminin ikinci kısmı olan $C * (X * C^T)$ matris çarpım işlemi DCT 2 donanımında gerçekleşir. DCT matrisindeki 16 bitlik kaydırma işleminden oluşan bitler çıkarılarak çıkışta işaretli 14 bitlik tam sayı değerli verilerden oluşan 8x8 boyutunda bir matris elde edilir Bu matris DCT katsayılarını içermektedir.

DCT katsayıları kullanılarak yazılımla belirlenmiş damgalama yöntemi uygulanır. Damgalama işlemi işlemci aracılığıyla yapılır. Böylece donanımlar değiştirilmeden

işlemci yazılımı değiştirilerek damgalama yöntemi kolaylıkla değiştirilebilir. Bu da sisteme esneklik sağlar.

Damgalanmış görüntünün tekrar uzamsal tanım kümesine dönüştürülmesi için IDCT işlemi uygulanması gerekir. Bu sebeple DCT katsayılarından oluşan matrisin ilk satırından başlanarak tüm satırları işlemci aracılığıyla sırayla IDCT 1 donanımına verilir. IDCT işleminin ilk kısmı olan $X * C$ matris çarpım işlemi bu donanımda gerçekleşir. Çıkışta işaretli 33 bitlik verilerden oluşan 8×8 boyutunda bir matris elde edilir.

Elde edilen yeni matrisin ilk sütunundan başlayarak tüm sütunları işlemci aracılığıyla sırayla IDCT 2 donanımına verilir. IDCT işleminin ikinci kısmı olan $C^T * (X * C)$ matris çarpım işlemi bu donanımda gerçekleşir. Çıkışta tam sayı değerleri işaretli 20 bitlik verilerden oluşan 8×8 boyutunda bir matris elde edilir.



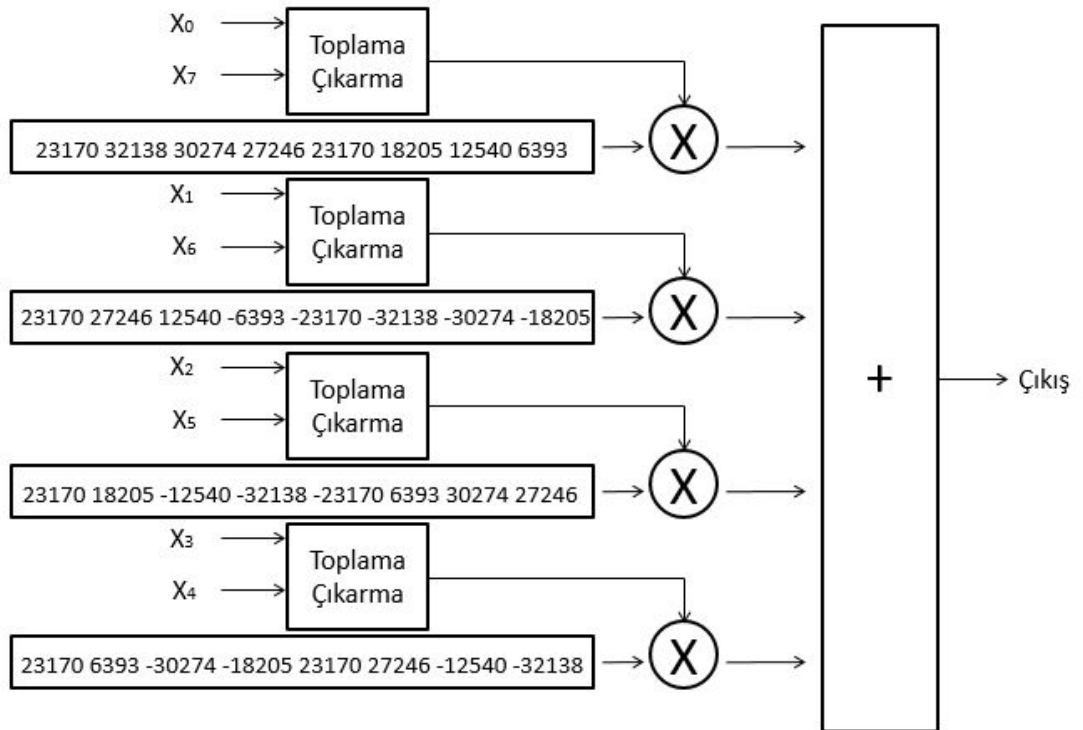
Şekil 3.1 : Sistemin genel yapısı.

Bu matris 8×8 piksellik görüntünün 128 çıkarılmış halini içermektedir. Oluşan matrise işlemci aracılığıyla yüz yirmi sekiz eklenerek görüntü başlangıçta durumuna getirilir. Şekil 3.1’de sistemin genel yapısı görülmektedir.

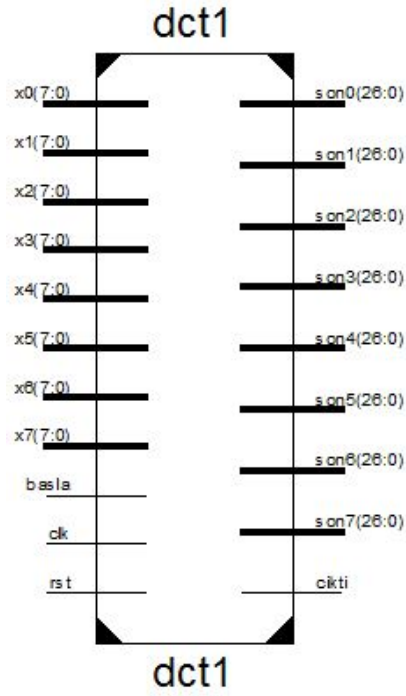
4. DONANIM TASARIMI

4.1.DCT 1 Donanımı

DCT işleminin ilk kısmı olan $X * C^T$ matris çarpım işlemi DCT 1 donanımında gerçekleşir. 8×8 boyutundaki X matrisinin her bir satırı bu donanıma girerek C^T matrisi ile çarpılır.

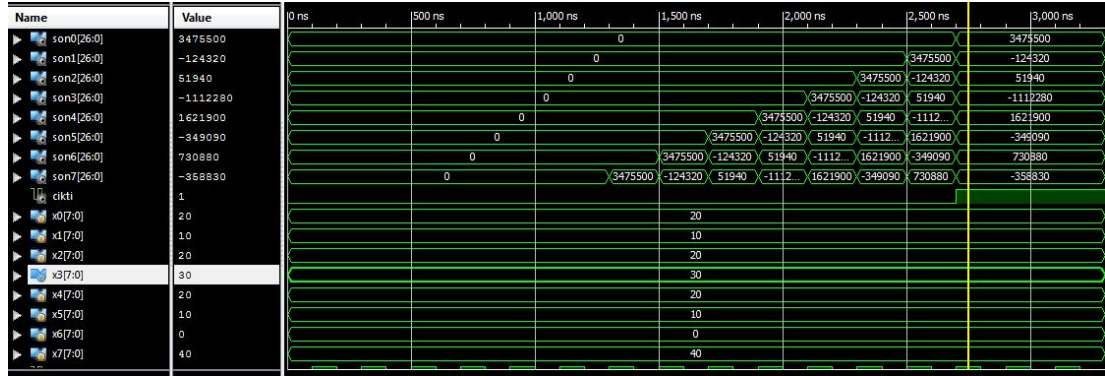


Şekil 4.1 : DCT 1 donanımının iç yapısı.



Şekil 4.2 : DCT 1 donanımının dış yapısı.

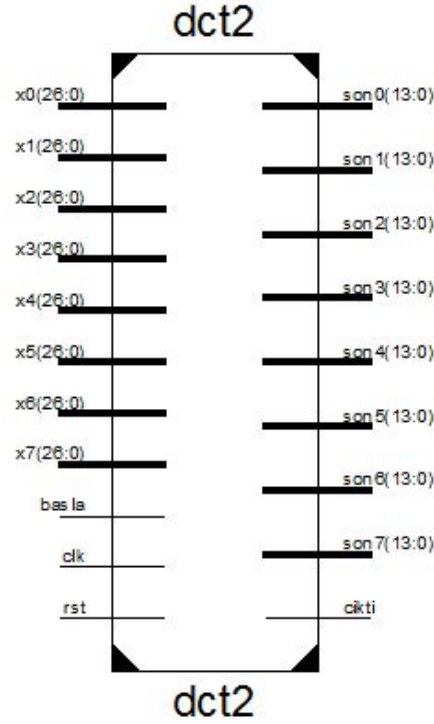
DCT 1 donanımının benzetimi yapıldığında şekilde görüldüğü gibi istenilen sonuçlar alınmıştır.



Şekil 4.3 : DCT 1 donanımının benzetimi.

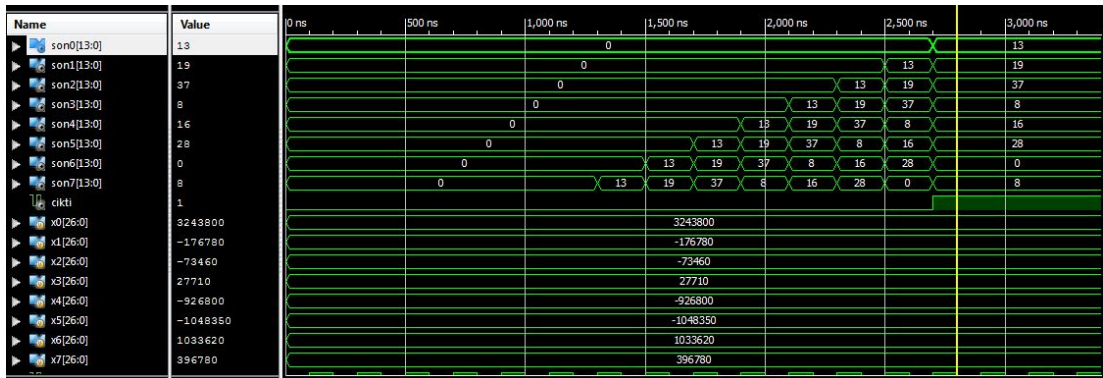
4.2. DCT 2 Donanımı

DCT işleminin ikinci kısmı olan $C*(X*C^T)$ matris çarpım işlemi bu donanımda gerçekleşir. DCT 1 donanımı sonrası oluşan matrisin ilk sütunundan başlayarak tüm sütunları sırayla DCT 2 donanımına verilir. DCT matrisindeki 16 bitlik kaydırma işleminden oluşan bitler çıkarılarak 14 bitlik işaretli tam sayı değerleri elde edilir. DCT 1 ve DCT 2 donanımlarının arka arkaya uygulanması ile DCT işleminin tamamı gerçekleşir. DCT 2 donanımının içyapısı DCT 1 ile aynı yapıdadır.



Şekil 4.4 : DCT 2 donanımının dış yapısı.

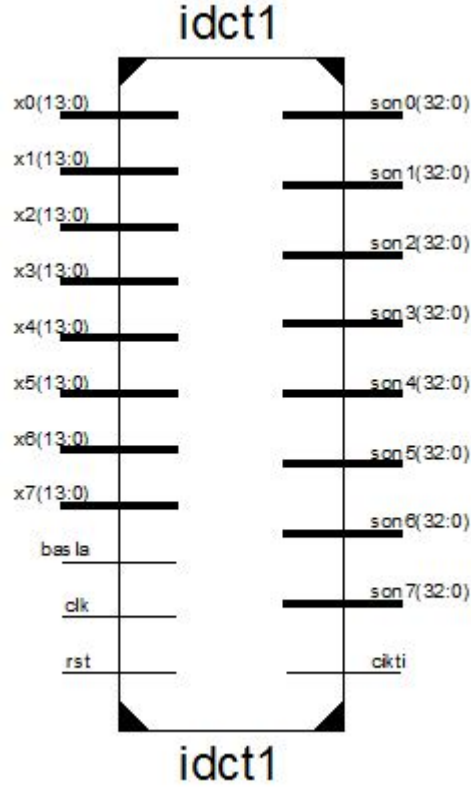
DCT 2 donanımının benzetimi yapıldığında şekilde görüldüğü gibi istenilen sonuçlar alınmıştır.



Şekil 4.5 : DCT 2 donanımının benzetimi.

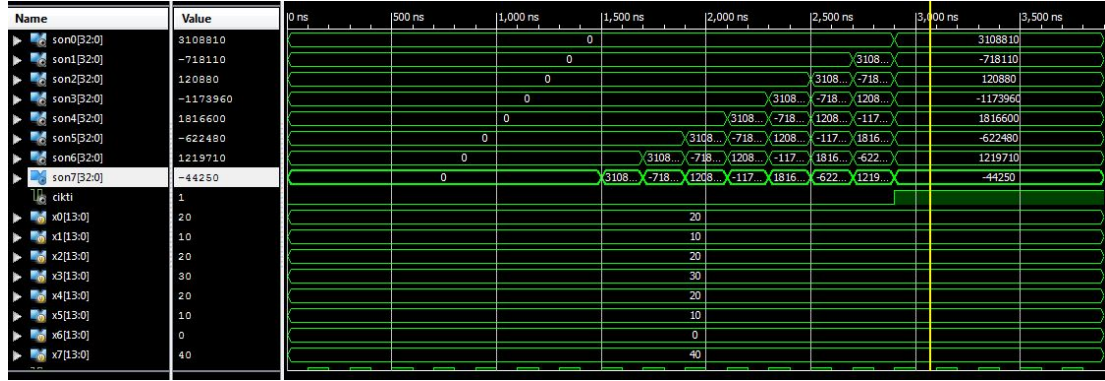
4.3. IDCT 1 Donanımı

Damgalanmış görüntünün tekrar uzamsal tanım kümesine dönüştürülmesi için IDCT işlemi uygulanması gerekir. Bu yüzden DCT katsayılarından oluşan matrisin ilk satırından başlanarak tüm satırları sırayla IDCT 1 donanımına verilir. IDCT işleminin ilk kısmı olan $X \cdot C$ matris çarpım işlemi bu donanımda gerçekleşir. Çıkışta işaretli 33 bitlik verilerden oluşan 8×8 boyutunda bir matris elde edilir.



Şekil 4.6 : IDCT 1 donanımının dış yapısı.

IDCT 1 donanımının benzetimi yapıldığında şekilde görüldüğü gibi istenilen sonuçlar alınmıştır.

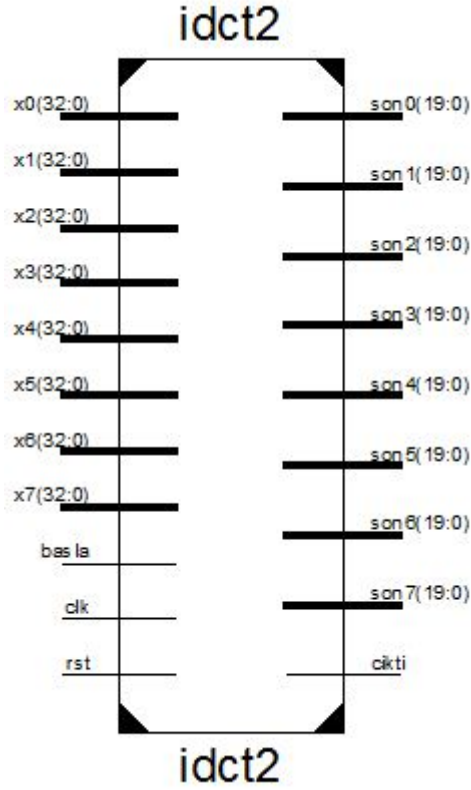


Şekil 4.7 : IDCT 1 donanımının benzetimi.

4.4. IDCT 2 Donanımı

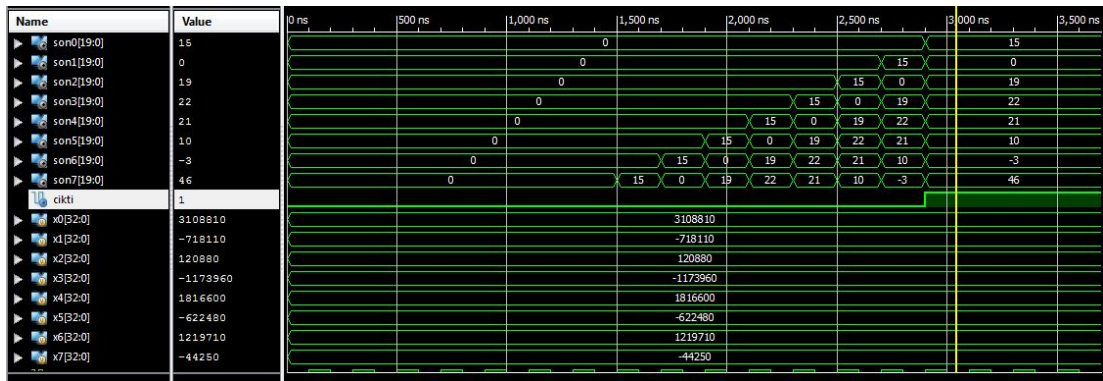
Elde edilen yeni matrisin ilk sütunundan başlayarak tüm sütunları sırayla IDCT 2 donanımına verilir. IDCT işleminin ikinci kısmı olan $C^T * (X * C)$ matris çarpım işlemi bu donanımda gerçekleşir. Çıkışta tam sayı değerleri işaretli 20 bitlik

verilerden oluşan 8x8 boyutunda bir matris elde edilir. IDCT 1 ve IDCT 2 donanımlarının arka arkaya uygulanması ile IDCT işleminin tamamı gerçekleştirilir.



Şekil 4.8 : IDCT 2 donanımının dış yapısı.

IDCT 2 donanımının benzetimi yapıldığında şekilde görüldüğü gibi istenilen sonuçlar alınmıştır.



Şekil 4.9 : IDCT 2 donanımının benzetimi.

4.5. Donanımların Zaman Analizi

Donanımlar Xilinx Spartan 6 FPGA üzerinde gerçekleştirilmiştir. Donanımların kapladığı alanı küçük tutmak için tasarım ardışık düzen yapısında tasarlanmıştır. Bu nedenle her bir donanımın bir satırı veya sütunu işlemesi için 13 saat çevrimi gereklidir. Donanımların maksimum çalışma frekansı Şekil 4.10'da belirtilmiştir. Donanımlar DCT 2 donanımının Şekil 4.10'da belirtilen maksimum çalışma frekansında çalıştırıldığında her bir 8x8 piksellik parçaya DCT ve IDCT işlemlerinin uygulanması toplam 4.161 mikro saniye sürmektedir.

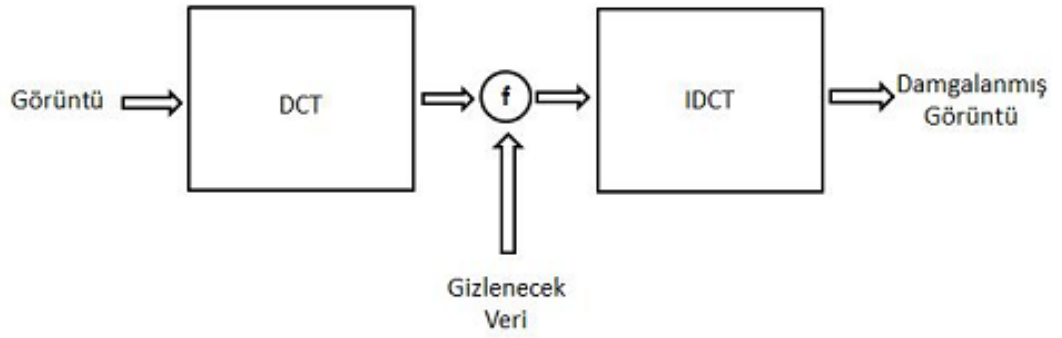
	Maksimum Çalışma Frekansı(MHz)
DCT 1	193.259
DCT 2	99.974
IDCT 1	181.232
IDCT 2	136.850

Şekil 4.10 : Donanımların maksimum çalışma frekansı.

5.Yazılım

Donanımları kontrol etmek, donanımlar arası veri akışını sağlamak ve damgalama işlemini yapmak için işlemci kullanılmıştır. Xilinx SDK programı aracılığıyla C programlama dilinde yazılım gerçekleştirilmiştir.

Damgalama işleminin yazılım ile gerçekleştirilmesi, donanımları değiştirmeden damgalama yöntemini değiştirme esnekliği sağlamıştır. DCT 2 donanımından gelen DCT katsayıları ile istenen damgalama yöntemi yazılımla yapılabilmektedir.



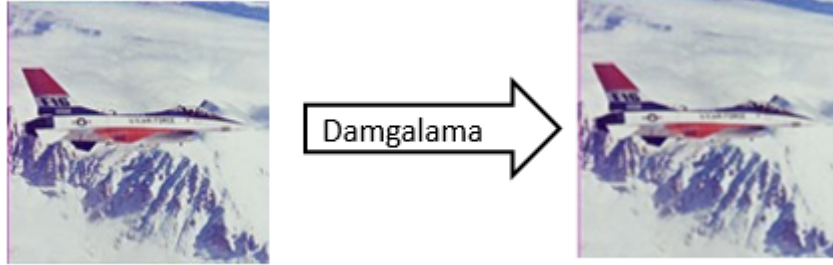
Şekil 5.1 : DCT ile damgalama işlemi.

6.Damgalama Yöntemi

Çalışmadaki amaç istenilen damgalama yönteminin yazılım ile değiştirilip uygulanabilmesidir. Bu nedenle çalışmada özel bir damgalama yöntemi seçilerek gerçekleştirme yoluna gidilmemiştir. Bunun yerine seçilen DCT katsayılarının en anlamsız üçüncü bitleri bir yapılarak damga çıkarımının başarı oranı test edilmiştir. Damganın JPEG sıkıştırma işleminden etkilenmemesi ve görüntüde büyük değişikliklerin olmaması için orta frekansları temsil eden DCT katsayıları seçilmiştir. DCT katsayı matrisi zigzag düzenine göre sıralanarak 3. Satır 3. sütündeki katsayıdan 4. satır 4. sütündeki katsayıya kadar olan katsayılara damgalama işlemi yapılmıştır. Şekil 6.3'de damgalama yapılan katsayı ve damganın çıkarılma başarısının yüzdesi verilmiştir.



Şekil 6.1 : Lena görüntüsünün damgalamadan önceki ve sonraki görünümü.



Şekil 6.2 : Airplane görüntüsünün damgalamadan önceki ve sonraki görünümü.

Katsayı	Lena	Airplane
3,3	0.8893	0.8854
2,4	0.8581	0.9010
1,5	0.9036	0.9271
1,6	0.8724	0.9010
2,5	0.8659	0.8984
3,4	0.9245	0.9479
4,3	0.9323	0.9284
5,2	0.9141	0.9362
6,1	0.9167	0.9297
7,1	0.9193	0.8906
6,2	0.9023	0.9076
5,3	0.9010	0.9023
4,4	0.8906	0.8945

Şekil 6.3 : Damga çıkarım başarı yüzdeleri.

7.Sonuçlar

Çalışmanın başlangıcında çeşitli görüntü damgalama yöntemleri incelenmiştir. Avantajları dikkate alınarak DCT tanım kümesindeki damgalama işlemleri için yazılım/donanım ortaklı sistem tasarımı FPGA üzerinde gerçekleştirilmiştir.

Tasarlanan sistem gerçekleştirildiği Spartan 6 LX45 FPGA üzerinde 99 MHz frekanslı saat işareti ile çalışabilmektedir. Bu saat işareti dikkate alındığında, sistem 8x8 piksellik görüntü parçasına DCT işlemi yapıp damgalayıp tekrar IDCT işlemi yaptığında 4.161 mikro saniye sürmektedir.

DCT ve IDCT işlemleri donanımda gerçekleştirilerek hız kazanılmıştır. Damgalama işlemi yazılımla yapılarak damgalama algoritmasının kolaylıkla değiştirilebilmesi sağlanmıştır. Böylece DCT kullanılarak geliştirilen damgalama yöntemlerinin gömülü sistemlerde gerçekleştirilmesinde hız ve esneklik sağlanmıştır. Geliştirilen yeni yöntemlerin gömülü sistemlere uygulanmasında donanımın değiştirilmesine gerek kalmadan yazılımın değiştirilmesi yeterli hale gelmiştir. Bunun sonucu olarak literatürdeki DCT kullanılarak geliştirilen damgalama algoritmaları sistem üzerinde yazılım aracılığıyla gerçekleştirilebilir.

KAYNAKLAR

- [1] **Cox, I.J., Kilian, J., Leighton, F.T., and Shamoon, T.**, 1997, Secure spread spectrum watermarking for multimedia, Image Processing, IEEE Transactions on , vol.6, no.12, pp.1673-1687.
- [2] **Sin-Joo Lee, and Sung-Hwan Jung**, 2001, A survey of watermarking techniques applied to multimedia, Industrial Electronics, IEEE International Symposium on ,vol.1, no., pp.272-277.
- [3] **Bahi, J. M., Couchot, J., and Guyeux, C.**, 2012, Steganography: A class of secure and robust algorithms, The Computer Journal, 55(6), 653.
- [4] **Bender, W., Gruhl, D., Morimoto, N., Lu, A.**, 1996, Techniques for data hiding, IBM Systems Journal, vol.35, no.3.4, pp.313-336.
- [5] **Pillai L.**, Video Compression Using DCT (v1.2), [Alıntı Tarihi: 5 Mayıs 2013], <http://www.cs.york.ac.uk/rts/docs/Xilinx-datasource-2003-q1/appnotes/xapp610.pdf>.
- [6] MicroBlaze Soft Processor Core, [Alıntı Tarihi: 3 Nisan 2013], <http://www.xilinx.com/tools/microblaze.htm>.
- [7] **Ingemar J. C.**, 2008. Digital watermarking and steganography, Morgan Kaufmann Publishers, Amsterdam
- [8] **Chu, Pong P.**, 2008. FPGA Prototyping by VHDL Examples. Wiley-Interscience, New Jersey.
- [9] **Koca, H.**, 2007. “Robot Manipulator Denetimi”, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.
- [10] Atlys™ Spartan-6 FPGA Development Board, [Alıntı Tarihi: 10 Mayıs 2013],<http://www.digilentinc.com>.
- [11] **Xilinx**, 2007. Embedded System Tools Reference Manual.
- [12] **Xilinx**, Software Development Kit Help Contents, [Alıntı Tarihi: 11 Mayıs 2013],http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_2/SDK_Doc/index.html.
- [13] **Xilinx**, 2011. EDK Concepts, Tools and Techniques.
- [14] **Xilinx**, 2007. MicroBlaze Processor Reference Guide.
- [15] **Watson, B.A.**, 1994. Image Compression Using the Discrete Cosine Transform, Mathematica Journal, 4(1), 81-88.

ÖZGEÇMİŞ

Adı Soyadı: Ahmet Turan Erozan

Doğum Yeri ve Tarihi: Sivas, 1990

Lisans Eğitimi: İstanbul Teknik Üniversitesi; 2008-2013