

**T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM
DALI**



**DİJİTAL KAMERA KULLANILARAK ROBOTUN
BİLGİSAYAR SİSTEMİYLE BİRLEŞİK ÇALIŞTIRILMASI**

YÜKSEK LİSANS TEZİ

ERMAN KÖYBAŞI

BALIKESİR, HAZİRAN 2013

**T.C.
BALIKESİR ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM
DALI**



**DİJİTAL KAMERA KULLANILARAK ROBOTUN
BİLGİSAYAR SİSTEMİYLE BİRLEŞİK ÇALIŞTIRILMASI**

YÜKSEK LİSANS TEZİ

ERMAN KÖYBAŞI

BALIKESİR, HAZİRAN 2013

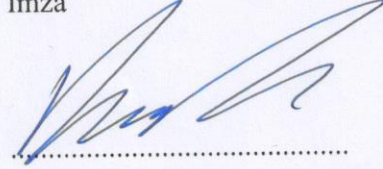
KABUL VE ONAY SAYFASI

Erman KÖYBAŞI tarafından hazırlanan "DİJİTAL KAMERA KULLANILARAK ROBOTUN BİLGİSAYAR SİSTEMİYLE BİRLEŞİK ÇALIŞTIRILMASI" adlı tez çalışmasının savunma sınavı 19.06.2013 tarihinde yapılmış olup aşağıda verilen jüri tarafından oy birliği / oy çokluğu ile Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalı Yüksek Lisans Tezi olarak kabul edilmiştir.

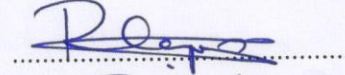
Jüri Üyeleri

İmza

Danışman
Yrd. Doç. Dr. Davut AKDAŞ



Üye
Prof. Dr. Rifat ÇAPAN



Üye
Yrd. Doç. Dr. Murat Erhan BALCI



Jüri üyeleri tarafından kabul edilmiş olan bu tez BAÜ Fen Bilimleri Enstitüsü Yönetim Kurulunca onanmıştır.

Fen Bilimleri Enstitüsü Müdürü

Prof. Dr. Hilmi NAMLI

.....

Bu tez alıřması Balıkesir niversitesi Bilimsel Arařtırma Projeleri Birimi tarafından 2011-46 nolu proje ile desteklenmiřtir.

ÖZET

**DİJİTAL KAMERA KULLANILARAK ROBOTUN BİLGİSAYAR
SİSTEMİYLE BİRLEŞİK ÇALIŞTIRILMASI
YÜKSEK LİSANS TEZİ
ERMAN KÖYBAŞI
BALIKESİR ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI
(TEZ DANIŞMANI: YRD. DOÇ. DR. DAVUT AKDAŞ)**

BALIKESİR, HAZİRAN 2013

Bu tez çalışmasında, Field Programmable Gate Array (FPGA) platformu görüntü işleme için kullanılmıştır. Bu platformun seçilme nedeni genel olarak daha az bilinmesine ve kullanılmasına rağmen paralel işlem yapabilme kapasitesi nedeniyle yüksek verime sahip olabilmesidir. Robotik sistemlerde gerçek zamanlı etkileşim yapılabilmesi için hız son derece önemli bir yere sahiptir. Robotların vereceği tepki süresini düşürebilmek ve gerçek hayat uygulamalarında bir yere sahip olabilir duruma getirebilmek için hızlı işlem yapmak hayati öneme sahiptir.

Görüntü işleme seçilmesinin nedeni ise robotlarda çevreyle ve insanlarla etkileşimi, gerçekçi ve daha kapsamlı kılmanın sensörlerden ziyade görüntü işlemeyle gerçekleştirilebilmesinin daha mümkün olmasındandır. Böylece ileride yapılacak olan insansı robotlarda yön belirleme, nesne tanıma gibi fonksiyonlar da kullanılabilir olacaktır.

ANAHTAR KELİMELELER: FPGA, görüntü işleme, filtreleme, Sobel

ABSTRACT

RUNNING THE ROBOT WITH COMPUTER SYSTEM USING FPGA MSC THESIS

ERMAN KÖYBAŞI

**BALIKESİR UNIVERSITY INSTITUTE OF SCIENCE
ELECTRICAL AND ELECTRONICS ENGINEERING
(SUPERVISOR: ASSIST. PROF. DR. DAVUT AKDAŞ)**

BALIKESİR, JUNE 2013

FPGA platform has been used for image processing in this thesis. The reason of choice for this platform is that it has very high efficiency due to its capacity of parallel computing even though being a less known and used platform. Speed is very essential in robotic systems to be able to make real time interaction. Making high speed processing is crucial to be able to lower reaction time and make robotics usable in real life applications.

The reason to choose image processing is that it makes it more possible to make robot environmental and human interaction realistic and more comprehensive than sensors. By this way it will be possible to use functions as direction finding or object recognition in the humanoid robots that will be built in the future.

KEYWORDS: FPGA, image processing, filtering, Sobel

İÇİNDEKİLER

Sayfa

ÖZET	i
ABSTRACT	ii
İÇİNDEKİLER	iii
ŞEKİL LİSTESİ	v
TABLO LİSTESİ	vii
SEMBOL VE KISALTIMA LİSTESİ	viii
ÖNSÖZ	x
1. GİRİŞ	1
2. TARİHÇE	3
2.1 Field Programmable Gate Array'lerin gelişimi.....	3
2.2 Sayısal Görüntü İşlemenin Gelişimi.....	7
3. ROBOTİKTE GÖRÜNTÜ İŞLEME	10
3.1 Robotik Sistemlerde Dış Dünyayla İletişim İçin Kullanılan Yöntemler.....	10
3.1.1 Encoder (Kodlayıcı).....	10
3.1.1.1 Mutlak Tip (Absolute) Encoder'lar.....	11
3.1.1.2 Artımsal Tip (Incremental) Encoder'lar.....	11
3.1.2 Devir Ölçer (Takometre)	13
3.1.3 Kameralı Sistemler	14
3.2 Filtreler	14
3.2.1 Highpass (Yüksek Geçiren) Filtre	14
3.2.2 Gaussian Lowpass (Alçak Geçiren) Filtre	15
3.2.3 Sobel Operator Filtresi.....	17
3.2.4 Laplace Filtresi	18
3.2.5 Kabartma Filtresi	21
3.3 Fabrika Uygulamaları.....	23
3.4 Robotik Sistem Uygulamaları	24
3.4.1 Çizgi Bulma	25
3.4.2 Orta Nokta Bulma.....	25
3.4.3 Renk Belirleme	25
3.4.4 Renk Takip Etme	26
4. DONANIM	27
4.1 Altera DE2-115 Eğitim Boardu.....	27
4.1.1 Yerleşim ve Elemanlar.....	28
4.1.2 DE2-115 Board'unun Blok Diyagramı.....	30
4.1.3 FPGA'in JTAG modunda ayarlanması.....	31
4.1.4 Basılan Düğmelerin ve Anahtarların Kullanımı	32
4.1.5 LED'lerin Kullanımı	33
4.1.6 7 Parçalı Göstergelerin Kullanımı	34
4.1.7 Saat Devresi	35
4.1.8 LCD Modülünün Kullanımı.....	36
4.1.9 VGA'nın Kullanımı	36
4.1.10 SRAM.....	39
4.1.11 SDRAM	40
4.1.12 FLASH.....	41
4.1.13 EEPROM	42

4.1.14	SD Kart	43
4.2	Terasic TRDB_D5M 5 MP Digital Camera Development Kit	43
4.3	QUARTUS II System Builder	46
4.3.1	Tasarım Akışı.....	46
4.3.2	DE2-115 Sistem Kurucu Kullanımı.....	47
5.	UYGULAMA	50
5.1	Yapılan Tanımlamalar	50
5.2	Sistem Tasarımı	51
5.2.1	Line Buffer (Satır Tamponu)	51
5.2.2	Multiply-Add (Çarpım-Toplam).....	51
5.2.3	Parallel Adder (Paralel Toplayıcı)	52
5.3	SİSTEMİN GERÇEKLEŞTİRİLMESİ.....	52
6.	SONUÇ VE ÖNERİLER	59
7.	KAYNAKLAR.....	61
8.	EKLER.....	66
8.1	EK-A	66
8.2	EK-B.....	80
8.2.1.1	FPGA Cihazı	80
8.2.1.2	FPGA Ayarlama.....	80
8.2.1.3	Hafıza Cihazları	80
8.2.1.4	SD Kart Yuvası	80
8.2.1.5	Bağlantılar	81
8.2.1.6	Saat.....	81
8.2.1.7	Ses	81
8.2.1.8	Ekran	81
8.2.1.9	Anahtarlar ve Göstergeler	82
8.2.1.10	Diğer Özellikler.....	82
8.2.1.11	Güç	82

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1: Basitleştirilmiş Programlanabilir Mantık Cihazı.....	4
Şekil 2.2: Karmaşık Programlanabilir Mantık Cihazı Örneği	5
Şekil 2.3: Tipik Bir Karmaşık Mantık Bloğu	5
Şekil 2.4: Xilinx CLB	6
Şekil 2.5: CLB'lerin FPGA Yapısı	7
Şekil 2.6: İlk Sayısal Görüntü.....	8
Şekil 3.1: Mutlak Tip Encoder Blok Diyagramı.....	11
Şekil 3.2: Artımsal Tip Encoder Blok Diyagramı	12
Şekil 3.3: Takometrenin Blok Diyagramı.....	13
Şekil 3.4: Yüksek Geçiren Filtre Uygulaması	15
Şekil 3.5: Alçak Geçiren Filtre Uygulaması.....	16
Şekil 3.6: Sobel Operatörü Çekirdekleri.....	17
Şekil 3.7: Sobel Filtresi Uygulaması	18
Şekil 3.8: Laplace Filtresi Çekirdekleri	19
Şekil 3.9: Laplace Filtresi Uygulaması.....	20
Şekil 3.10: Gaussian'ın Laplace'ı Uygulaması.....	21
Şekil 3.11: Kabartma Filtresi Uygulaması.....	22
Şekil 4.1: DE2-115 Paket İçeriği	27
Şekil 4.2: DE2-115 Board'u üstten görünüm.....	28
Şekil 4.3: DE2-115 Board'u alttan görünüm	29
Şekil 4.4: DE2-115 Board'unun blok diyagramı	31
Şekil 4.5: JTAG Ayarlama Şeması.....	32
Şekil 4.6: Düğmeler ve Cyclone IV E FPGA Arasındaki Bağlantılar.....	32
Şekil 4.7: Anahtarların Debounce Edilmesi	33
Şekil 4.8: Kaydırılabilir Anahtarlar ve Cyclone IV E FPGA arasındaki bağlantılar.....	33
Şekil 4.9: LED ve Cyclone IV E FPGA Arasındaki Bağlantılar.....	34
Şekil 4.10: 7 Parçalı Gösterge HEX0 ile Cyclone IV E FPGA arasındaki bağlantılar.....	35
Şekil 4.11: Saat Dağılımı Blok Diyagramı	35
Şekil 4.12: LCD Modülü ile Cyclone IV E PGA Arasındaki Bağlantılar	36
Şekil 4.13: FPGA ve VGA Arasındaki Bağlantılar	37
Şekil 4.14: VGA Yatay Zamanlama Tanımlaması	38
Şekil 4.15: FPGA ve SRAM Arasındaki Bağlantılar	40
Şekil 4.16: FPGA ve SDRAM Arasındaki Bağlantılar	41
Şekil 4.17: FPGA ve Flash Arasındaki Bağlantılar	42
Şekil 4.18: FPGA ve EEPROM Arasındaki Bağlantılar	42
Şekil 4.19: FPGA ve SD Kart Yuvası Arasındaki Bağlantılar	43
Şekil 4.20: TRDB-D5M Kamera Kiti.....	44
Şekil 4.21: Bir Tasarım Üretiminin Genel Akış Şeması.....	47
Şekil 4.22: DE2-115 Sistem Kurucu Kullanıcı Ara Yüzü.....	48
Şekil 5.1: Sistemde Gerçek Kamera Görüntüsü	53
Şekil 5.2: Sistemde Gri Ölçekli Görüntü.....	54
Şekil 5.3: Düşük Eşik Değerli Sobel Filtresi.....	55
Şekil 5.4: Yüksek Eşik Değerli Sobel Filtresi	56

Şekil 5.5: Çalışan Sistemin Görüntüsü	57
--	-----------

TABLO LİSTESİ

	<u>Sayfa</u>
Tablo 4.1: Paket İçeriği	28
Tablo 4.2: VGA Yatay Zamanlama Tanımlaması	38
Tablo 4.3: VGA Dikey Zaman Tanımlaması	39
Tablo 4.4: Anahtar Performans Parametreleri.....	45

SEMBOL VE KISALTMA LİSTESİ

TTL	: Transistor-Transistor Logic (Transistör-Transistör Mantığı)
V	: Volt
mW	: mili Watt
nm	: Nanometre
PCB	: Printed Circuit Board (Baskılı Devre Kartı)
s	: Saniye
µs	: Mikro Saniye
ASIC	: Application Specific Integrated Circuit; (Uygulamaya Özel Bütünleşik Devre)
Hz	: Hertz
kHz	: Kilo Hertz
mHz	: Mega Hertz
Mbps	: Mega bit per second (saniye başına mega bit)
Fps	: Frame per second (saniye başına kare)
Kb	: Kilo Bayt
Mb	: Mega Bayt
Kbit	: Kilo bit
Mbit	: Mega bit
CPLD	: Complex Programmable Logic Device (Karmaşık Programlanabilir Mantık Cihaz)
CRT	: Cathode Ray Tube (Katot Işınlı Tüp)
CNC	: Computer Numerical Control (Bilgisayarlı Sayısal Kontrol)
RAM	: Random Access Memory (Rastgele Erişimli Bellek)
SRAM	: Statik RAM (Statik Rastgele Erişimli Bellek)
SDRAM	: Synchronous Dynamic RAM (Senkron Dinamik Rastgele Erişimli Bellek)
CMOS	: Complementary Metal-Oxide-Semiconductor(Bütünleyici Metal Oksit Yarıiletken)
CCD	: Charge-Coupled Device (Yüklenme İliştirilmiş Araç)
LED	: Light Emitting Diode (Işık Saçan Diyot)
MUX	: Multiplexer (Çoklayıcı)
LUT	: Look-Up Table (Başvuru Tablosu)

PLL	: Phase-Locked Loop
RS232	: Recommended Standard 232
VGA	: Video Graphics Array (Video Grafik Dizisi)
RGB	: Red-Green-Blue (Kırmızı-Yeşil-Mavi)
ADC	: Analog Digital Converter (Analog Sayısal Dönüştürücü)
DAC	: Digital Analog Converter (Sayısal Analog Dönüştürücü)
I/O	: In/Out (Giriş/Çıkış)
PLA	: Programmable Logic Array (Programlanabilir Lojik Dizileri)
PAL	: Programmable Array Logic (Programlanabilir Diziler')
FPGA	: Field-Programmable Gate Array (Alanda Programlanabilir Kapı Dizileri)

ÖNSÖZ

Bu çalışmada FPGA platformu üzerinde görüntü işleme ve filtreleme işlemlerinin yapılması ve FPGA platformunun etkin kullanımı hedeflenmiştir.

Literatürde görüntü işleme uygulamaları üzerine yapılan karşılaştırmalarda FPGA platformunun gömülü uygulamalar için en uygun mikrokontrolör olduğu, daha düşük çalışma frekansına rağmen paralel işlem yapabilme yetisiyle masaüstü bilgisayarlara eşdeğer veya daha iyi performans verdiği görülmüştür[15,28,32,34]. Fakat literatürdekinin aksine Xilinx yerine Altera marka FPGA boardu ve buna bağlı olarak da ISE Webpack yerine Quartus II yazılımı kullanılmıştır. Kenar belirleme filtreleri içerisinde ise literatürde de sıkça kullanılan[5,22,23,24,29,30,33,34] Sobel filtresinin uygulaması tercih edilmiştir.

Bu tez çalışması için öncelikle görüntü işleme ve verilog üzerine çalışılmış daha sonra sırasıyla kameradan görüntü elde edilmesi, ayrıştırılması, Sobel filtresi tasarımı, filtrenin gerçekleştirilmesi için line buffer uygulanması ve son olarak Sobel filtresinin gerçek zamanlı olarak uygulanması gerçekleştirilmiştir. Uygulamada amaç olarak gerçek zamanlı olarak kameradan alınan görüntüye Sobel filtresi uygulamak ve böylece nesnelere kenarlarını belirlemek seçilmiştir.

Çalışma sonucunda hedeflenen amaca ulaşılmıştır. Uygulanan filtrenin farklı eşik değerlerindeki uygulamaları görülmüş ve karşılaştırılmıştır. İleride yapılacak çalışmalar için bir temel oluşturulmuştur.

Bu çalışmamda bana sabır gösteren ve yardımcı olan danışmanım Yrd. Doç. Dr. Davut AKDAŞ'a, yine yardımlarını ve desteğini esirgemeyen Arş. Gör. Dr. Sabri BİCAKCI'ya ve bana her zaman destek olan ve beni motive eden aileme teşekkür etmeyi bir borç bilirim.

1. GİRİŞ

Robotik sistemlerin gelişim süreci boyunca arařtırmacıların hedefi her zaman çevreyle olan etkileşimi en yüksek düzeye çıkarabilmek olmuştur. Farklı sistemler aracılığıyla robotların kendi kendisine çalışabilir hale getirilebilmesi için uğraşı verilmiştir. Bu sistemlere örnek olarak düşük seviyede sensörler verilebilir. Sensörler yardımıyla çevreyle etkileşim sağlanmış, kendi kendisine çalışabilen robotlar geliştirilmiştir. Günümüzde otomasyon sistemlerinde [18,26,27] kamera kullanımı ile görüntü işleme yapılarak insansız üretimler sağlanmaktadır. İnsansı robotlarda ise hafızasında kayıtlı olan nesnelere tanıma, engelleri fark etme gibi uygulamaları geliştirilmektedir.

Görüntü işlemenin temeli ilk sayısal görüntüye kadar dayandırılabilir. Alınan görüntünün sayısal olarak kaydedilmesi için yapılan işlemler elbette ki görüntü işleme olarak adlandırılabilir. İşlemci hızlarının artık bilgisayar boyutlarının küçülmesi ile daha küçük boyutlu sistemler kurmak ve daha hızlı işlem yapmak mümkündür. Bu küçülme sonucunda akıllı telefonlarda bile görüntü işleme uygulamalarını görmek mümkün olmuştur. Örnek olarak telefon kilidinin yüz tanıma sistemiyle açılması veya ekrana bakıldığı sürece ekranın açık kalması verilebilir. Bunun dışında fabrikalarda insansız üretim için ve süreç takibi için uygulamalar da yapılmaktadır.

Bu tez çalışmasında, FPGA platformu kullanılarak bir görüntü işleme sistemi tasarlanmıştır. Paralel işlem kapasitesi kullanılarak gerçek zamanlı bir kenar belirleme algoritma uygulaması yerleştirilmiş ve başarıyla çalıştırılmıştır. Robotik sistemlerde kullanımı için gerekli altyapı sağlanmıştır.

Ulaşılabilen literatürde bilgisayarlı ve FPGA'li görüntü işleme sistemleri Karl Pauwels ve arkadaşları [15], Mario-Alberto Ibarra-Manzano, Dora-Luz Almanza-Ojeda [28], Bogdan Belean ve arkadaşları [21], Bhupinder Verma ve H.K. Serdana [32] ve N. Sudha ve Aruppukottai Rajarathinam Mohan[34] tarafından karşılaştırılmış ve FPGA'lerin gömülü sistem uygulamaları için uygun olduğu belirtilmiştir.

Daniel Seidner [39,40]'ta FPGA üzerinde düşük maliyetli görüntü işleme mimarisi için harici bir satır hafızası geliştirerek yüksek çözünürlüklü görüntülerin de düşük hafızaya sahip FPGA'ler tarafından işlenebilmesine olanak tanımıştır.

Ikram E. Abdou ve William K. Pratt yaptıkları çalışmada [24] Sobel filtresinin eşik/kenar belirleme için en doğru sonuç veren filtre olduğunu diğer filtrelerle (Roberts ve Prewitt) karşılaştırarak belirlemiştir.

Bunun yanı sıra ulaşılabilen literatürdeki çalışmaların tamamında Xilinx marka FPGA boardu ve ISE Webpack yazılımı kullanılmıştır.

İkinci bölümde FPGA platformunun ve Sayısal Görüntü İşleme üzerine literatürün gelişiminden bahsedilmiştir.

Üçüncü bölümde robotikte görüntü işleme konusu, yöntemleri, alternatifleri, gerçek yaşam ve araştırma amaçlı uygulamalarıyla anlatılmıştır. Görüntü işlemede kullanılan filtreler niteliksel ve niceliksel olarak analiz edilmiştir.

Dördüncü bölümde kullanılan donanım ve yazılımlar kapsamlı bir şekilde incelenmiştir. Donanımsal elemanlar, bağlantı noktaları, giriş ve çıkışlar ve ayarlanmalarından bahsedilmiştir.

Beşinci bölümde ise yapılan uygulamada kullanılan modüller, sistem tasarımı anlatılmış, sistemin çalışır haldeki görüntüleri eklenmiştir. Elde edilen sonuçlar ve yapılan karşılaştırmalar verilmiş ve yorumlanmıştır.

2. TARİHÇE

2.1 FPGA'lerin gelişimi

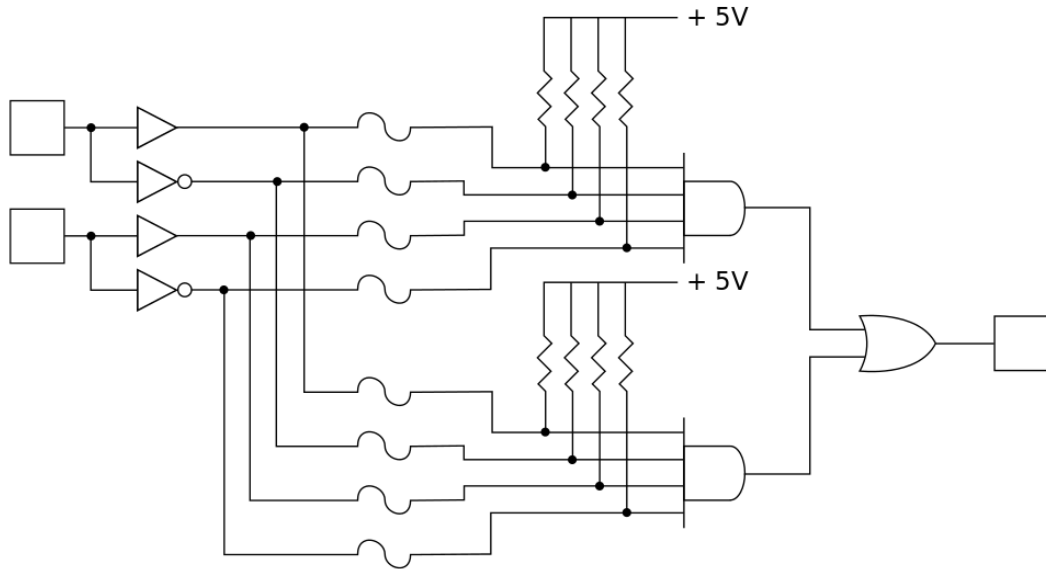
1970'lerde sayısal mantık donanımının fikri başlangıcıyla muazzam sayıda farklı cihaz ortaya çıkmıştır. Bugün çok yaygın şekilde kullanılan TTL (transistor – transistor logic) mantığıyla çalışan 7400/5400 entegre devre serisine ve CMOS (Complementary Metal Oxide Semiconductor; Bütünleyici Metal Oksit Yarıiletken) teknolojisine ön ayak olmuştur. Bu teknolojiler PCB (Printed Circuit Board; Baskılı Devre Kartı) tasarımında halen kullanılmaktadır. Bunun yanında son 30 yıldır temel sayısal cihazlar için programlamanın daha kullanışlı hale getirilmesi çabaları da sürmektedir. TTL mantığının sıklıkla kullanılması yüksek hızlı anahtarlama yapabilmesi, düşük güç tüketimi ve düşük maliyetinden dolayıdır.

Farklı devre tasarım tekniklerinin araştırılmasına ihtiyaç duyulması çoğu sayısal sistemde iki farklı tasarım yaklaşımının kullanılmasından ileri gelmektedir. Donanım tarafında hedef sıklıkla en ileri performans içindir; daha hızlı, daha küçük, daha düşük güçlü ve daha ucuza doğru bir ilerleme vardır. Bu da her bir yonganın ayrı ayrı tasarlanmasını, yerleştirilmesini, üretilmesini ve paketlenmesini gerektiren özel bütünleşik devre tasarımına (ASIC: Application Specific Integrated Circuit; Uygulamaya Özel Bütünleşik Devre) yol açmıştır. Geniş çaplı üretimler için bu yöntem maliyet açısından uygundur ama açık olarak görüleceği üzere bu çok pahalı (silikon işleme sürecinde sadece maskeler \$500000 üzerinde tutabilir) ve zaman alıcı (bir yıl kadar) bir yaklaşımdır[1].

Yazılımsal açıdan ise yaklaşım daha çok Intel Pentium, PowerPC veya ARM gibi standart bir işlemci mimarisi kullanmak ve böyle bir platforma yüklenebilen yazılım uygulamaları geliştirmek şeklindedir. Bu yaklaşım yöntemi geliştirme veya uyarılama yapmak için daha hızlıdır. Fakat bununla birlikte işletim sistemine duyulan ihtiyaç ve derleyici(compiler) verimsizliğinden kaynaklanan, işlemci üzerindeki ek yük ve donanım ile yazılım arasındaki dolaylı (doğrudan olmayan) ilişkiden kaynaklanan performans kaybı mevcuttur.

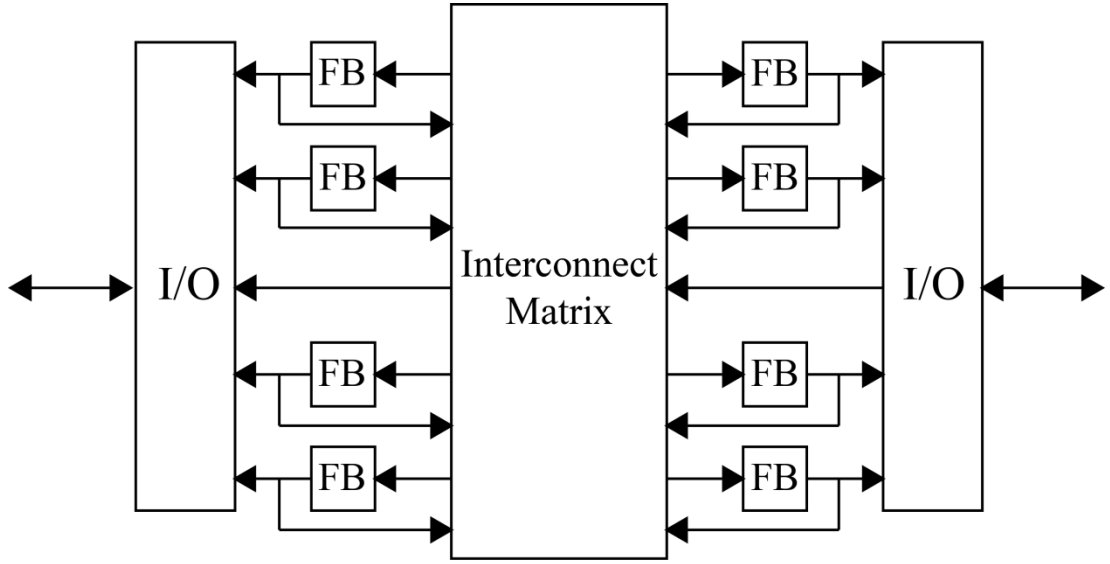
Sonuç olarak programlanabilir cihazlar; yüksek performanslı bir platformda donanım tasarımı olan, optimum kaynaklara sahip, işletim sistemine ihtiyaç duymayan, yeniden ayarlanabilen ve cihazların tekrar programlanabildiği bir orta yol yaklaşımı olarak geliştirilmiştir.

Bu yaklaşım neticesinde ortaya çıkan, programlanabilen ilk cihaz türü Programlanabilir Dizi Mantığı'ydı(PAL: Programmable Array Logic). Bu cihaz bağlantı dizileri kullanılarak birbirine bağlanabilen mantık kapısı dizisinden oluşmaktadır. Bu cihazlar az sayıda flip-flop destekleyebilir (genellikle 10'dan az) ve küçük durum makinelerini uygulayabilir (Şekil 2.1).



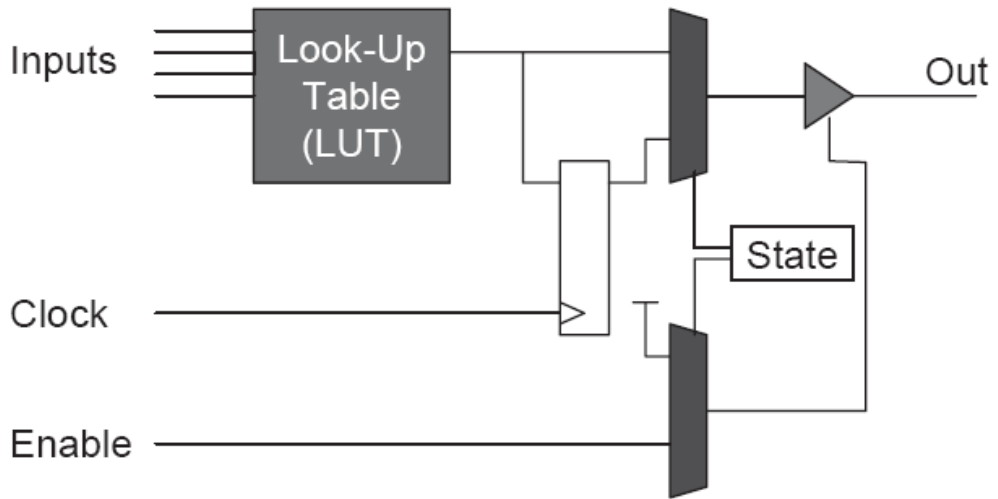
Şekil 2.1: Basitleştirilmiş Programlanabilir Mantık Cihazı

Karmaşık Programlanabilir Mantık Cihazları (CPLD: Complex Programmable Logic Device) basit PAL cihazlarının kısıtlamalarını çözmek için geliştirilmiştir. Bu cihazlar PAL'larla aynı temel ilkeyi kullanır fakat yönlendirme blokları kullanarak bağlanan makro-blok dizilerine (her biri yaklaşık olarak bir PAL'a eşit) sahiptir (Şekil 2.2).



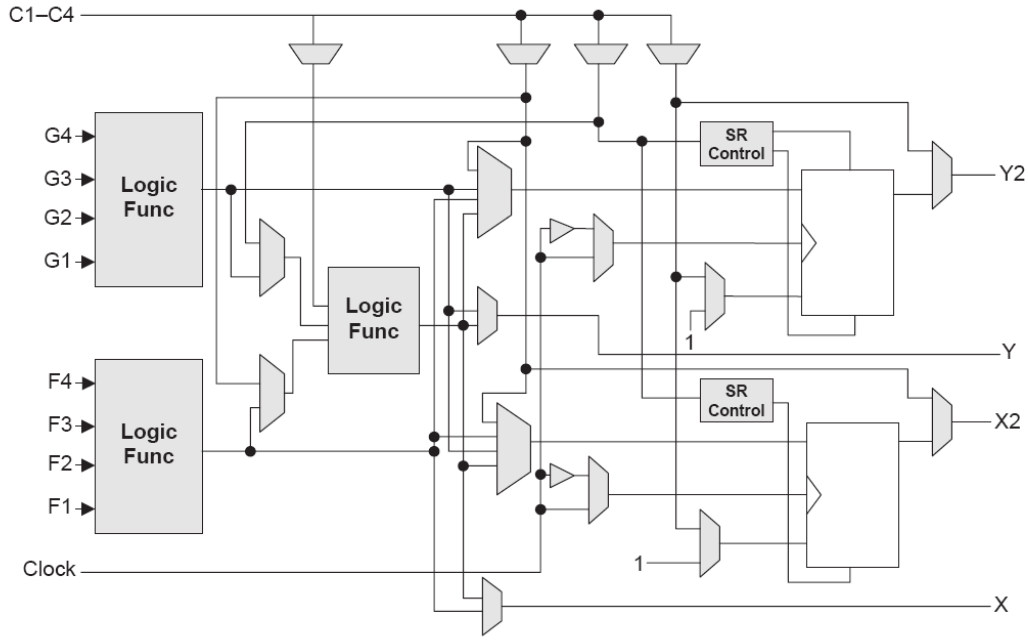
Şekil 2.2: Karmaşık Programlanabilir Mantık Cihazı Örneği

FPGA'ler (Field Programmable Gate Array: Alan Programlanabilir Kapı Dizileri) CPLD'den sonraki adımdır. FPGA sabit kapı dizisi yerine Karmaşık Mantık Bloğu (CLB: Complex Logic Block) kavramını kullanır. FPGA'ler cihaz üzerinde yönlendirmenin yanı sıra her mantık bloğunun da en iyi olacak şekilde konfigüre edilebilmesine izin veren cihazlardır. Tipik bir CLB cihazı Şekil 2.3'te gösterilmiştir.



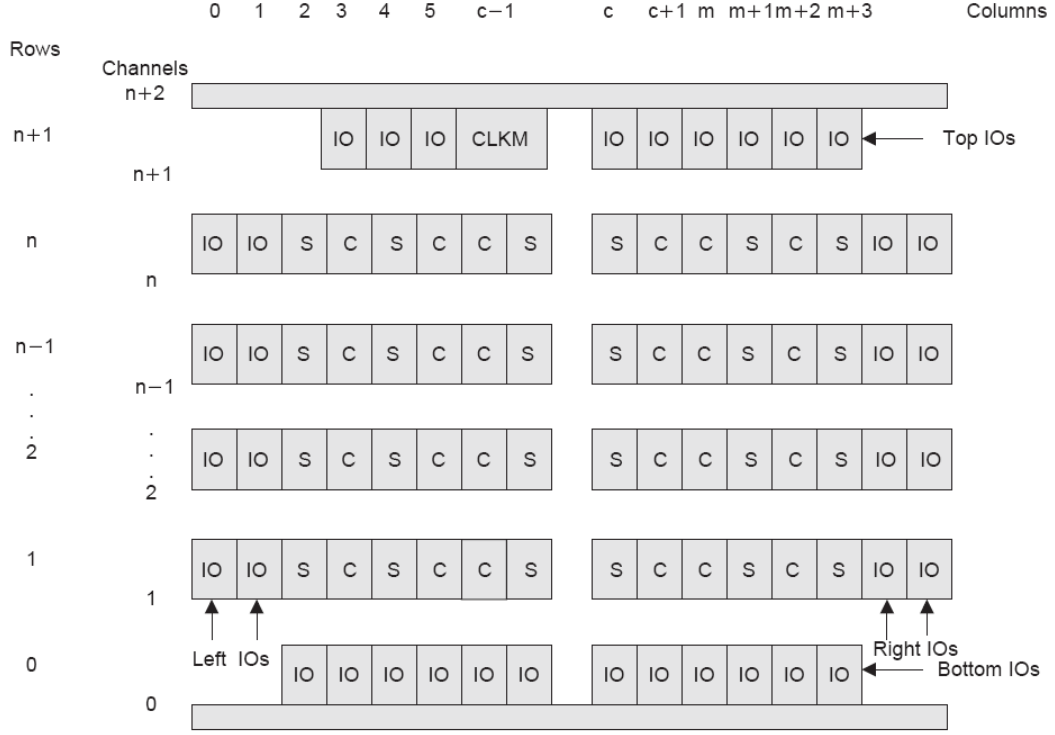
Şekil 2.3: Tipik Bir Karmaşık Mantık Bloğu

CLB'nin programlandığında belirli bir mantık fonksiyonu vermeye ayarlanabilen bir Arama Tablosu (LUT: Lookup Table) vardır. Ayrıca etkinleştirme sinyaliyle birlikte CLB'nin kombinyoryal (combinatorial) (saatsiz) veya senkron (saatli çalışmasına izin veren saatli (clocked) d-tipi bir flip-flopa sahiptir. Şekil 2.4'te gösterilen Xilinx marka CLB'de 4 girişli iki LUT ve çeşitli çoklayıcı (multiplexer) ve flip-floplar gerçek bir cihaz üzerinde görülebilmektedir.



Şekil 2.4: Xilinx CLB

Tipik bir FPGA'de tek bir yonga üzerinde oldukça karmaşık cihazların uygulanabilmesini ve kolayca ayarlanabilmesini sağlayan yüzlerce veya binlerce CLB bulunur. Modern FPGA'ler tek bir cihaz üzerinde birden fazla 32-bit işlemci tutabilecek kadar kapasiteye sahiptir. Tipik bir FPGA yerleşimi (CLB yönünden) Şekil 2.5'te gösterilmiştir[1].



Şekil 2.5: CLB'lerin FPGA Yapısı

2.2 Sayısal Görüntü İşlemenin Gelişimi

Elektronik görüntü tarayıcıları 1930'larda basım endüstrisinde renk ayrımı yapmak için kullanılmıştır. Fakat bunlar elektronik olarak görüntü verisini kaydetmeyen, elektronik sinyalleri taramayla eşzamanlı olarak filme kayıt yapmak için kullanmış analog cihazlardır. Şimdiki bildiğimiz anlamda ilk sayısal görüntü (şekil 2-6) 1957 yılında Russell Kirsch ve iş arkadaşları tarafından National Bureau of Standards'ta (Ulusal Standartlar Bürosu) oluşturulmuştur [6]. Bu görüntüyü oluşturan sistem silindirik bir tarayıcı ve bilgisayarda görüntünün saklanması sağlayan sayısal veriyi üreten foto-çoklayıcı tüpünden oluşmaktadır. Foto-çoklayıcı tüpler elektromanyetik spektrumda ultraviyole, görünen ve yakın-kızılötesi aralıklara yüksek derecede hassas vakum tüpleridir.

İlk sayısal kamera tasarımları da bu tarama fikirleri üzerine temellendirilmiştir. Bu yüzden bu cihazlarla fotoğraf çekmek oldukça uzun bir zaman alıyordu ve son tüketicinin kullanımına uygun değildi. Teknolojinin gelişimi için ordu oldukça etkili olmuş ve 1976 yılında ilk sayısal gözetleme uydusu olan KH-11'in icadını sağlayan araştırmayı desteklemiştir. KH-11 ile ilk kez gerçek zamanlı

optik gözlem yapmak mümkün olmuştur. Daha önceki uydular görüntüyü filme kaydediyor ve içinde film olan bir teneke kutuyu havada bir uçak tarafından yakalanması için fırlatıyordu. KH-11'deki sistemin kısıtlı bant genişliği 1970'lerde görüntü kodlama ve sıkıştırma araştırmaları için büyük bir isteklendirme kaynağı olmuştur.



Şekil 2.6: İlk Sayısal Görüntü

CCD'ler (Charge Coupled Device: [elektriksel] Yük Bağlaşık Cihaz) ilk olarak 1969 yılında Willard Boyle ve George E. Smith tarafından AT&T Bell Labs'te icat edilmiştir. Konsepti açıklayan ilk makalede kullanılabilir amaçlar hafıza, gecikme hattı ve görüntüleme olarak listelenmiştir. Entegre devre ile çalışan ilk CCD uygulaması 8 bitlik basit bir kaydırma kaydedicisi devresidir. 1971 yılında ise Bell Laboratuvarı araştırmacılarından Michael F. Tompsett ve arkadaşları basit çizgisel cihazlarla görüntü yakalamayı başarmışlardır. CCD'lerin sıra düzeniyle geliştirilmesi sayısal kameraları mümkün kıldı. Bu cihazları kullanan ilk görüntüleme sistemleri 1973 yılında astronomik teleskoplardır.

Fairchild Semiconductor, RCA ve Texas Instruments'ın da dâhil olduğu birçok şirket bu icat üzerine geliştirme programları başlatmıştır. Fairchild'da eski bir Bell araştırmacısı olan Gil Amelio önderliğinde yürütülen araştırmalar sonucunda ilk ticari ürün ortaya çıkmıştır. 1974 yılında çizgisel 500 elemanlı cihaz ve 2 boyutlu 100*100 piksellik cihaz üretilmiştir. 1975 yılında Kodak'ta çalışan bir elektrik mühendisi olan Steven Sasson tarafından Fairchild'ın 100*100'lük CCD'si kullanılarak ilk sayısal fotoğraf makinesi icat edilmiştir [3]. İlk sayısal gözetleme uydusu olan ve 1976 yılında uzaya fırlatılan KH-11'de ise 800*800'lük CCD kullanılmıştır[4]. İlk siyah beyaz sayısal kameralar 1980'lerde kullanılmaya başlanmıştır fakat bunlar deneysel kullanımlar ve araştırma kullanımlarıyla kısıtlı kalmıştır. Teknolojinin gelişimi 1980'lerde tüketiciye yönelik video kayıt cihazını mümkün kılmıştır. Ancak CCD'lerin düşük çözünürlükleri nedeniyle bu cihazlarda kullanımı kısıtlı kalmıştır. Bu dönemdeki CCD'lerde renk üretimi için elektronik tüp kullanan televizyon kamerası gibi üç filtre ve üç cihaz kullanılmıştır. Daha sonra ise CCD dizilerine mozaik filtrelerin CCD elemanlarının üstüne yerleştirilmesi yöntemiyle doğrudan renk eklenmiştir. Her eleman üç bantlı görüntünün bir bandını kaydetmektedir. Tam çözünürlüklü renkli görüntü üç sinyalin uzamsal interpolasyonu ile elde edilir. Bu yöntem günümüzdeki renkli fotoğraf makinelerinin de temel yöntemidir [2].

3. ROBOTİKTE GÖRÜNTÜ İŞLEME

3.1 Robotik Sistemlerin Dış Dünyayla İletişimi İçin Kullanılan Yöntemler

Robotik sistemler, özellikle tam otonom sistemler yapıları gereği dış dünyayla iletişim kurmak, çevresini ve çevresinde olanları değerlendirmek zorundadır. Bu iletişim temel olarak ikiye ayrılabilir, çevrenin durumunu değiştirmek (bir yerden kaldırma ve bir yere yerleştirme işlemleri) ve güç değiş-tokuşu (birleştirme veya yüzey düzeltme işlemleri). Bu iletişimi sağlamak için ise çeşitli yöntemler kullanılabilir. Temel seviyede iletişim kurmak için kullanılacak yöntem çeşitli sensörler kullanmaktır. Örnek olarak encoder'lar (kodlayıcılar) ve tachometer'lar (devir ölçerler) verilebilir.

Daha üst seviye iletişim kurmak için ise kamera kullanmak dolayısıyla görüntü işleme yolunu tercih etmek gerekmektedir. Sisteme entegre edilecek kamera modülü ve kameradan gelen görüntünün işlenmesi ve değerlendirilmesi sonucu alınan veri ile robotik sistemlerin kontrolü mümkündür. Görüntü işleme için ulaşılabilen literatürde kullanıldığı görülen filtreler ve kenar belirleme yöntemleri Gaus filtresi [5], [7], [22] yüksek ve alçak geçiren filtre [7], Shock filtresi [21] ve Sobel filtresidir [22], [23], [24].

Bahsedilen iletişim yöntemleri aşağıda detaylandırılmış, çalışma yöntemleri açıklanmıştır.

3.1.1 Encoder (Kodlayıcı)

Encoder'lar yani sinyal üreticiler, bağlı olduğu shaftın hareketine karşılık, sayısal (dijital) bir elektrik sinyali üreten elektromekanik bir cihazdır. Encoder'lar çalışma şekillerine göre; dönel olarak çalışan shaft encoder'lar ve doğrusal olarak çalışan linear encoder'lar olmak üzere ikiye ayrılırlar[8].

Testere dişli bir kol ya da ölçü tekerleği gibi uygun bir mekanik dönüştürme cihazı ile birlikte kullanıldığı zaman shaft encoder'lar,

- Açısal yer değiştirme,
- Lineer ve dairesel hareket,

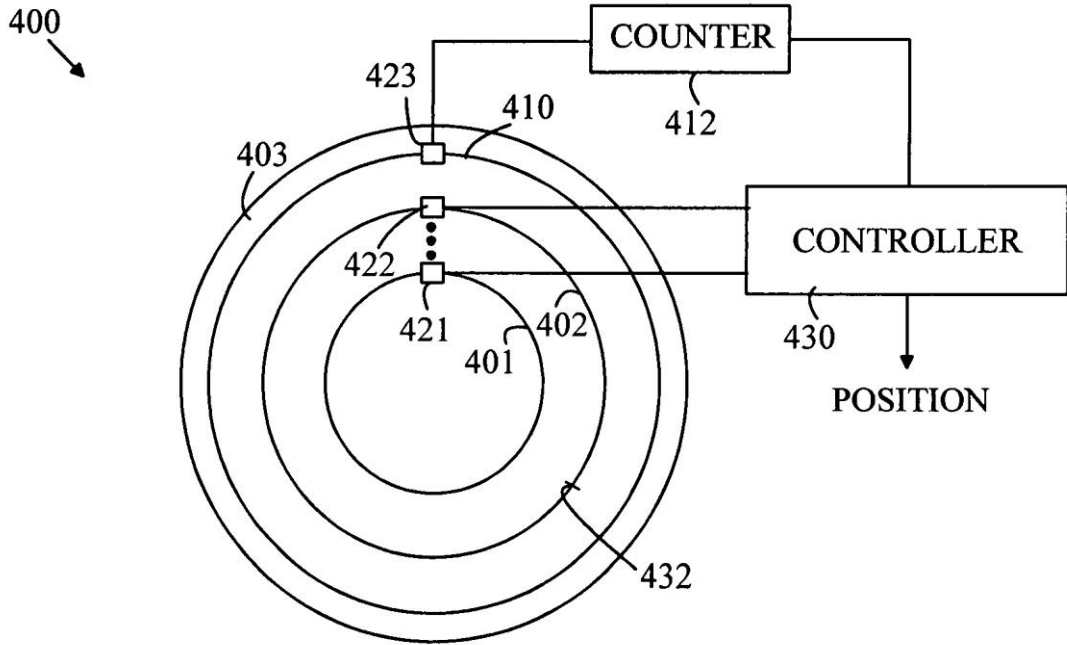
- Dönüş hızı,
- İvme,

gibi büyüklükleri ölçmek için kullanılabilir.

Çıkış tiplerine göre de ikiye ayrılırlar. Bunlar sırasıyla Mutlak tip ve Artımsal tip'tir.

3.1.1.1 Mutlak Tip (Absolute) Encoder'lar

Bu encoder'lar, her pozisyonlarında farklı sayılardaki bitlerden oluşan dijital bit dizileri şeklinde birbirine benzemeyen çıkışlar üreterek, gerçek pozisyonlarını tam olarak gösterirler. Enerjisi kesilse bile mevcut durumunu korur. Şekil 3-1'de mutlak tip encoder blok diyagramı görülmektedir[9].



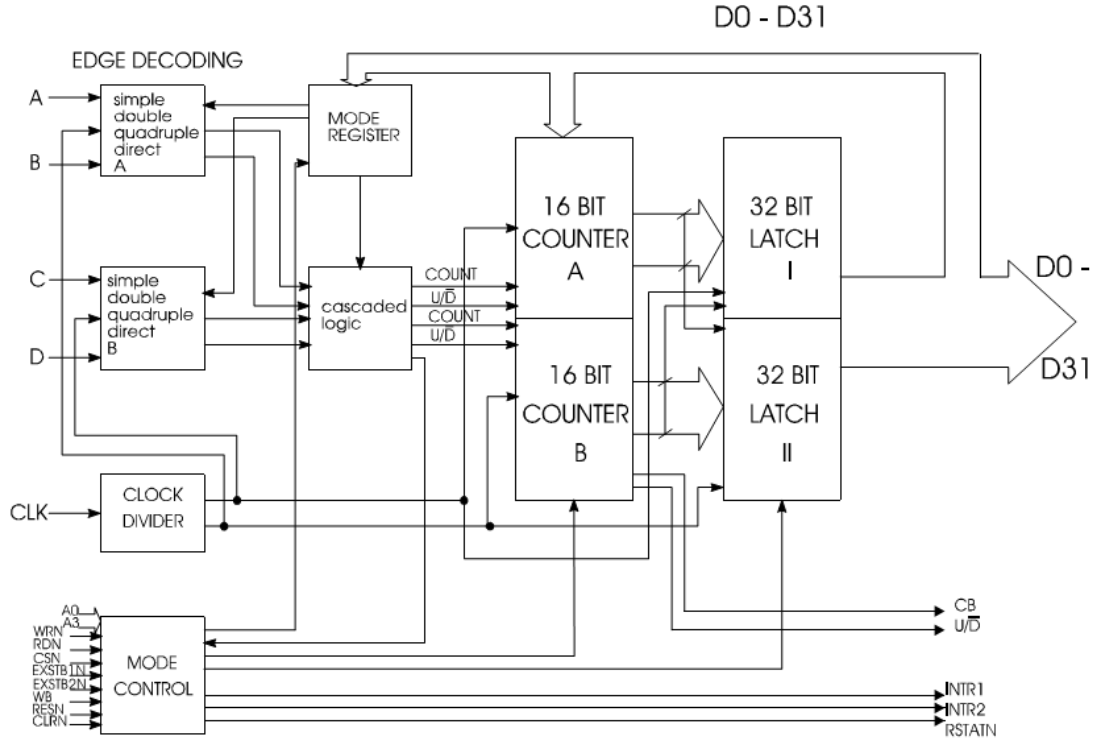
Şekil 3.1: Mutlak Tip Encoder Blok Diyagramı

3.1.1.2 Artımsal Tip (Incremental) Encoder'lar

Bu encoder'lar, her pozisyonda benzer çıkış sinyalleri (kare dalga) üretirler. Bu sinyaller hız ölçümü (bir devir ölçer ile birlikte) ya da sayma işlemi için (bir

Sayıcı ile birlikte) kullanılabilirler. Şekil 3-2’de artımsal tip encoder için blok diyagramı görülebilir.

Bu tip encoder’lar absolute encoderlar’dan farklı olarak enerjilendikleri anda buldukları yeri 0 kabul eder ve bunun üstüne işlem yapar. Absolute encoder’lar, incremental encoder’lara göre maliyeti daha yüksektir.



Şekil 3.2: Artımsal Tip Encoder Blok Diyagramı

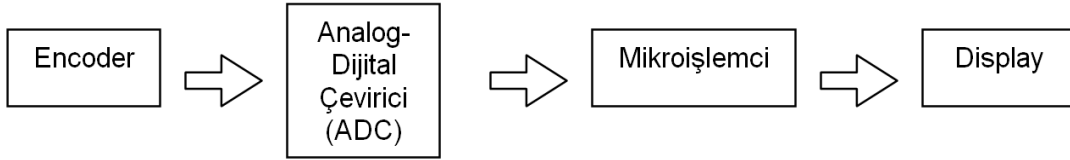
512 / 1024 / 1048 / 4096 pulse (darbe) ve üstü encoder’lar mevcuttur. Bu pulse’lar encoder ‘ın 1 tam tur attığında üreteceği pulse sayısıdır. Doğal olarak 4096 pulse’lı bir encoder daha hassastır. Çünkü nokta aralıkları daha azdır.

Örneğin bir motora bağlı encoder’ın pozisyonuna göre motorun kontrolünü sağlayabiliriz, hangi pozisyonda olduğuna göre feedback (geri besleme) olarak işlemlerimizi ona göre düzenleyebiliriz.

3.1.2 Devir Ölçer (Takometre)

Dönen bir cismin devir sayısını bir hız birimi cinsinden (dev/dak, rpm, m/s, km/h) veren cihazlara takometre denir.

Temel takometre devresinin akış diyagramı Şekil 3.2’de görülebilir. Dönen mil ya da tekerlekten alınan bilgi “Encoder” ‘da analog bilgiye çevrilir, “ADC” ‘de analog bilgi dijitale çevrilir. Mikroişlemci yardımıyla birim çevirmeleri ve hesaplamalar yapılarak dijital bilgi ikilikten onluk sayı sistemine çevrilir ve ekran vasıtasıyla sonucu sayısal olarak görülebilir. Bu tür takometreler dijital sonuç verdiği için bunlara “Dijital takometre” denir.



Şekil 3.3: Takometrenin Blok Diyagramı

Takometreler sanayideki farklı alanlarda farklı amaçlarda kullanım olanağına sahiptirler. Örneğin kumaş sarma makinalarında kumaşın sarıldığı lüvert isimli dev makaranın dönüş sayısından yola çıkarak, mikro işlemciler sayesinde kumaşın kaç metre olduğunu bize takometreye benzer bir sistem verir. Yine tekstil sektöründe dokunacak kumaşın ipliğinin boyu yine aynı yöntemle ölçülür.

Plastik enjeksiyon makinalarında kalıp boşluğuna doldurulacak hammaddenin eritilip aynı zamanda ilerletildiği “vida” kısmının dönüş sayısı yine bu yöntemle kontrol edilir.

Günümüzde kullanılan takım tezgâhlarının birçoğu nümerik kontrollü olup bunlara kısaca CNC tezgahları denir. Bu makinalarının kendi bilgisayarlarına yüklenen programlar dâhilinde kesme, delme, diş açma gibi birçok işlem yapılabilir. Program temel programlama mantığıyla aynı olup önce kullanılacak takım tanıtılıp sonra yapılacak işleve göre koordinatlar bilgisayara girilir. Verilen bu koordinatlara gitmesinde yine buna benzer bir mantık kullanılır.

Sanayide bazı üretim hatlarında iş süresinin kısılması, iş kalitesinin artması ve işçi sağlığı gibi sebeplerden ötürü 3 ilerleme ve 3 dönme olmak üzere kullanılan robotların istenen koordinatlarda istenen pozisyona geçmesi yine bu sisteme benzer şekilde yapılır.

3.1.3 Kameralı Sistemler

Robotik sistemlerde en gelişmiş ve karmaşık iletişim yöntemi ise görüntü işlemedir. Dış dünyanın bir veya daha fazla kamera aracılığıyla gözlenmesi ve gerçek zamanlı olarak bu görüntünün işlenerek ihtiyaç duyulan kısımlarının alınması ile gerekli iletişim gerçekleştirilir. Bu yöntem için kullanılan metotlar ve sistemler aşağıda ayrıntılı bir şekilde incelenmiştir.

3.2 Filtreler

Görüntü işleme yöntemlerinin uygulanabilmesi için ilk önce alınan görüntünün istenen boyuta, renk düzenine ve formata getirilmesi; kullanılacak kısımların vurgulanması veya diğer bölgelerden yalıtılması gerekmektedir. Bunun için çeşitli filtreler uygulanabilir/uygulanması gerekmektedir.

3.2.1 Highpass (Yüksek Geçiren) Filtre

Yüksek geçiren filtre çoğu keskinleştirme metodunun temelidir. Bir görüntüde karanlık veya aydınlık bakımından aralarında az farklılık olan bitişik alanlardaki kontrast artırıldığında görüntü keskinleşmiş olur.[7]

Görüntüdeki bulanıklığı, tekrarlanan pikselleri azaltmak için kullanılır. Keskinleştirme işlemi görür. Görüntü keskinleştirmeyi pikseller arasındaki farkı arttırarak yapar. Negatif değer verilen piksellerin ağırlığıyla değişiklik yapılan pikselin ağırlığı arasındaki fark arttıkça ana piksel değerindeki değişim de o kadar artar. Keskinleştirme derecesi merkez ağırlığının değiştirilmesiyle ayarlanabilir [11].

$$\begin{bmatrix} 0 & -2 & 0 \\ -2 & 11 & -2 \\ 0 & -2 & 0 \end{bmatrix} / 3 + 0 \quad (1)$$



Şekil 3.4: Yüksek Geçiren Filtre Uygulaması

3.2.2 Gaussian Lowpass (Alçak Geçiren) Filtre

İki boyutlu filtrelerde, Gaussian filtresinde olduğu gibi, filtre çekirdeği kullanılır [5]. Bu filtre görüntüdeki gözle görülebilir ölçekteki renk geçişlerinin yerlerini saptar ve daha sonra kenarları yumuşatmak için ara renkler yaratır. Merkezdeki değer değiştirilerek filtrenin ağırlığı değiştirilebilir. Bu filtre sonucunda oluşan çıkış hücresi (pikseli) filtrenin 3*3'lük penceresine giren hücre (piksel) değerlerinin aritmetik ortalamasıdır [7].

Örnekteki filtrede görüldüğü üzere piksellere verilen ağırlık merkezden uzaklaştıkça azalmaktadır bu da filtrenin dairesel bir etki oluşturmasını sağlayacaktır. Bu tarz yumuşatma odak dışı lensin oluşturacağı etkiyi yaratır [11].

$$g(x) = e^{-x^2/2\sigma^2} [22] \quad (2)$$

Gaus filtresi özellikle gürültünün normal dağılımının (Gaus dağılımı) olduğu görüntülerde, gürültüyü azaltmada etkilidir [22].



Şekil 3.5: Alçak Geçiren Filtre Uygulaması

3.2.3 Sobel Operator Filtresi

Sobel operatörü görüntü üzerinde iki boyutlu uzaysal gradyan ölçümü gerçekleştirir ve böylece kenarlara tekabül eden yüksek uzaysal frekans bölgelerini vurgular. Başka bir deyişle filtreleri uyguladığımız uzaysal(spatial) bölgedeki yüksek değerlikli yani komşu bölgelerinden renk ve kontrast olarak ayrılan alanlara vurgu yaparak kenarları belirler. Genellikle gri tonlamalı bir girdi görüntüsünün her noktasındaki yaklaşık mutlak gradyan büyüklüğünü bulmak için kullanılır. Operatör şekil 3-4'te görüldüğü gibi bir çift 3x3'lük konvolüsyon filtresinden (çekirdeğinden) oluşmaktadır. Bir çekirdek diğerrinin 90° döndürülmüşüdür[10].

$$\begin{array}{ccc} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{array} \qquad \begin{array}{ccc} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{array}$$

G_x **G_y**

Şekil 3.6: Sobel Operatörü Çekirdekleri

Sobel filtresi 3 adımlı bir işlemdir. Her iki çekirdek içinde görüldüğü gibi ağırlıklar toplamı 0'dır. Bu iki filtrenin kullanılmasındaki amaç sırasıyla x ve y'de yaklaşık türev almaktır. Filtrelerin sonuçlarına $D_x(x, y)$ ve $D_y(x, y)$ diyelim. D_x de D_y de pozitif veya negatif değerler alabilir bu yüzden sonuca 0.5 eklemek gerekir ki 0 değeri orta sonuçların ([0..1]'e) kenetlenmesinden kaçınmak için orta griye denk gelsin.

Sobel filtresinin son adımında, önceki adımlarda elde edilen kısmi türevleri ($D_x(x, y)$ ve $D_y(x, y)$) temel alarak gradyan büyüklüğün yaklaşık değeri bulunur. Sobel filtresi $S(x, y)$ 'nin sonucu olan gradyan büyüklük basitçe[12]:

$$S(x, y) = \sqrt{(D_x(x, y))^2 + (D_y(x, y))^2} \quad (3)$$

olarak bulunur.

İşleme tabi tutulacak görüntülerde ışık kaynağının da yer alması durumunda iki yönlü yerine dört yönlü [22] sobel filtresi kullanmak daha verimli sonuç vermektedir.



Şekil 3.7: Sobel Filtresi Uygulaması

3.2.4 Laplace Filtresi

Laplacian görüntünün ikinci uzaysal türevinin 2 boyutlu izotropik ölçümüdür[13]. Görüntünün Laplacian'ı ani yoğunluk değişikliklerinin olduğu

bölgeleri vurgular ve bu sebeple sıklıkla kenar belirleme için kullanılır. Laplacian genellikle görüntünün gürültüye olan hassasiyetini azaltmak için Gaussian alçak geçiren filtresine yakınsayan bir filtreyle yumuşatıldıktan sonra uygulanır. Filtre normal kullanımda tek bir gri tonlamalı bir girdi görüntüsü alır ve başka bir gri tonlamalı görüntüyü çıktı olarak verir. Piksel yoğunluk değerleri $I(x,y)$ olarak verilen bir görüntünün Laplacian $L(x,y)$ 'ı aşağıdaki gibidir.

$$L(x,y) = \frac{\delta^2 I}{\delta x^2} + \frac{\delta^2 I}{\delta y^2}$$

Görüntü üzerinde bu denklem bir konvolüsyon filtresi kullanarak hesaplanabilir. Giriş görüntüsü ayrık pikseller kümesi olarak temsil edildiğinden Laplacian tanımında ikinci türevini yaklaşık olarak alabilen ayrık konvolüsyon çekirdeği bulunmalıdır. En sık kullanılan küçük çekirdeklerden iki tanesi Şekil 3-5'te gösterilmiştir.

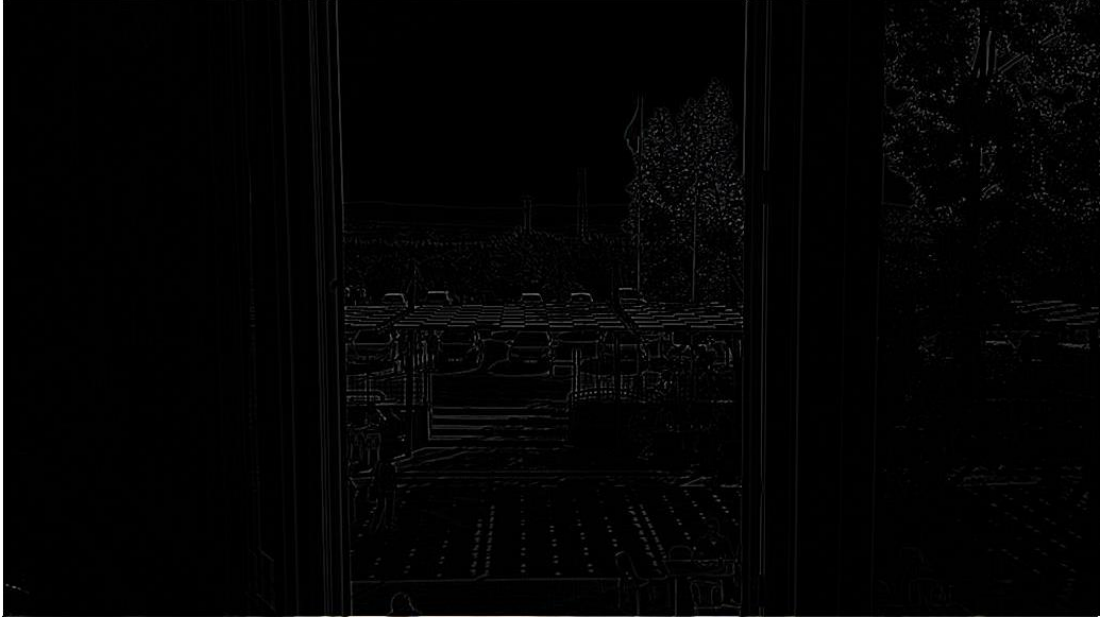
$$\begin{array}{ccc} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{array} \qquad \begin{array}{ccc} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{array}$$

Şekil 3.8: Laplace Filtresi Çekirdekleri

Burada Laplacian'ın negatif tepe değeri kullanılarak oluşturulmuş çekirdekleri daha sık kullanıldıkları için örnek olarak verilmiştir, bunun yanında zıt işaretli bir sayı düzeni kullanmak da aynı derecede geçerlidir. Bu çekirdeklerden biri kullanılarak, standart konvolüsyon metotları yoluyla Laplacian hesaplanabilir. Bu çekirdekler görüntü üzerinde ikinci dereceden türev yakınsadığı için gürültüye yüksek derecede hassastır. Buna karşı koyabilmek için görüntüye Laplacian uygulanmadan önce genellikle Gaussian filtresiyle yumuşatılır. Bu ön-işleme adımı türev adımından önce yüksek frekanslı gürültü elemanlarını azaltır [13].



Şekil 3.9: Laplace Filtresi Uygulaması



Şekil 3.10: Gaussian'ın Laplace'ı Uygulaması

3.2.5 Kabartma Filtresi

Kabartma filtresi bir çeşit kenar belirleme filtresidir [25]. Kenar belirleme genel kullanımında bir eksen boyunca pozitif ve negatif değer farkı ekleme yoluyla

çalıřır, bu yüzden iki piksel arasındaki deęer farkı ne kadar büyük olursa dönen deęer de o kadar büyük olur. Kabartma filtresinde, filtre deęerlerimiz 0 yerine 1'e eklendięi için görüntüyü aydınlatmak adına 127 ekleriz. Aksi takdirde görüntünün büyük bir kısmı siyaha gömülür[11].

$$\begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix} / 1 + 127 \quad (4)$$



Şekil 3.11: Kabartma Filtresi Uygulaması

3.3 Fabrika Uygulamaları

Günümüzde otomasyon işleri genellikle sensörler ve lazer algılayıcılar kullanılarak gerçekleştirilmektedir. Daha ileri seviye ve gelişmiş otomatik kontrol sistemleri için görüntü işleme temeline dayanan uygulamalar geliştirilmektedir. Örneğin elektrikli araba üreticisi Tesla firması kendi kendisini süren araba üretme planları dâhilinde dünyanın en büyük arama motoru şirketi olan Google'la görüşmelerde bulunmuş fakat bu aracın teknolojik olarak ışık ve mesafe sensörü kullanılarak hayata getirilmesinin maliyet açısından elverişli olmadığı ve kendi planlarının gören bir araba yapmak olduğunu açıklamıştır. Gören araba konseptiyle çevreyle iletişimini sensörlere değil kamera ve yazılım tabanlı bir sisteme dayandırmayı hedeflemişlerdir [17].

Görsel inceleme sistemi, çalışan sistemin üretim hattının esnekliğini artırmak için değil yalnızca denetimi için kullanılır. Fakat bu bile sistemin önemli ölçüde otomatikleştirilme kabiliyetini önemli ölçüde artırır. Otomatik Görsel İnceleme (OGİ) sistemi ölçüm, kalibrasyon, bütünlük ve kalite kontrolü için kullanılabilir. Ölçümler ve kalibrasyon alanlarında küçük açıklıkların kalibrasyonu, nesne boyutu ölçümü, parçaların hizalanması ve çatlak düzeni analizi sıklıkla yapılan uygulamalardır. Otomotiv fabrikaları, yemek endüstrisi ve diğer üretim hatlarında bütünlük kontrolü bu tarz bir görsel sistem kullanılarak yapılır.

Talaşlı imalat sürecinde çıkış ürünün yüksek kaliteli olması ve üreten cihazın minimum çalışmama süresi olması hedeflenir. Bunun için torna bıçağı durumu gözleme sistemleri kullanılır. Bu sürecin otomatik hale görüntü işleme yöntemiyle getirilmesi için bir inceleme çalışması yapılmıştır. Bu çalışmada doğrudan ve dolaylı torna bıçağı durumu gözleme sistemlerinin alt kolları olan iki boyutlu ve üç boyutlu teknikler ile çevrimiçi ve çevrimdışı yöntemlerin ne olduğu ve kimler tarafından araştırılıp kullanıldığı verilmiştir. Bu tablo ve araştırmalar incelendiğinde görüntü işleme ile sistem otomasyonu için oldukça çeşitli araştırma ve çalışmalar yürütüldüğü görülebilmektedir. Bu da görüntü işleme temelli otomasyon yöntemlerinin uygulanabilirliğinin ve kapsamının genişliğini açık bir şekilde ortaya koymaktadır.

Yapılmış bir uygulamada kömür ateşinin karakterizasyonu için görüntü işleme temelli bir sistem kurulmuştur. Bu sistemden elde edilen görüntüler daha sonra belli başlı belirleyici parametrelere ulaşmak için işlenmiş ve son olarak da fiziksel büyüklükler ve özel yanma durumlarıyla bağlantılar kurularak gerekli veriler toplanmıştır [26].

Parça tanıma ve sınıflandırma görüntü sisteminin en önemli uygulamalarından biridir. Otomotiv kalıplarının ve parçalarının sıralanması, parçaların belirlenmesi ve paletlerden indirilmesi önemli uygulamalardır. Yiyeceklerin ve diğer ürünlerin sıralanması ve bu tanımlama uygulamalarının örnekleridir.

Hâlihazırda yapılmış olan uygulamaların bir tanesinde mermerlerin desen şekil ve yoğunluklarına göre, görüntü işleme kullanılarak otomatik olarak önceden belirlenmiş değerlere sahip gruplara ayrıştırılması sağlanmıştır [27].

Otomasyon süreci çeşitli robotlarla temellendirilmiştir. Bu robotların birlikte çalışabilmesi için bir yön gösterim sistemine ihtiyaçları vardır. Yönlendirme ve ayarlama süreçleri ayrıca akıllı yön gösterim sistemlerine ihtiyaç duyar. Genel olarak otomotiv fabrikalarındaki robotik hareketin yön gösterimi görüntü sisteminin kaynaklama veya diğer süreçler için konum belirleme amaçlı akıllı sensör olarak kullanılır [18].

3.4 Robotik Sistem Uygulamaları

Hayatın her yönünde kullanılabilme potansiyeline sahip olan görüntü işleme temelli sistemlerin fabrikalar dışında diğer bir sıklıkla kullanıldığı alan da robotik sistemler yani robotlardır. Fabrika robot sistemleri dışında kalan çeşitli robot türleri bulunmaktadır. Bunlara örnek olarak hobi robotları (mini sumo, çizgi izleyen, labirent), biped, quadroped, eklemli robot kol vb. verilebilir. Bu robotlarında kontrolü, otomasyonu ya da çevreyle etkileşimi için görüntü işleme sistemlerine başvurmak mümkündür. Robotlar için görüntü işleme yöntemlerinden kullanılacak olanlar bu kısımda açıklanmıştır.

3.4.1 Çizgi Bulma

Tekerlekli robotların belli bir hat üzerinde servis veya ring yapabilmesi için kullanılacak en temel yöntem bir çizgiyi takip etmesidir. QRD sensörleri kullanılarak daha primitif düzeyde de uygulaması yapılabilen bu yöntem görüntü işleme sistemleriyle bir üst seviyeye kolaylıkla çıkarılabilir. Görüş açısının tamamının içindeki çizgiyi bulmasını mümkün kılan sistem sayesinde sadece anlık yolun takibini değil gelmekte olan yolun analizinin yapılmasının mümkün olması nedeniyle çok daha hızlı, verimli ve gürbüz (robust) sistemler geliştirilmesini mümkün kılar. Hobi robotlarında daha basit seviyede kullanılabilceği gibi insansız kara taşıtları için yol gösterici sistem olarak kullanılması da mümkündür.

Yol takibinden farklı bir kullanım amacı ise sınır belirlemedir. Robotun çalışacağı alanın sınırlarını belirlemek için veya önceden öğretilmiş nesne kodlarını tanıması için çizgi bulma kullanılabilir.

3.4.2 Orta Nokta Bulma

Robotun etkileşim içinde olduğu nesnenin taşıma veya denge amaçlı olarak orta noktasını görüntü üzerinden tespit etmesi istikrarlı bir deneyim için önemlidir. Farklı bir kullanım amacı olarak nesne tanıma yöntemlerinin uygulanabilmesi, önceden veri tabanında olan tanımlanmış görüntülerle karşılaştırılabilmesi için öncelikle nesnenin orta noktasının bulunması gerekmektedir. Nesne tanımanın bir sonraki adımı olan tanımlanamayan, önceden bilinmeyen, veri tabanına kaydedilmemiş nesnelerin düzgün ve robust biçimde öğrenilebilmesi için de yine ilk olarak orta noktasının bulunması gerekmektedir.

3.4.3 Renk Belirleme

Robot üzerine entegre edilmiş kamera sisteminden 3 renk kanallı gelen görüntülerin belli renklerinin filtre edilmesiyle yani sadece tek bir renk kanalının işlenip diğerlerinin göz ardı edilmesiyle istenen renk izole edilip aranan renge

ulařılabilir. Yol bulma veya haritalama iřlemleri iin kullanılabilir olan bu yntem ile belli zellikleri bilinen aynı nesnelere ayırt etme iřlemi de yapılabilir.

3.4.4 Renk Takip Etme

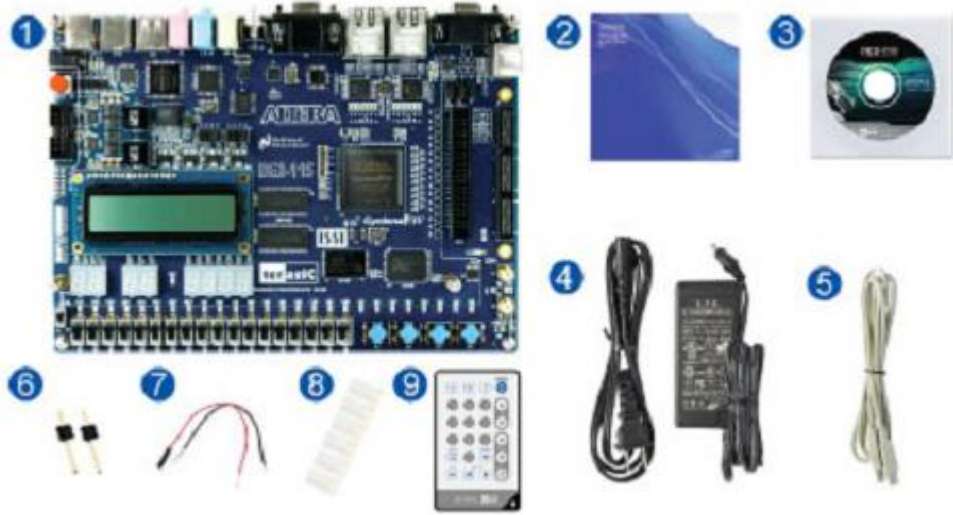
nceden kodlanmış ya da hareket halindeyken (on-the-go) ğrenilmiş rengin dolayısıyla cismin takibi iin oluşturulmuş kamera dzeneęi zerinden gelen grntde filtreleme ile renk belirleme yapılır. Bu belirleme kaydedildikten sonra gerek zamanlı olarak robotun bulunduęu ortam taranır ve kayıtlı olan renk grldęu yere robot ynlendirilerek takip etme gerekleřtirilir.

4. DONANIM

Araştırılan literatürde FPGA üzerinde yapılan görüntü işleme uygulamalarında ağırlıklı olarak Xilinx FPGA'lerle çalışıldığı görülmüştür [5,7,15,16,21]. Yapılan uygulamada literatürdeki FPGA'lerle benzer özelliklere sahip olan Altera firmasının DE2-115 eğitim boardu ve bu boardla uyumlu kamera kiti Terasic marka TRDB_D5M kullanılmıştır.

4.1 Altera DE2-115 Eğitim Boardu

Bu çalışmada kullanılan platform olan Altera DE2-115'in paket içeriği, Şekil 4-1'de Tablo 4-1'de verilmiştir.



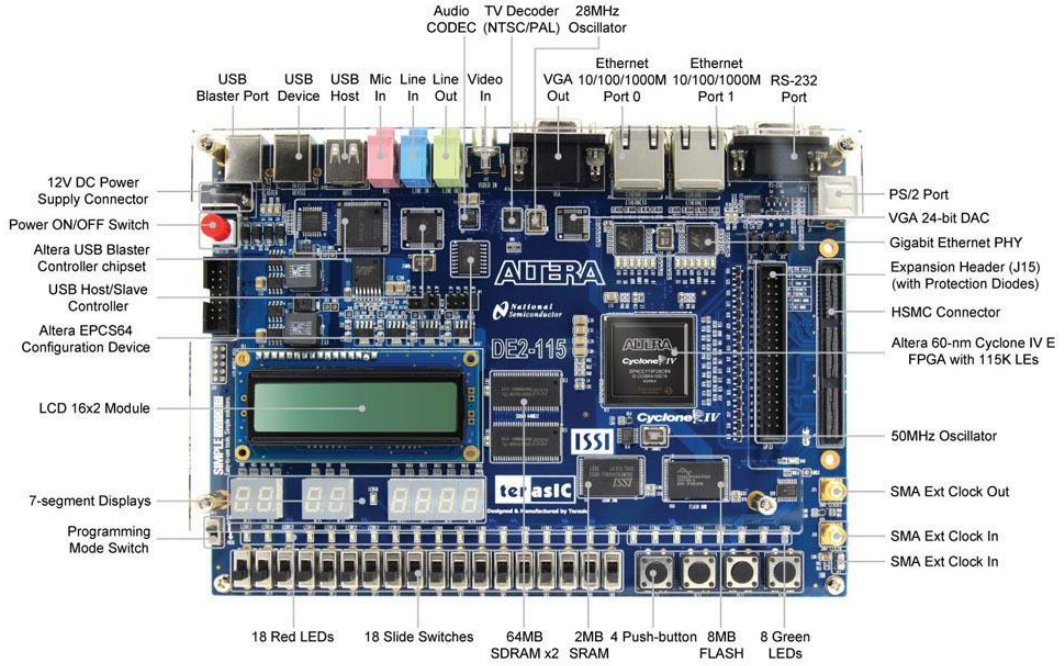
Şekil 4.1: DE2-115 Paket İçeriği

Tablo 4.1: Paket İçeriği

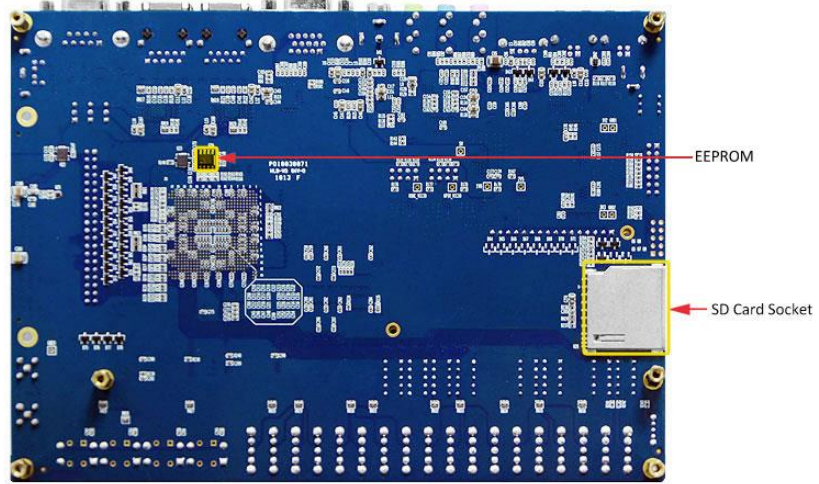
Sıra	İçerik
1	Altera DE2-115 Board
2	Windows için Altera Complete Design Suite DVD'si
3	DE2-115 Sistem CD'si
4	12V/2A'lık DC Güç Adaptörü
5	USB Programlama Kablosu
6	2 Adet 1 pinlik header
7	2 Adet Kablo Şeridi
8	6 Silikon Ayaklık
9	Uzaktan Kumanda

4.1.1 Yerleşim ve Elemanlar

Şekil 4-2 ve 4-3'de DE2-115'in üstten ve alttan görünüşleri bulunmaktadır. Belli başlı elemanların yerleşimleri ve bağlantı noktaları görülmektedir.



Şekil 4.2: DE2-115 Board'u üstten görünüm



Şekil 4.3: DE2-115 Board'u alttan görünüm

DE2-115 board'unda kullanıcıların basit devrelerden çeşitli çoklu ortam uygulamalarına kadar çok farklı devre tasarımlarını board üzerinde uygulamasına olanak sağlayacak çeşitlilikte özelliklere sahiptir. Şekillerde verilen donanımlar aşağıdaki gibidir [20].

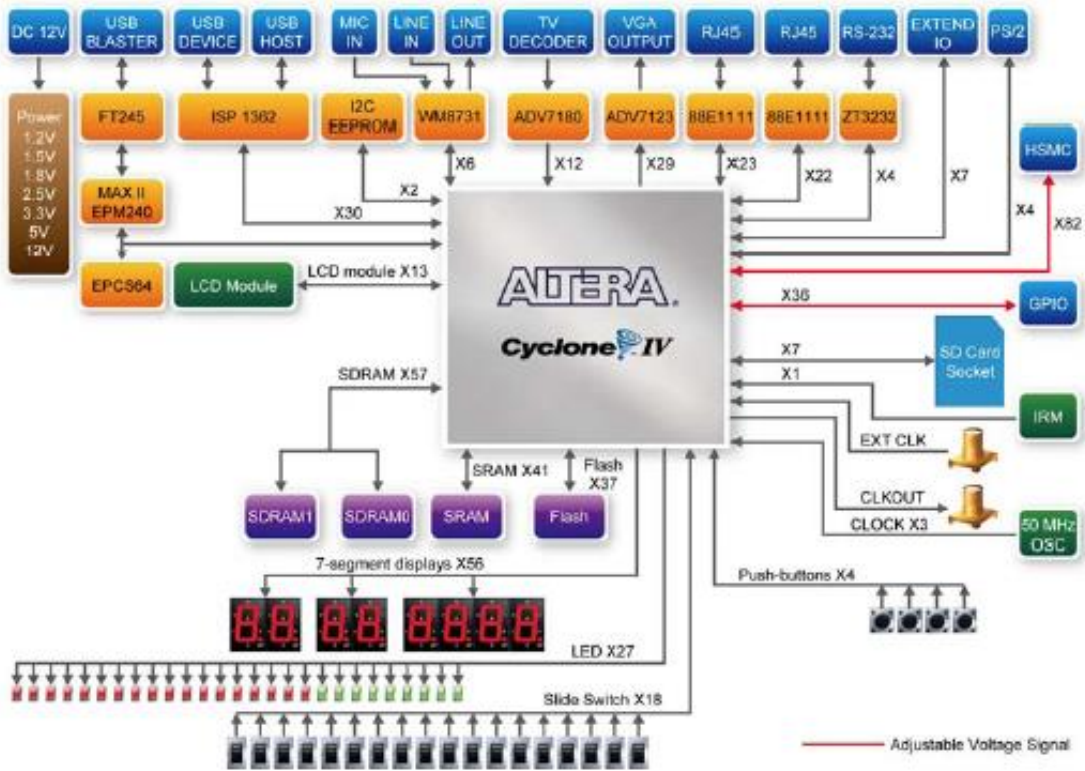
- Altera Cyclone® IV 4CE115 FPGA cihazı
- Altera Seri Yapılandırma Cihazı– EPCS64
- Programlama için USB Blaster (board üstünde), JTAG ve Active Serial (AS) modlarının ikisi de desteklenmektedir
- 2MB SRAM
- 2 adet 64MB SDRAM
- 8MB Flash bellek
- SD Kart Yuvası
- 4 Adet basılabilir düğme
- 18 Adet kaydırılabilir anahtar
- 18 Adet kırmızı kullanıcı LED'i
- 9 Adet yeşil kullanıcı LED'i
- Saat kaynakları için 50MHz osilatör

- Line giriş ve çıkışı ve mikrofon girişi jaklarıyla birlikte 24-bit CD kalitesinde ses CODEC'i
- VGA çıkış bağlantısıyla birlikte VGA Dijital Analog Çevirici (8 bitlik yüksek hızlı, üçlü Dijital Analog Çevirici)
- TV Dekoderi (NTSC/PAL/SECAM) ve TV giriş bağlantısı
- RJ45 bağlantılı 2 Gigabit Ethernet PHY
- USB A ve B tipi konektörleriyle birlikte USB Host/Slave Kontrolörü
- RS-232 alıcı vericisi ve 9-pinlik bağlantısı
- PS/2 fare/klavye konektörü
- Kızılötesi Alıcı
- Harici saat giriş/çıkışı için 2 adet SMA konektörü
- Diyot korumalı bir adet 40-pinlik genişletme yuvası
- Bir adet high speed mezzanine card (HSMC) konektörü
- 16x2'lik LCD modülü

Bu donanım özelliklerinin yanı sıra DE2-115 board'unun standart giriş çıkış ara yüzleri için yazılım desteği ve çeşitli elemanlara erişim için denetim masası olanağı vardır. Quartus II yazılımı yardımıyla Verilog, VHDL veya şematik tasarım yöntemlerinden birini kullanarak sistem tasarımları gerçekleştirilebilir.

4.1.2 DE2-115 Board'unun Blok Diyagramı

Şekil 4-4'de DE2-115 board'unun blok diyagramı yer almaktadır. Şekilde görülebileceği gibi kullanıcıya maksimum esnekliği sağlamak için bütün bağlantılar Cyclone IV E FPGA cihazı üzerinden yapılmıştır. Böylece kullanıcı FPGA'yi ayarlayarak her hangi bir sistem tasarımı uygulamasını yapabilir.



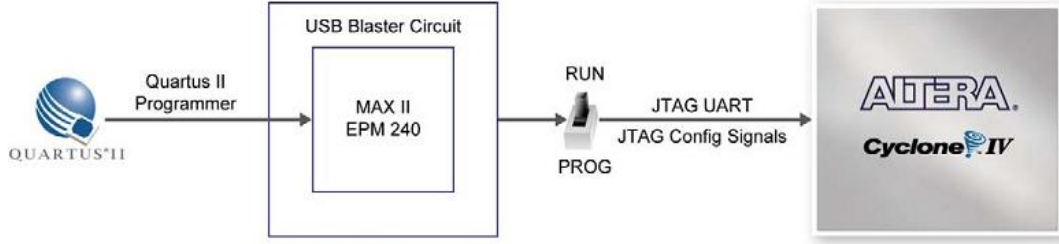
Şekil 4.4: DE2-115 Board'unun blok diyagramı

Şekil 4-4'teki blokların detaylı açıklaması EK-B kısmında verilmiştir.

4.1.3 FPGA'in JTAG modunda ayarlanması

Şekil 4-5'te görülen JTAG ayarlama düzeneği kurulduktan sonra aşağıdaki işlemler yapılmalıdır.

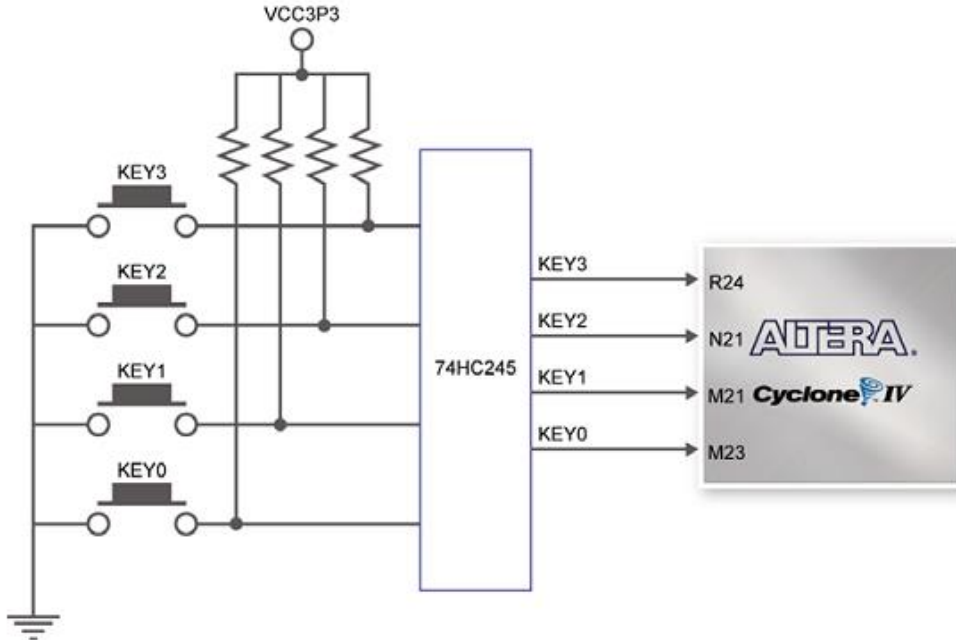
- DE2-115 Board'unun gücü açılır
- RUN/PROG anahtarını RUN konumuna getirerek JTAG programlama devresini ayarlanır
- USB kablosuyla USB Blaster portuna bağlantı yapılır.
- Quartus II programı kullanılarak .sof dosya uzantılı bit akışı dosyası seçilip FPGA programlanabilir.



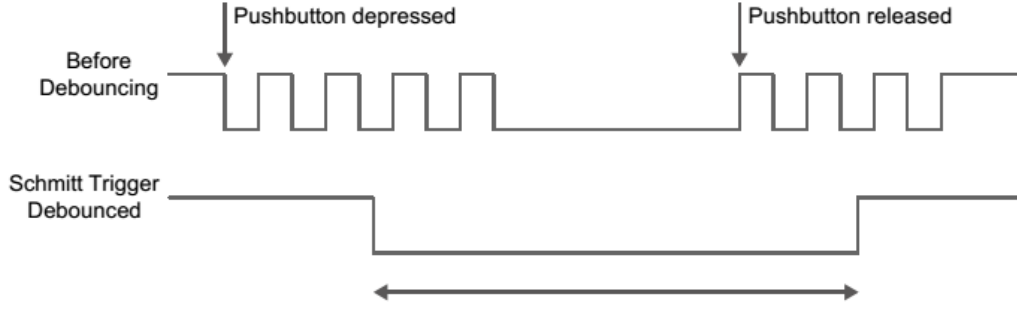
Şekil 4.5: JTAG Ayarlama Şeması

4.1.4 Basılan Düğmelerin ve Anahtarların Kullanımı

DE2-115 board'unda 4 adet basılan düğme bulunur (şekil 4-6). Her bir anahtar Schmitt Trigger devresi kullanılarak debounce edilmiştir (şekil 4-7). KEY0, KEY1, KEY2, KEY3 olarak adlandırılan Schmitt Trigger cihazlarının dört çıkışı doğrudan Cyclone IV E FPGA'e bağlıdır. Her bir düğme basılı değilken yüksek mantık seviyesindedir; ve basılı duruma geldiğinde düşük mantık seviyesini sağlar. Düğmeler debounce edilmiş olduğu için devrede saat veya sıfırlama girişi olarak kullanmaya uygundur.

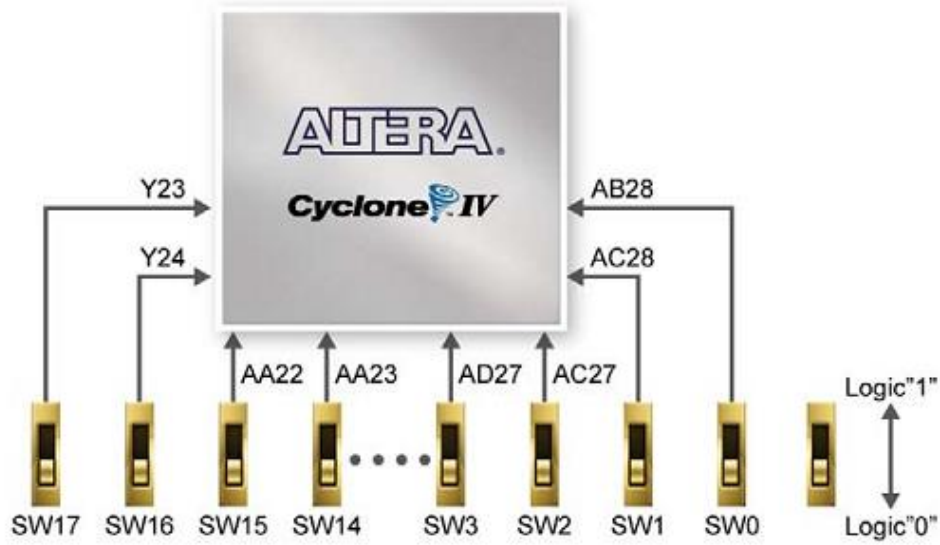


Şekil 4.6: Düğmeler ve Cyclone IV E FPGA arasındaki bağlantılar



Şekil 4.7: Anahtarların Debounce Edilmesi

DE2-115 board’unda ayrıca 18 kaydırılabilir anahtar bulunur (Şekil 4-8). Bu anahtarlar debounce edilmemiştir ve devreye seviye hassasiyetli veri girişi için kullanılacağı varsayılmıştır. Her anahtar Cyclone IV E FPGA üzerindeki bir pine bağlıdır. Anahtar aşağı konumundayken FPGA’ye düşük mantık seviyesi sağlar, yukarı konumda ise yüksek mantık seviyesi sağlar.



Şekil 4.8: Kaydırılabilir Anahtarlar ve Cyclone IV E FPGA arasındaki bağlantılar

4.1.5 LED’lerin Kullanımı

DE2-115 board’unda 27 adet kullanıcının kontrol edebildiği LED bulunmaktadır. 18 kırmızı LED 18 kaydırılabilir anahtarın üzerinde konumlanmıştır. 8 yeşil LED de basılabilen düğmelerin üzerinde yer almaktadır (9. yeşil LED 7 parçalı göstergelerin ortasında bulunmaktadır). Her LED doğrudan Cyclone IV E

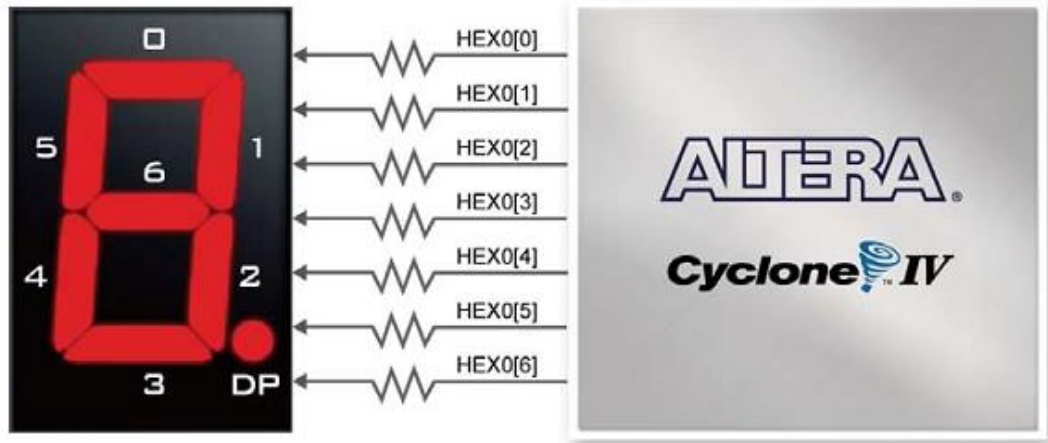
FPGA üzerindeki bir pin tarafından sürülmektedir, ilişkili pinin yüksek mantık seviyesine sürülmesi LED'i yanar konuma getirir, düşük seviyeye sürmek ise söndürür. Şekil 4-9'da LED'ler ve Cyclone IV E FPGA arasındaki bağlantılar görülmektedir.



Şekil 4.9: LED ve Cyclone IV E FPGA arasındaki bağlantılar

4.1.6 7 Parçalı Göstergelerin Kullanımı

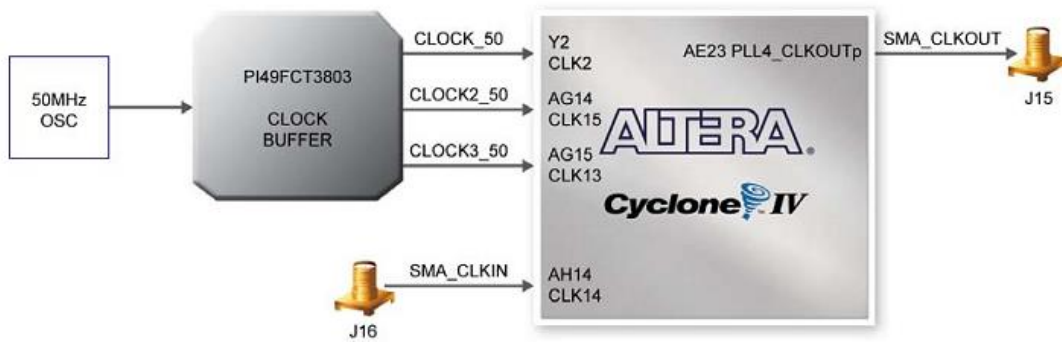
DE2-115 Board'unda sekiz adet 7 parçalı gösterge bulunur. Bu göstergeler çeşitli büyüklüklerdeki sayıları gösterebilmek amacıyla iki çift ve dörtlü bir grup olarak düzenlenmiştir. Şekil 4-10'da görüldüğü gibi yedi parça (ortak anotlu) Cyclone IV E FPGA'deki pinlere bağlıdır. Herhangi bir parçaya düşük mantık seviyesi uygulandığında o parça yanar, yüksek mantık seviyesi uygulandığında ise söner.



Şekil 4.10: 7 Parçalı Gösterge HEX0 ile Cyclone IV E FPGA arasındaki bağlantılar

4.1.7 Saat Devresi

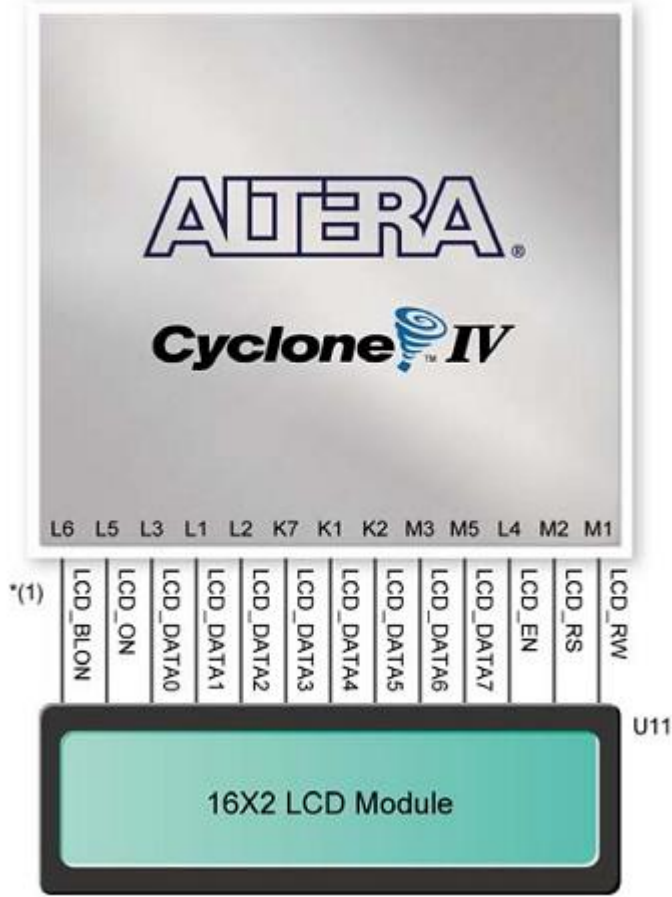
DE2-115 board'unda 50 MHz saat sinyali üreten bir adet osilatör bulunur. 50 MHz'lik saat sinyalini düşük titreşimle FPGA'ye dağıtmak için saat tamponu kullanılır. Dağıtılan saat sinyalleri FPGA'ye kullanıcı mantık devrelerinin saatini ayarlamak için bağlanır. Board'da harici saat kaynağı bağlamak veya saat çıkış sinyalini sürebilmek için iki adet SMA konektörü bulunur. Ek olarak bütün saat girişleri kullanıcıların bu saatleri PLL (phase locked loop) devrelerine kaynak saat olarak kullanabilmeleri için FPGA'in PLL saat giriş pinlerine bağlanır. DE2-115 board'undaki saat dağıtımını Şekil 4-11'de görülmektedir.



Şekil 4.11: Saat Dağılımı Blok Diyagramı

4.1.8 LCD Modülünün Kullanımı

LCD modülünün kendi içinde yazı karakteri bulunmaktadır. Uygun komutların HD44780 adlı gösterge kontrolörüne gönderilmesiyle yazı gösterimi için kullanılabilir. LCD modülünün Cyclone IV E FPGA'e olan bağlantılarının şematik diyagramı Şekil 4-12'de görülmektedir.

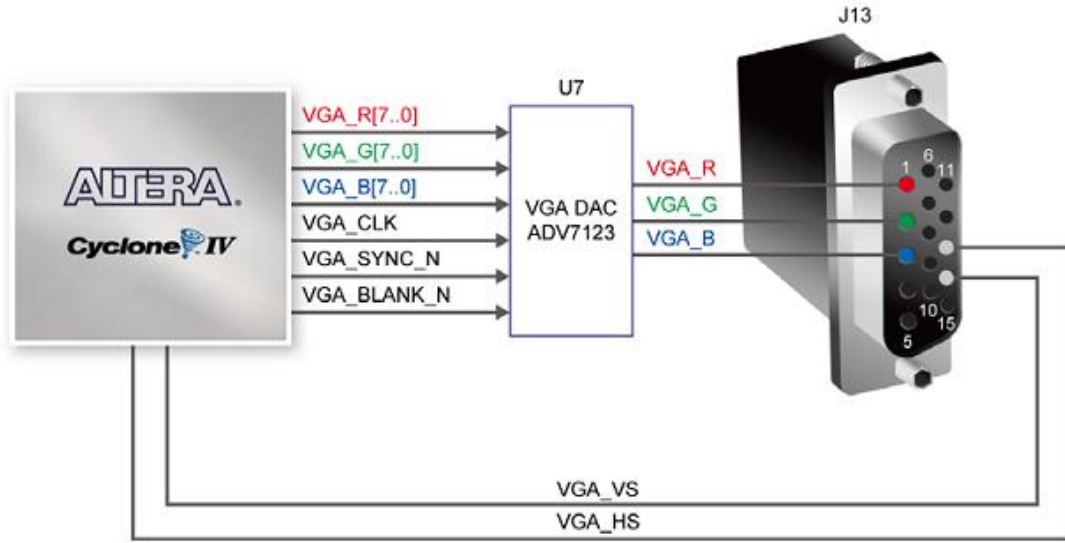


Şekil 4.12: LCD Modülü ile Cyclone IV E PGA arasındaki bağlantılar

4.1.9 VGA'nın Kullanımı

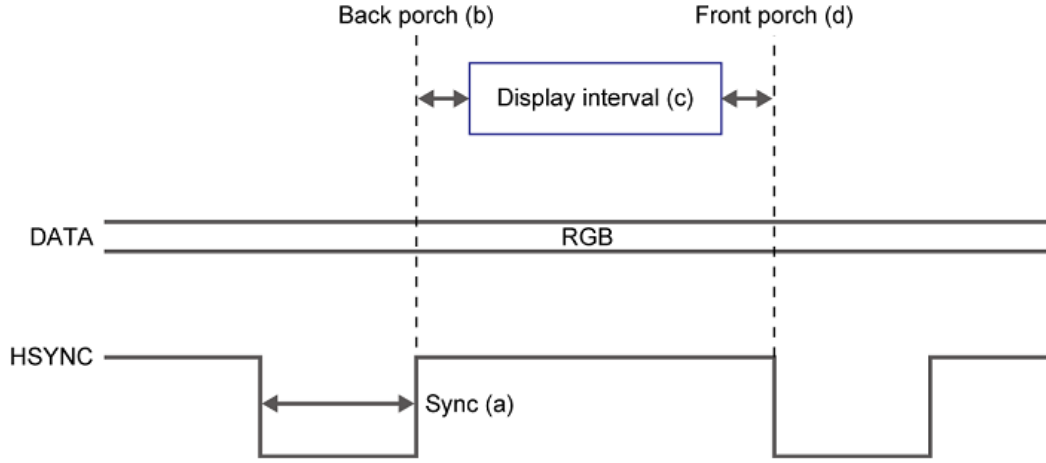
DE2-115 board'unda VGA çıkışı için 15 pinlik bir D-SUB konektörü bulunur. VGA senkronizasyon sinyalleri doğrudan Cyclone IV E FPGA tarafından sağlanır ve analog veri sinyallerini (kırmızı, yeşil ve mavi) sağlamak için Analog Devices ADV7123 üçlü 10 bit yüksek hızlı video dijital analog çevirici (sadece en

yüksek 8 bit kullanılır) kullanılır. 100 MHz bant genişliğinde SXGA standart (1280*1024) çözünürlüğünü destekler. İlişkili şematik Şekil 4-13'de görünmektedir.



Şekil 4.13: FPGA ve VGA Arasındaki Bağlantılar

Şekil 4-14'de VGA ekranda gösterilen her satır (yatay) için temel zaman gereksinimleri gösterilmiştir. Belirli bir süre boyunca bir veri satırının bitip diğerinin başlayacağını belirten aktif-düşük bir darbe sinyali (şekilde (a) süresi) ekranın yatay senkronizasyon (hsync) girişine uygulanır. Ekrana veri (RGB) çıkışı hsync darbesi oluşturulduktan sonra back porch (b) denen süre boyunca kapalı (0 V'a sürülmüş) olmalıdır. Bunu da gösterim aralığı (c) takip eder. Veri gösterim aralığı boyunca RGB verisi görüntülenmekte olan satır boyunca her pikseli sırayla sürer. Son olarak RGB sinyallerinin bir sonraki hsync darbesi oluşmadan önce tekrar kapalı hale gelmesinin gerektiği front porch denen bir zaman aralığı vardır. Dikey senkronizasyon (vsync) da vsync darbesinin bir karenin sonu ve diğerinin başını belirtmesi ve verinin karedeki satırlar kümesine (yatay zamanlama) karşılık gelmesi haricinde Şekil 4-14'deki gibidir. Tablo 4-2 ve 4-3 a, b, c, d zaman aralıklarının farklı çözünürlük ve sürelerini yatay ve dikey zamanlama için gösterir.



Şekil 4.14: VGA Yatay Zamanlama Tanımlaması

Tablo 4.2: VGA Yatay Zamanlama Tanımlaması

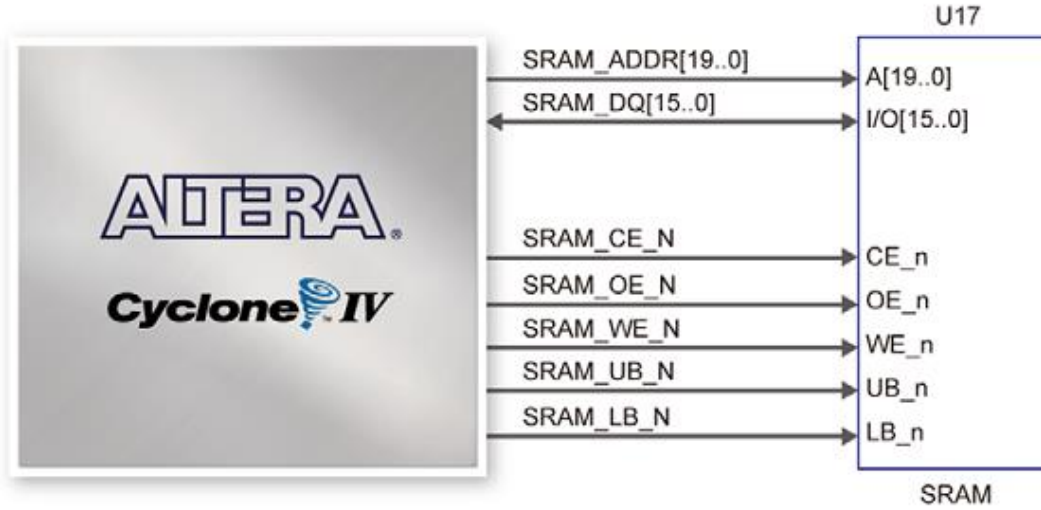
VGA Modu		Yatay Zaman Tanımlaması				
Ayar	Çözünürlük (YxD)	A(us)	B(us)	C(us)	D(us)	Piksel Saati (MHz)
VGA (60Hz)	640x480	3.8	1.9	25.4	0.6	25
VGA(85Hz)	640x480	1.6	2.2	17.8	1.6	36
SVGA(60Hz)	800x600	3.2	2.2	20	1	40
SVGA(75Hz)	800x600	1.6	3.2	16.2	0.3	49
SVGA(85Hz)	800x600	1.1	2.7	14.2	0.6	56
XGA(60Hz)	1024x768	2.1	2.5	15.8	0.4	65
XGA(75Hz)	1024x768	1.8	1.9	13.7	0.3	75
XGA(85Hz)	1024x768	1.0	2.2	10.8	0.5	95
1280x1024(60Hz)	1280x1024	1.0	2.3	11.9	0.4	108

Tablo 4.3: VGA Dikey Zaman Tanımlaması

VGA Modu		Dikey Zaman Tanımlaması				
Ayar	Çözünürlük (YxD)	a(satır)	b(satır)	c(satır)	d(satır)	Piksel Saati (MHz)
VGA (60Hz)	640x480	2	33	480	10	25
VGA(85Hz)	640x480	3	25	480	1	36
SVGA(60Hz)	800x600	4	23	600	1	40
SVGA(75Hz)	800x600	3	21	600	1	49
SVGA(85Hz)	800x600	3	27	600	1	56
XGA(60Hz)	1024x768	6	29	768	3	65
XGA(75Hz)	1024x768	6	29	768	3	75
XGA(85Hz)	1024x768	3	36	768	1	95
1280x1024(60Hz)	1280x1024	3	38	1024	1	108

4.1.10 SRAM

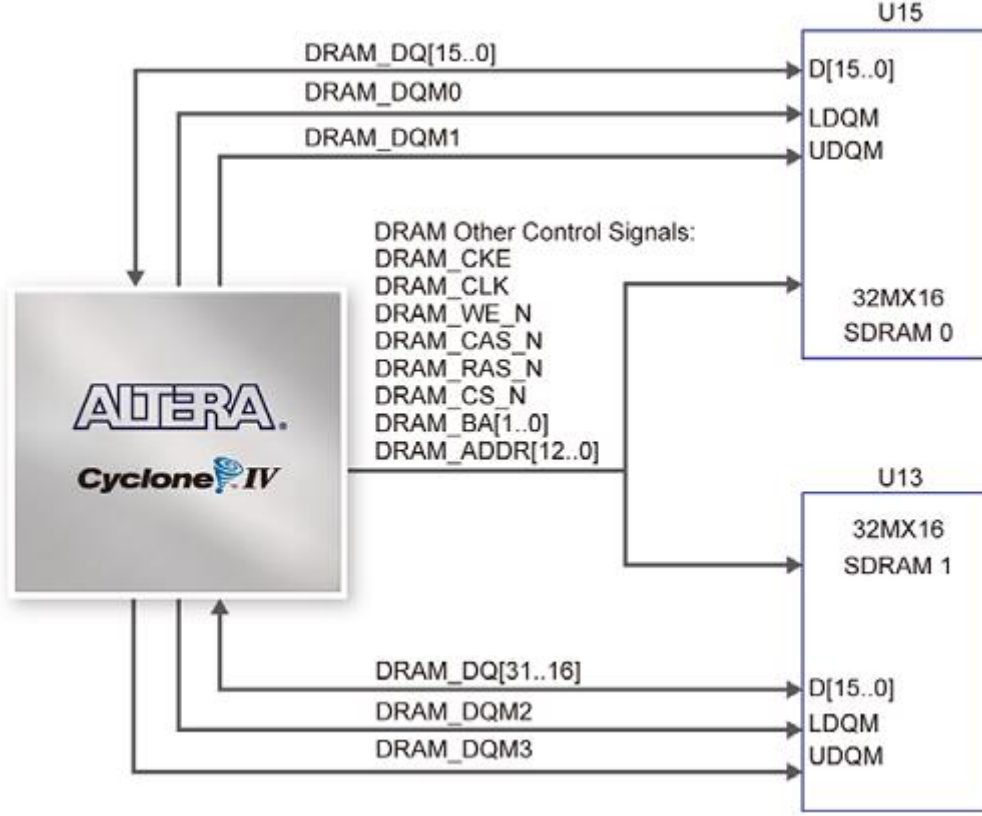
DE2-115 board'unda 16 bit veri genişliği olan 2 MB SRAM hafıza bulunmaktadır. Standart 3.3 V tekil güç kaynağı beslemesi altında en yüksek performans frekansı olan 125 MHz'e sahip olması nedeniyle çok yüksek veri çıktısına ihtiyaç duyan yüksek hızlı ortam işleme uygulamaları için uygun olup Şekil 4-15'te gösterilmiştir.



Şekil 4.15: FPGA ve SRAM Arasındaki Bağlantılar

4.1.11 SDRAM

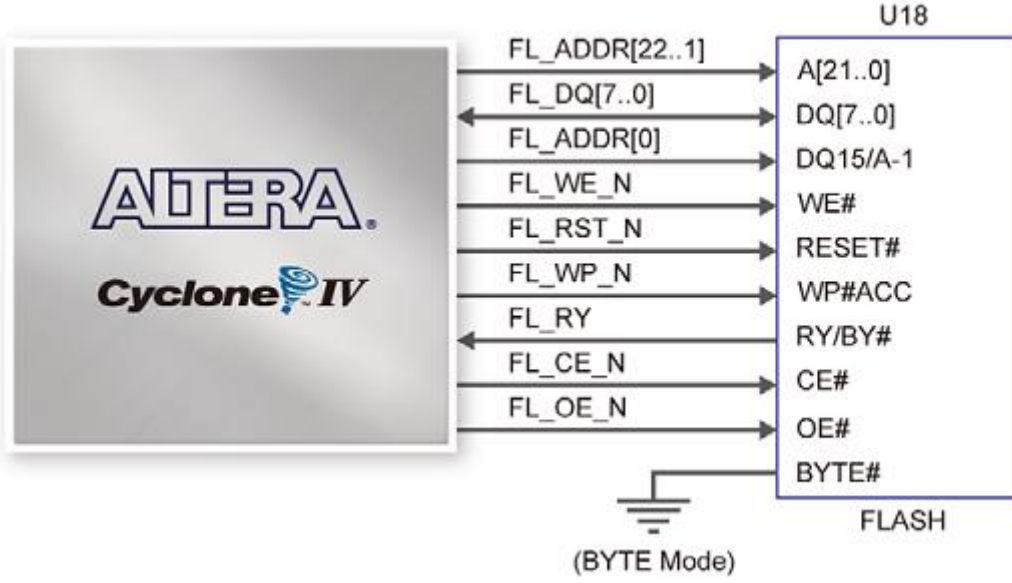
Board üzerinde iki adet 64 MB'lık SDRAM'den oluşan 128 MB SDRAM hafızası bulunmaktadır. Her bir cihaz FPGA'ye bağlı ayrı 16 bitlik veri yollarına, ortak kontrol ve adres yollarına sahiptir. Bu yongalar 3.3 V LVCMOS sinyal standartlarını kullanır. FPGA ve SDRAM arasındaki bağlantılar Şekil 4-16'da verilmiştir.



Şekil 4.16: FPGA ve SDRAM Arasındaki Bağlantılar

4.1.12 FLASH BELLEK

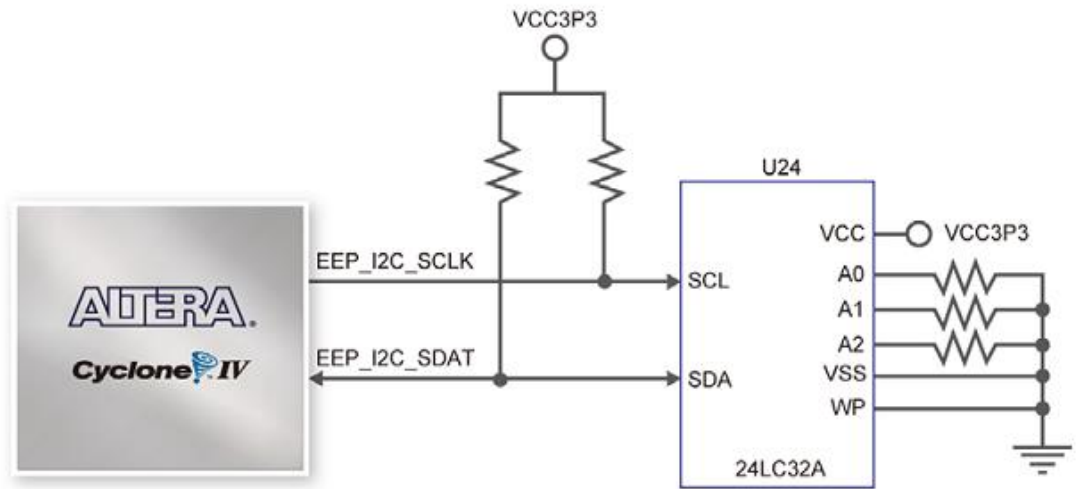
Board üzerine 8 bit veri yoluna sahip 8 MB'lık flash hafıza monte edilmiş durumdadır. Cihaz 3.3V CMOS sinyal standartını kullanmaktadır. Güç kesildiği zaman silinmeyen yapısı nedeniyle genellikle yazılım kaynak dosyalarını, görüntüleri, sesleri veya diğer ortam dosyalarını depolamak için kullanılır. FPGA ve Flash arasındaki bağlantılar şekil 4-17'de gösterilmiştir.



Şekil 4.17: FPGA ve Flash Arasındaki Bağlantılar

4.1.13 EEPROM

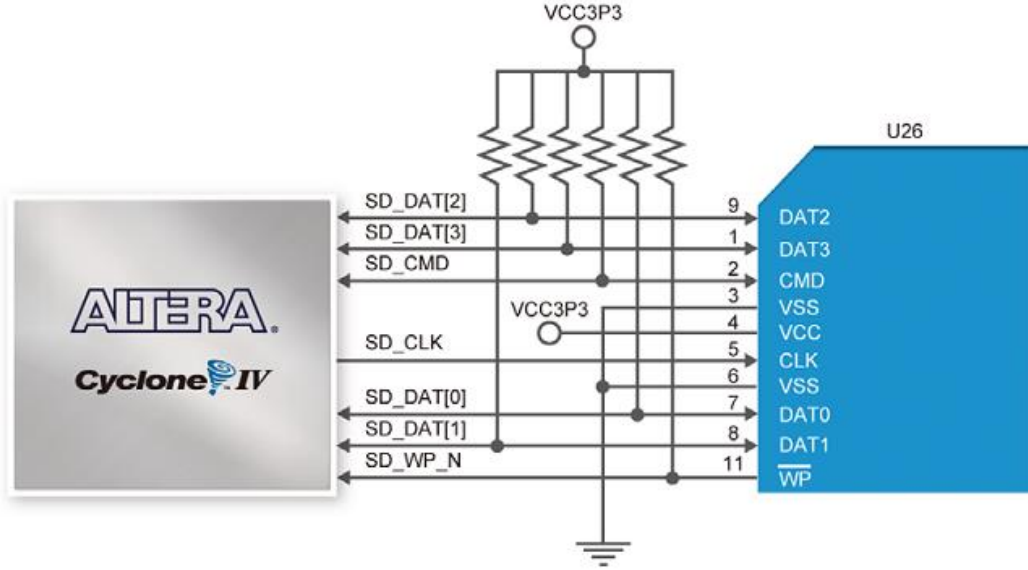
Board üzerinde 32 KB'lık EEPROM bulunmaktadır. I2C yolunun faydasıyla kullanıcı bu hafızayı sürüm bilgisi, MAC adresi veya diğer tanımlama bilgileri gibi kullanıcı verileri için kullanabilir. Şekil 4-18 EEPROM'un şematik görüntüsünü vermektedir. EEPROM'un ayarlanmış erişim adresi yazım için 0xA0, okuma için ise 0xA1'dir.



Şekil 4.18: FPGA ve EEPROM arasındaki bağlantılar

4.1.14 SD Kart

Bir çok uygulama veri depolama için SD Kart veya CF kart gibi büyük harici depolama cihazları kullanır. DE2-115 board'u SD Kart erişimi için ihtiyaç duyulan donanımı sağlamaktadır. Kullanıcılar SD Karta SPI modunda ve 4 bit veya 1 bit SD Kart modunda erişmek için kendi yaptıkları kontrolörleri uygulayabilir. Şekil 4-19 alakalı sinyalleri göstermektedir [19].



Şekil 4.19: FPGA ve SD Kart Yuvası Arasındaki Bağlantılar

4.2 Terasic TRDB_D5M 5 MP Digital Camera Development Kit

Terasic firması tarafından üretilen TRDB_D5M 5 Mega Piksellik, Altera DE4/DE2-115/DE2/70/DE2 ve DE1 board'larında kullanılabilen bir kamera geliştirme kitidir. Kit içeriği olarak

- Tek CMOS sensörüne sahip TRDB_D5M board'u.
- Tasarım için referans CD'si.

bulunmaktadır.

Şekil 4-20'de TRDB-D5M kamera kiti görülmektedir.



Şekil 4.20: TRDB-D5M Kamera Kiti

Kameranın genel özellikleri aşağıda belirtildiği şekildedir.

- Yüksek saniye başına kare sayısı
- Üstün düşük ışık performansı
- Düşük karanlık akımı
- Global sıfırlama, bütün satırların ışık almasını aynı anda başlatır
- Ampul ışık alma modu, rasgele ışık alma zamanları için
- İstek üzerine resim karesini kaydetmek için snapshot modu
- Yatay ve dikey görüntü aynalama
- Görüş açısını daraltmadan görüntü boyutunu küçültmek için satır ve sütun atlama modları
- Yeniden boyutlandırma sırasında görüntü kalitesini artırmak için satır ve sütun “binning” modu

- Sade iki-kablo seri arayüz
- Programlanabilir kontroller; kazanç, kare sayısı, kare boyutu, ışık alımı
- Otomatik siyah seviyesi ayarı
- Yonga üstünde PLL

Tablo 4.4: Anahtar Performans Parametreleri[20]

Parametre		Değer
Aktif Piksel		2592Y x 1944D
Piksel Boyutu		2.2 μm x 2.2 μm
Renk Filtre Dizisi		RGB Bayer Şablonu
Deklanşör Tipi		Global Sıfırlama Serbestisi
Maksimum veri hızı/ana saat		96 MHz'de 96 Mp/s
Kare Hızı	Tam Çözünürlük	15 fps'ye kadar programlanabilir
	VGA(640x480)	70 fps'ye kadar programlanabilir
ADC Çözünürlüğü		12 bit
Tepki Verme		1.4 V/lux-sn (550nm)
Piksel Dinamik Menzil		70.1 dB
SNRMAX		38.1 dB
Gerilim	Güç	3.3 V
	Giriş/Çıkış	1.7 V ~ 3.1 V

4.3 QUARTUS II System Builder

DE2-115 sistem kurucusu Windows tabanlı, kullanıcılara kısa sürede DE2-115 board'una Quartus II projesi üretmelerinde yardımcı olmak için kullanılan bir programdır. Üretilen Quartus II projesinde:

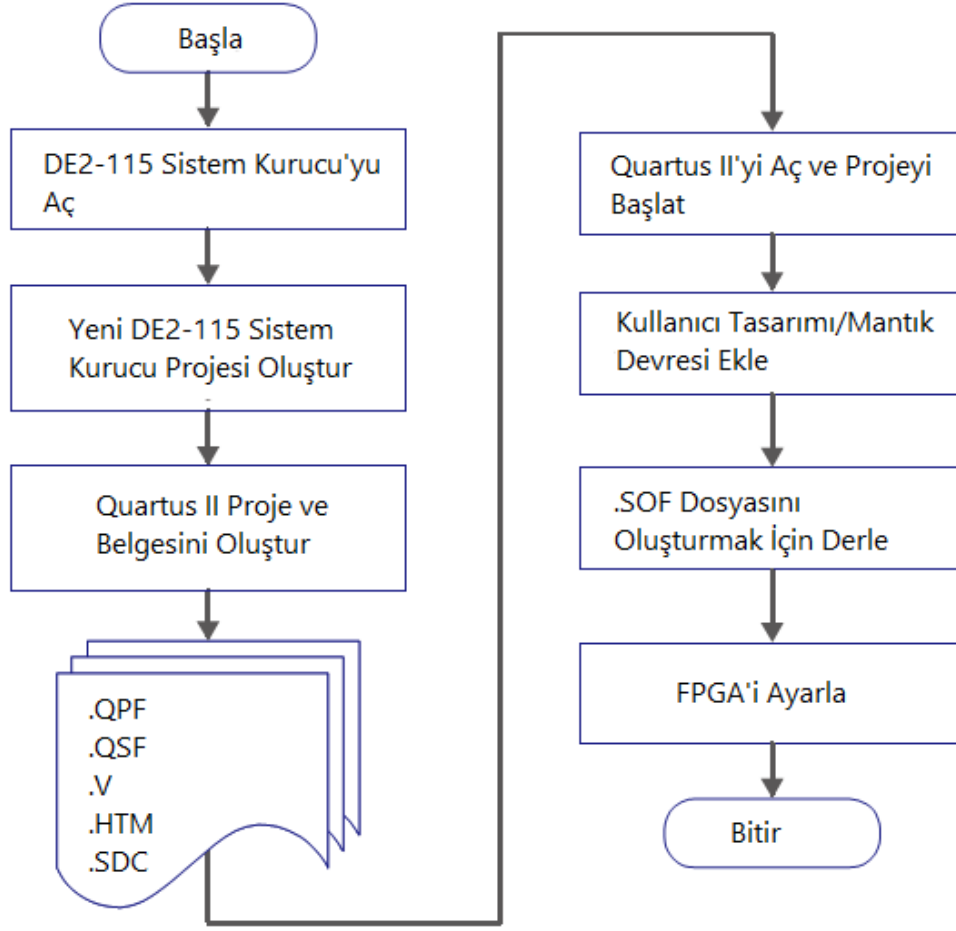
- Quartus II Proje Dosyası (.qpf)
- Quartus II Ayar Dosyası (.qsf)
- Üst Seviye Tasarım Dosyası (.v)
- Özet Tasarım Kısıt Dosyası (.sdc)
- Pin Atama Belgesi (.htm)

bulunur.

Üstteki dosyaları temin ederek DE2-115 sistem kurucu, kullanıcıların elle düzenledikleri üst seviye tasarım dosyalarında veya pin atamalarında oluşabilecek hataların önüne geçer.

4.3.1 Tasarım Akışı

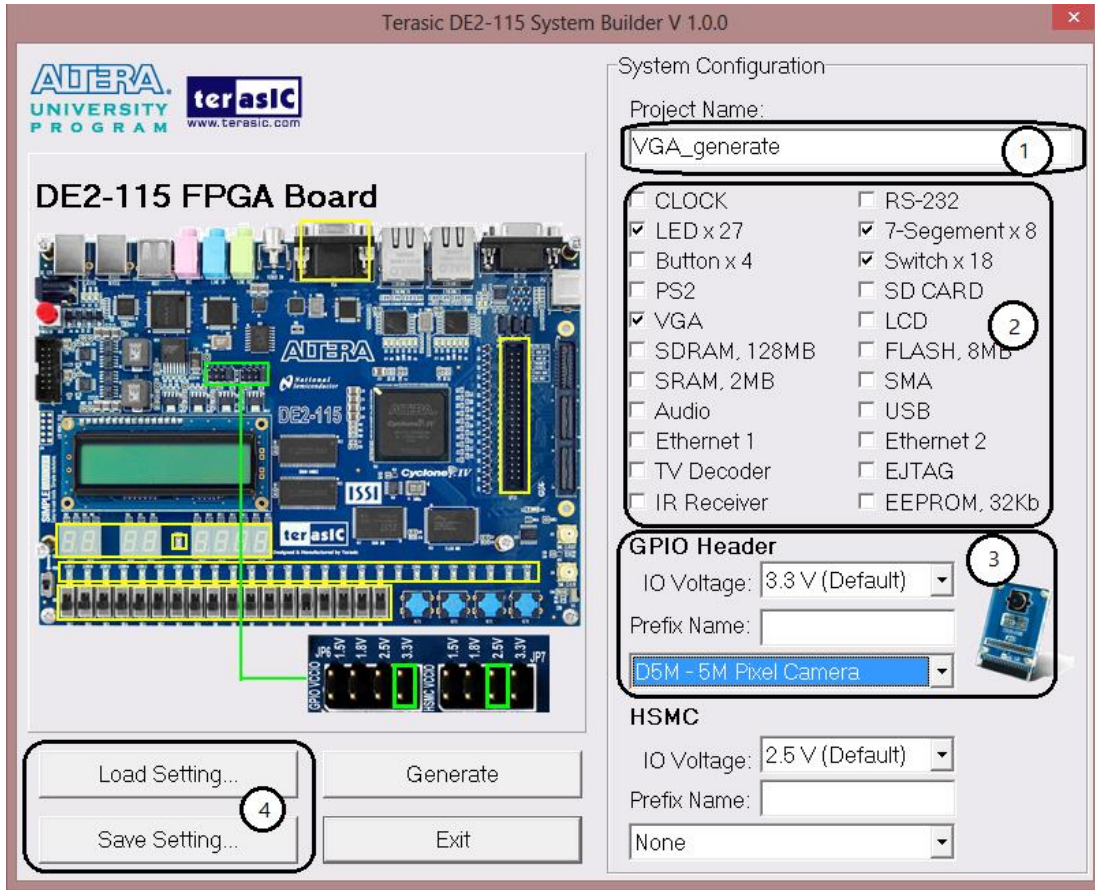
Tasarım akışı Şekil 4-21'de görülmektedir. Program açılıp tasarımın ihtiyaç duyduğu elemanlar doğrultusunda yeni bir proje yaratılır. Ayarlar tamamlandığında DE2-115 sistem kurucu üst seviye tasarım dosyası (.v) ve Quartus II ayar dosyasını (.qsf) içeren iki ana dosya üretir. Üst seviye tasarım dosyası kullanıcıların kendi tasarımlarını/mantıksal devrelerini ekleyebilmesi için verilog HDL paketini içerir. Quartus II ayar dosyası FPGA cihaz çeşidi, üst seviye pin atamaları ve kullanıcı tarafından belirlenmiş her giriş çıkış için giriş çıkış standardı hakkında bilgi içerir. Son olarak Quartus II programlayıcısı üzerinden JTAG ara yüzü kullanılarak SOF dosyası DE2-115 board'una yüklenir.



Şekil 4.21: Bir Tasarım Üretiminin Genel Akış Şeması

4.3.2 DE2-115 Sistem Kurucu Kullanımı

DE2-115 sistem kurucu, sistem CD'si içinde yer alan DE2_115_system_builder klasöründen doğrudan çalıştırılır. Sistem kurucunun kullanıcı ara yüzü Şekil 4-22'de görülmektedir.



Şekil 4.22: DE2-115 Sistem Kurucu Kullanıcı Ara Yüzü

Şekil 4-22’de “1” ile gösterilen yere, otomatik bir şekilde üst-seviye tasarımın adı olarak atanacak olan proje ismi yazılır.

Sistem ayarlama (system configuration) bölümü altında DE2-115’de kullanılmak istenen elemanları seçme olanağı sağlanmaktadır (Şekil 4-22’de “2” ile gösterilen kısım). DE2-115’in her elemanı basit bir tikleme işlemiyle sisteme eklenebilir veya sistemden çıkarılabilir. Eğer eleman devreye sokulmuşsa DE2-115 sistem kurucu otomatik olarak alakalı pin isimleri, pin konumu, pin yönü ve giriş çıkış standardı dâhil pin atamalarını yapar.

DE2-115 boardundaki GPIO bağlantı noktasına GPIO genişletme kartları Şekil 4-22’de “3” ile gösterildiği üzere bağlanabilir. Açılır menüden tasarımda kullanılmak istenen ek kart seçilir. Sistem kurucu otomatik olarak alakalı pin isimleri, pin konumu, pin yönü ve giriş çıkış standardı dahil pin atamalarını yapar. Özelleştirilmiş bir ek kart kullanılıyorsa “GPIO Default” seçeneği seçilip pin adı, yönü ve giriş çıkış standardı kartın mahiyetine göre değiştirilir.

DE2-115 sistem kurucu aynı zamanda Şekil 4.22'de "4" ile gösterilen kısımda, ön tanımlı ayarları geri yükleme, ayarları geri yükleme ve kullanıcı board ayarlarını kaydetme seçenekleri sunar. Kullanıcı o anki board ayarlarını bir .cfg dosyasına kaydedip daha sonra DE2-115 sistem kurucuya tekrar yükleyebilirler [19].

5. UYGULAMA

Bu kısımda gerçek zamanlı sistem için uygun olan Sobel filtresinin FPGA boardu üzerinde gerçekleştirilmesinden bahsedilmiştir. İlk olarak FPGA üzerine gömülen yazılımın çalışabilmesi için gerekli olan yapılmış tanımlamalardan bahsedilmiş daha sonra ise sistemde tasarlanan modüller ve çalışma şekilleri anlatılmıştır.

5.1 Yapılan Tanımlamalar

DE2-115 board'u üzerinde Quartus II programı yardımıyla yapılan uygulamada öncelikle sistemin çalışması için ve üzerinde uygulama geliştirilebilmesi için bazı temel tanımlamalar yapılmıştır. Bu tanımlamalar kısaca aşağıdaki gibidir.

- Clock: Sistemin senkronize çalışmasını sağlayan saat birimidir. DE2-115 üzerinde giriş (input) olarak tanımlı 50 MHz'lik 3 adet saat bulunmaktadır.
- EJTAG: Hata ayıklama ve performans ince ayarı için kullanılan birimdir. Giriş/çıkış olarak tanımlanmıştır.
- VGA: DE2-115 board'u üzerinden DE-15 konektörü kullanılarak bir ekrana görüntü çıktısı almak için kullanılan birimdir. Bu kısımda VGA'nın yatay ve dikey senkronizasyon sinyalleri ile üç ayrı renk sinyali (kırmızı, yeşil, mavi) çıkış olarak tanımlanmıştır. Böylece kameradan alınan görüntüler standart bir CRT ekran üzerinde görüntülenebilmektedir.
- SDRAM: İşlenecek olan verileri geçici süreyle saklamak için SDRAM modülü tanımlanmıştır. SDRAM 'in adresi ve senkronizasyon saati tanımlanmıştır.
- GPIO: Genel amaçlı giriş çıkış bağlantı portu olan GPIO portuna görüntü işleme sisteminin en önemli modülü olan kamera bağlanmıştır. Bu bağlantıda kameranın ihtiyaç duyduğu senkronizasyon, validate, reset, trigger gibi parçalar sisteme tanımlanmıştır.

5.2 Sistem Tasarımı

DE2-115 sistem kurucusu kullanılarak yapılan tanımlamalarla kullanılacak olan elemanlar için gerekli kod altyapısı hazırlanmıştır. Quartus II programının MegaWizard Plug-in manager'ı kullanılarak görüntü işleme fonksiyonlarının FPGA üzerine implemente edilebilmesi için gerekli olan alt fonksiyonlar hazır kütüphanelerden seçilerek eklenir.

Uygulamada FPGA üzerine kenar belirlemek için kullanılan sobel filtresi gömülmüş olup, sobel fonksiyonu implementasyonu için gerekli olan alt fonksiyonlar aşağıda açıklanmıştır.

5.2.1 Line Buffer (Satır Tamponu)

Verilog'un yapısı nedeniyle kameradan görüntüleri C programlama dilinde yapılacak bir uygulama gibi doğrudan iki boyutlu bir diziye aktarıp, indeks elemanı aracılığıyla üzerinde filtre işlemi uygulanacak piksel değerlerine erişilememektedir. Bu sebepten dolayı da yapılacak işlem için öncelikle bir line buffer tasarlanmıştır. Bu tasarımla pikseller 3'erli sırayla alınarak üzerinde gerekli matematiksel işlemler yapılmaya hazır hale getirilmiştir.

5.2.2 Multiply-Add (Çarpım-Toplam)

Line buffer'dan üçerli sırayla alınan pikseller daha sonra alt_multadd megafunction'ına gönderilir. Sobel filtresi uygulamasında yapılacak olan matris işlemi şu şekilde özetlenebilir:

$$\begin{matrix} G_1 & G_2 & G_3 & X_1 & X_2 & X_3 \\ G_4 & G_5 & G_6 * & X_4 & X_5 & X_6 = X_{5\text{filt}} \\ G_7 & G_8 & G_9 & X_7 & X_8 & X_9 \end{matrix} \quad (5)$$

Burada filtre uygulanacak olan sadece X_5 pikselidir. Yapılacak işlem

$$G_1X_1 + G_2X_2 + G_3X_3 + G_4X_4 + G_5X_5 + G_6X_6 + G_7X_7 + G_8X_8 + G_9X_9 = X_{5filtered} \quad (6)$$

denklemini şeklinde tanımlanabilir. Multiply-add megafunction'ında çarpım işlemleri her bir ikili için ayrı ayrı gerçekleştirilir ve bir sonraki adıma geçilir.

5.2.3 Parallel Adder (Paralel Toplayıcı)

Parallel Adder megafunction'ına gelen ikili çarpım sonuçları önce yan yana satır oluşturan üçlüler olmak üzere toplanır. Daha sonra elde edilen üç satır toplamı da toplanarak o piksel için filtre uygulanmış değer bulunur.

Düşey ve yatayda olmak üzere iki kısımdan oluşan sobel filtresi bütün piksellere uygulanarak iki ayrı matris elde edildikten sonra, elde edilen matrislerin mutlak değerleri alınarak toplanır. Böylece sobel filtresinden geçirilmiş görüntü elde edilmiş ve ekranda görüntülenmesi için VGA modülüne gönderilir.

Ana modül kısmında, oluşturulan sobel alt modülüne giriş verisi olarak görüntünün gri ölçekli kısmı gönderilmesi gerekmektedir. Gri görüntü oluşturulurken

$$Gri = 0.299 * K + 0.587 * Y + 0.114 * M \quad (7)$$

fonksiyonu kullanılır. Bu fonksiyondan da görülebileceği gibi Y değerinin (yeşil) ağırlığı K'nin (kırmızı) iki katı, M'nin (mavi) ise yaklaşık 5 katıdır. Bu nedenden dolayı görüntünün doğrudan yeşil kanalı sobel filtresine gönderilerek gri ölçekli görüntü oluşturma aşaması bertaraf edilmiş olur.

5.3 SİSTEMİN GERÇEKLEŞTİRİLMESİ

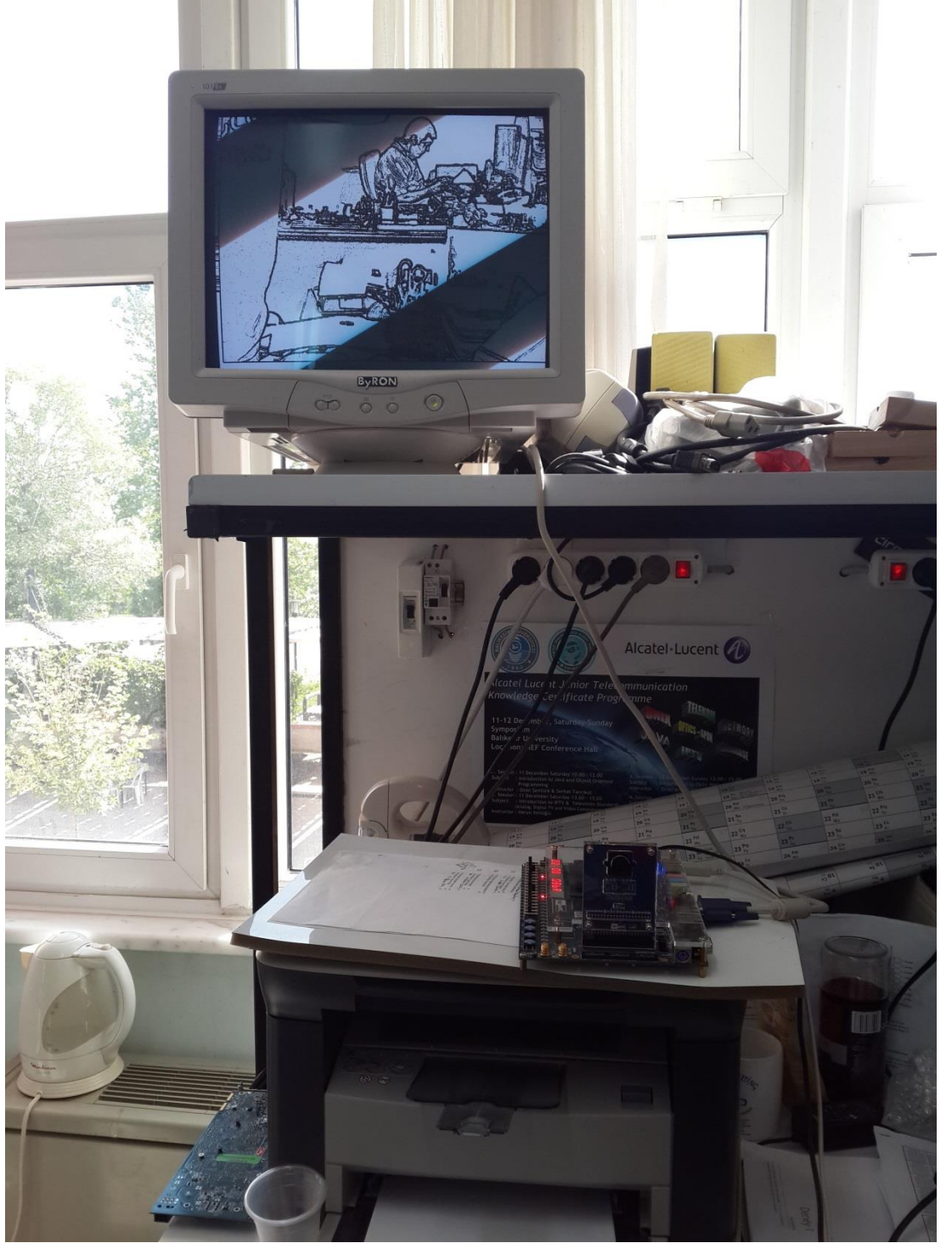
Altera DE2-115 eğitim board'u ve Terasic TRDB_D5M kamera kullanılarak kurulan sistemin üzerine kamera modülünün baz yazılımı temel alınarak kenar belirleme amaçlı gerçek zamanlı sobel filtresi yüklenmiştir.



Şekil 5.1: Sistemde Gerçek Kamera Görüntüsü



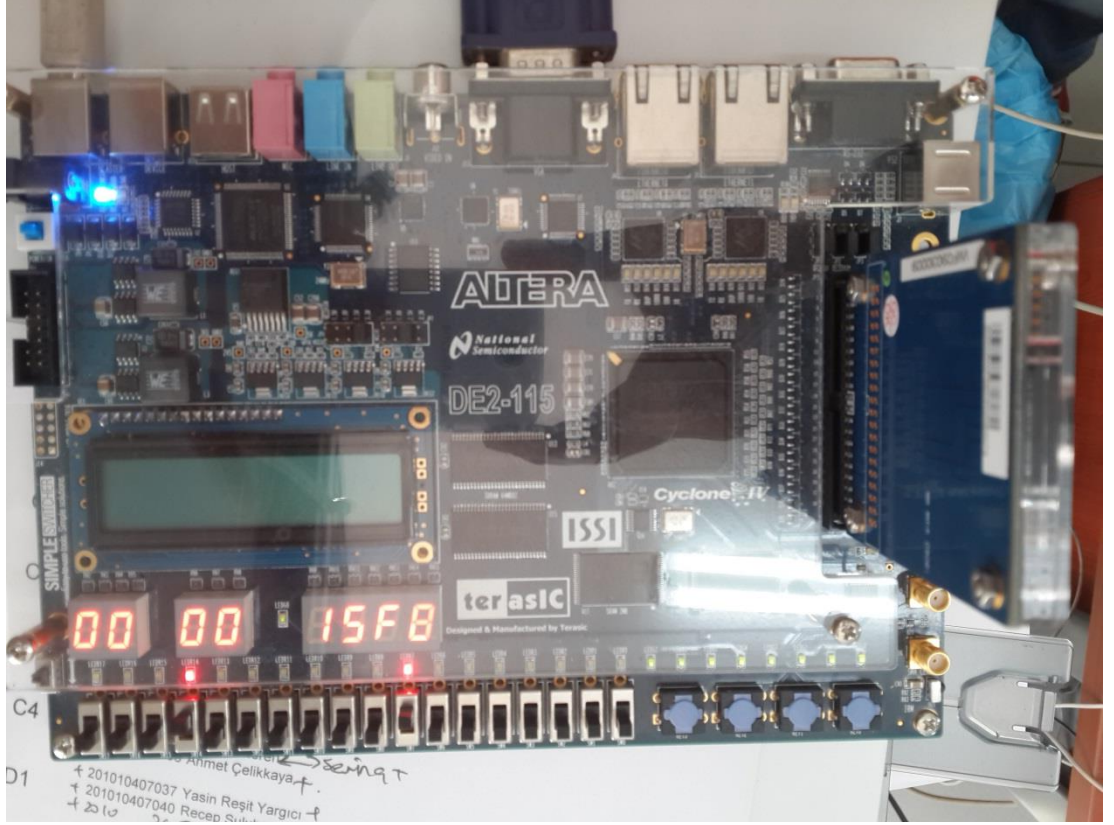
Şekil 5.2: Sistemde Gri Ölçekli Görüntü



Şekil 5.3: Düşük Eşik Değerli Sobel Filtresi



Şekil 5.4: Yüksek Eşik Değerli Sobel Filtresi



Şekil 5.5: Çalışan Sistemin Görüntüsü

Şekil 5.1’de sistemin müdahale edilmemiş, doğrudan kamera çıkışını aktaran hali görülmektedir. Şekil 5.2’de ise Sobel filtresinin uygulanabilmesi için gerekli olan gri ölçekli hale getirilmiş bulunmaktadır.

Şekil 5.3’te Sobel filtresinin gerçek zamanlı uygulaması görülmektedir. Bu uygulamada eşik değeri düşük tutulmuş buna bağlı olarak da nesnelerin dış hatlarının yanı sıra içerisinde barındırdıkları desen ve çizgiler de vurgulanmış bu yüzden nesne tanımlama işlemi için gerekli altyapı oluşturulamamıştır.

Şekil 5.4’te ise aynı uygulamanın yüksek eşik değeri tanımlanarak yapılmış ve böylece nesnelerin sadece dış hatlarının elde edilmesi sağlanmıştır. Burada eşik değeri ön tanımlı olarak kullanılmıştır. Benzer bir uygulamada da [23] yine ön tanımlı eşik değeri tek karelik görüntü işlemesi için kullanılmıştır. Çalışılacak ortam belli olan uygulamalarda ön tanımlı eşik değeri zaman ve işlemci gücünden tasarruf ettireceğinden dolayı daha verimli çalışacaktır. Buna karşılık olarak adaptif filtre daha çok işlemci zamanı gerektirdiğinden gerçek zamanlı sistemde çalışmaya uygun değildir.

Son olarak Şekil 5.5'te sistemin çalışma anındaki görüntüsü görülmektedir. 7 segment display'de sistem açıldığı andan beri işlenen kare sayısı onaltılık tabanda yazmaktadır.

6. SONUÇ VE ÖNERİLER

Görüntü işleme robotik sistemlerin otomatik olarak çalışır hale getirilebilmesi için gerekli altyapıyı sağlayabilecek bir işlemdir. Ancak gerektirdiği işlemci yükü nedeniyle seri mikroişlemcilerde gerçek zamanlı uygulaması yapılamamaktadır [28,29]. Robot sistemlerinde gömülü sistemlerin kullanılması gerekliliğinden dolayı ise masaüstü bilgisayarlar işlemci gücü olarak yeterli olabilse bile boyut olarak uygun olamamaktadır [15]. Bu nedenlerden dolayı literatürde de bir çok örneği olduğu üzere [30,31,32] gerçek zamanlı görüntü işleme uygulaması için FPGA sistemi tercih edilmiştir. FPGA paralel işlemler yapabilmesiyle diğer mikroişlemci ve mikrokontrolör sistemlerinden ayrı bir yerde konumlanmaktadır. Modüller halinde çalışması nedeniyle bağımsız şekilde kısımlar halinde programlanabilmektedir. Bu da ayrı filtrelerin ana modülden bağımsız bir şekilde geliştirilebilmesine olanak sağlamaktadır.

Görüntü işlemenin uzaysal (spatial) alanda uygulaması, görüntünün piksellerinin matrislere dönüştürülerek üzerlerinde matematiksel işlemler yapılmasıyla gerçekleşir. Paralel işlem yapabilmenin görüntü işlemede en büyük avantajı büyük boyutlu matrislerin (tek bir görüntü karesinin) bir kaç saat çevriminde işlenebilmesi ve böylece daha düşük frekans hızındaki donanımlarda gerçek zamanlı işleme yapılabilmesine olanak sağlamasıdır. Görüntü işleme sistemlerinde seri (ardıl) işlem yüksek kapasiteli veri transferi gerekmektedir. Bunun yerine paralel işlem yapıldığında ise aynı anda görüntü karesinin birden fazla pikseli işleminden geçirilerek çok daha yüksek hızlara daha düşük konfigürasyon ve maliyetle ulaşmak mümkün olabilmektedir.

Literatürdeki örneklerinden de yola çıkarak [5,22,23,24,29,30,33,34] kenar belirleme uygulaması için Sobel filtre algoritmasının Verilog dilinde yazılarak FPGA üzerine gömülmesinin uygun olduğuna karar verilmiştir. Çalışmada Sobel filtresinin çekirdek matrisi, çalışma prensibi ve matematiksel denklemleri anlatılmıştır.

Bu tez çalışmasında literatürdeki çalışmaların [15,16,21,28,31,37,38] aksine Altera firmasının üst seviye FPGA eğitim boardu kullanılmıştır. FPGA sistemi

üzerine kamera modülü kullanarak robot sistemlerinde kullanım için görüntü işleme entegrasyonu yapılmıştır. En yaygın kullanımlı kenar belirleme filtresi olan Sobel filtresinin uygulaması yapılmış ve gerçek zamanlı olarak çalıştığı gözlemlenmiştir. Sobel filtresi için farklı eşik değerleri deneysel olarak denenmiş ve nesne tanımlama altyapısı için uygun olan değer ampirik yöntemlerle bulunmuştur. Daha ileri çalışmalarda kalıp ve nesne tanımlama uygulamaları, yön belirleme algoritmaları ve haritalama çalışmaları yapılması; farklı çözünürlükler, saniye başına kare değerleri gibi geliştirmeler ve yeni uygulamalar yapılması mümkündür.

7. KAYNAKLAR

- [1] Wilson, R. P., *Design Recipes for FPGAs*, Oxford: Newnes, (2007) 5-9
- [2] Trussel, H. J. and Vrhel M. J., *Fundamentals of Digital Imaging*, Cambridge University Press (2008) 2-3
- [3] Dobbin B., “Kodak engineer had revolutionary idea: the first digital camera [online]”, (18 February 2013) [http://www.seattlepi.com/business/article/](http://www.seattlepi.com/business/article/Kodak-engineer-had-revolutionary-idea-the-first-1182624.php) Kodak-engineer-had-revolutionary-idea-the-first-1182624.php
- [4] Vick P. C., “KH-11 KENNAN RECONNAISSANCE IMAGING SPACERAFT [online]”, (18 February 2013), <http://www.globalsecurity.org/space/systems/kh-11.htm>
- [5] Wasfy W., Zheng H., “General Structure Design for Fast Image Processing Algorithms Based upon FPGA DSP Slice” *Physics Procedia* 33 (2012) 690 – 697
- [6] Newman E. M., “Fiftieth Anniversary of First Digital Image Marked [online]” (21 February 2013) http://www.nist.gov/public_affairs/releases/image_052407.cfm
- [7] Elamaran V., Praveen A., Reddy M. S., Aditya V. L., Suman K., “FPGA Implementation of Spatial Image Filters using Xilinx System Generator” *Procedia Engineering* 38 (2012) 2244-2249
- [8] Çonkır T., “Encoder Nedir ve En Uygun Encoder Nasıl Seçilir [online]” (28 February 2013) <http://www.otomasyondergisi.com.tr/arsiv/yazi/encoder-nedir-ve-uygun-encoder-nasil-secilir>
- [9]”Sinyal Üreticiler [online]” (01 March 2013), <http://www.otomasyonhaber.net/haber/1360-teknik-bilgiler-sinyal-ureticiler-encoderler.html>

- [10] Fisher R., Perkins S., Walker A., Wolfart E., “Sobel Edge Detector [online], (02 March 2013) <http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>
- [11] Graus C., “Image Processing for Dummies with C# and GDI+ Part 2 - Convolution Filters [online], (08 March 2013), <http://www.codeproject.com/Articles/2008/Image-Processing-for-Dummies-with-C-and-GDI-Part-2>.
- [12] Wyman C., “Sobel Filtering [online]” (09 March 2013), <http://homepage.cs.uiowa.edu/~cwyman/classes/spring08-22C251/homework/sobel.pdf>
- [13] Fisher R., Perkins S., Walker A., Wolfart E “Laplacian/Laplacian of Gaussian [online]”, (11 March 2013), <http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>
- [14] Landes D. S., *The Unbound Prometheus: Technological Change and Industrial Development in Western Europe from 1750 to the Present*. Cambridge, New York: Press Syndicate of the University of Cambridge (1969).
- [15] Pauwels K., Tomasi M., Diaz J., Ros E., Van Hulle M. M., “A Comparison of FPGA and GPU for Real-Time Phase-Based Optical Flow, Stereo, and Local Image Features” *IEEE Transactions On Computers*, Vol. 61, No. 7, July 2012.
- [16] Saadi S., Guessoum A., Bettayeb M., “ABC optimized neural network model for image deblurring with its FPGA implementation” *Microprocessors and Microsystems* 37 (2013) 52–64.
- [17] Reisinger D., “Tesla eyes 'autopilot' partnership with Google [online]” (8 May 2013) http://news.cnet.com/8301-11386_3-57583233-76/tesla-eyes-autopilot-partnership-with-google/
- [18] Golnabi A., Asadpour A. “Design and application of industrial machine vision systems” *Robotics and Computer-Integrated Manufacturing* 23 (2007) 630–637.
- [19] Corp., Altera, DE2_115 User Manual", (12 April 2013), ftp://ftp.altera.com/up/pub/Altera_Material/12.1/Boards/DE2-115/DE2_115_User_Manual.pdf, (2011).

- [20] Inc. Terasic, Technologies, “TRBD-D5M User Guide” (12 April 2013), http://www.terasic.com.tw/attachment/archive/282/TRDB_D5M_UserGuide.pdf
- [21] Belean B., Borda M., Bertrand L. G., Terebes R., “FPGA based system for automatic cDNA microarray image processing” *Computerized Medical Imaging and Graphics* 36 (2012) 419– 429.
- [22] Zhu F., Ning J., Ren Y., Peng J., “Optimization of Image Processing in Video-based Traffic Monitoring” *Elektronika IR Elektrotehnika*, ISSN 1392-1215, Vol. 18, No. 8, 2012.
- [23] Lin C-S., Yeh P. T., Chen D-C., Chiou Y-C., Lee C-H., “The identification and filtering of fertilized eggs with a thermal imaging system” *Computers and Electronics in Agriculture* 91 (2013) 94–105.
- [24] Abdou I. E., Pratt K.W., “Quantitative Design and Evaluation of Enhancement / Thresholding Edge Detectors”, *Proceedings Of The IEEE*, Vol. 67, No. 5, May 1979.
- [25] Latham V. L., Latham L. E., Oppenheimer S. B., “Desktop computer-based image analysis of cell surface fluorescence patterning from a photographic source”, *Acta histochem. (Iena)* 98,295-300 (1996) Gustav Fischer Verlag Jena . Stuttgart· New York.
- [26] Gonzalez-Cencerrado A., Pena B., Gil A., “Coal flame characterization by means of digital image processing in a semi-industrial scale PF swirl burner” *Applied Energy* 94 (2012) 375–384.
- [27] Selver M.A., Akay O., Alim F., Bardakçı S., Ölmez M., “An automated industrial conveyor belt system using image processing and hierarchical clustering for classifying marble slabs”, *Robotics and Computer-Integrated Manufacturing* 27 (2011) 164–176.
- [28] Ibarra-Manzano M-A., Almanza-Ojeda D-L., “An FPGA Implementation for Image Interpretation Based on Adaptive Boosting Algorithm in the Real-Time Systems” *Procedia Technology* 3 (2012) 187 – 195.

- [29] Tu Y-W., Ho M-T., “Design and implementation of robust visual servoing control of an inverted pendulum with an FPGA-based image co-processor”, *Mechatronics* 21 (2011) 1170–1182.
- [30] Kreizer M., Liberzon A., “Three-dimensional particle tracking method using FPGA-based real-time image processing and four-view image splitter” *Exp Fluids* (2011) 50:613–620 DOI 10.1007/s00348-010-0964-3.
- [31] Trieu D. B. K., Maruyama T., “Real-time color image segmentation based on mean shift algorithm using an FPGA” *J Real-Time Image Proc* DOI 10.1007/s11554-012-0319-9.
- [32] Verma B., Sardana H. K., “Real Time Image Segmentation using watershed algorithm on FPGA” *International Journal of Engineering Science and Technology (IJEST)*, Vol. 3 No. 6 June 2011, 5282-5287.

EKLER

8. EKLER

8.1 EK-A

Not: Kodların tamamı çok uzun olması nedeniyle burada sadece ana modülün kodu verilmiştir. Kodların tamamı CD’de bulunmaktadır.

```
ENET1_RX_COL,  
ENET1_RX_CRS,  
ENET1_RX_DATA,  
ENET1_RX_DV,  
ENET1_RX_ER,  
ENET1_TX_CLK,  
ENET1_TX_DATA,  
ENET1_TX_EN,  
ENET1_TX_ER,
```

```
////////// TV Çözücü //////////
```

```
TD_CLK27,  
TD_DATA,  
TD_HS,  
TD_RESET_N,  
TD_VS,
```

```
////////// USB 2.0 OTG //////////
```

```
OTG_ADDR,  
OTG_CS_N,  
OTG_DACK_N,  
OTG_DATA,  
OTG_DREQ,  
OTG_FSPEED,  
OTG_INT,  
OTG_LSPEED,  
OTG_RD_N,  
OTG_RST_N,  
OTG_WE_N,
```


////////// IR Alıcı //////////
IRDA_RXD,

////////// SDRAM //////////
DRAM_ADDR,
DRAM_BA,
DRAM_CAS_N,
DRAM_CKE,
DRAM_CLK,
DRAM_CS_N,
DRAM_DQ,
DRAM_DQM,
DRAM_RAS_N,
DRAM_WE_N,

////////// SRAM //////////
SRAM_ADDR,
SRAM_CE_N,
SRAM_DQ,
SRAM_LB_N,
SRAM_OE_N,
SRAM_UB_N,
SRAM_WE_N,

////////// Flash //////////
FL_ADDR,
FL_CE_N,
FL_DQ,
FL_OE_N,
FL_RST_N,
FL_RY,
FL_WE_N,
FL_WP_N,

////////// GPIO, GPIO D5M, 5Mpiksellik kameraya bağlanır //////////
D5M_D,
D5M_FVAL,
D5M_LVAL,

```

D5M_PIXLCLK,
D5M_RESET_N,
D5M_SCLK,
D5M_SDATA,
D5M_STROBE,
D5M_TRIGGER,
D5M_XCLKIN
);

//=====
// PARAMETRE bildirimleri
//=====

//=====
// PORT bildirimleri
//=====

////////// CLOCK //////////
input          CLOCK_50;
input          CLOCK2_50;
input          CLOCK3_50;

////////// Sma //////////
input          SMA_CLKIN;
output         SMA_CLKOUT;

////////// LED //////////
output         [8:0]    LEDG;
output         [17:0]   LEDR;

////////// KEY //////////
input          [3:0]    KEY;

////////// EJTAG //////////
inout         [6:0]    EX_IO;

```

////////// SW //////////

input [17:0] SW;

////////// SEG7 //////////

output [6:0] HEX0;
output [6:0] HEX1;
output [6:0] HEX2;
output [6:0] HEX3;
output [6:0] HEX4;
output [6:0] HEX5;
output [6:0] HEX6;
output [6:0] HEX7;

////////// LCD //////////

output LCD_BLON;
inout [7:0] LCD_DATA;
output LCD_EN;
output LCD_ON;
output LCD_RS;
output LCD_RW;

////////// RS232 //////////

output UART_CTS;
input UART_RTS;
input UART_RXD;
output UART_TXD;

////////// Klavye ve Fare için PS2 //////////

inout PS2_CLK;
inout PS2_CLK2;
inout PS2_DAT;
inout PS2_DAT2;

////////// SDCARD //////////

output SD_CLK;
inout SD_CMD;
inout [3:0] SD_DAT;

```

input                SD_WP_N;

////////// VGA //////////
output               [7:0]      VGA_B;
output               VGA_BLANK_N;
output               VGA_CLK;
output               [7:0]      VGA_G;
output               VGA_HS;
output               [7:0]      VGA_R;
output               VGA_SYNC_N;
output               VGA_VS;

////////// Ses //////////
input                AUD_ADCDAT;
inout                AUD_ADCLRCK;
inout                AUD_BCLK;
output               AUD_DACDAT;
inout                AUD_DACLCK;
output               AUD_XCK;

////////// EEPROM için I2C //////////
output               EEP_I2C_SCLK;
inout                EEP_I2C_SDAT;

////////// Ses TV çözücü için I2C //////////
output               I2C_SCLK;
inout                I2C_SDAT;

////////// Ethernet 0 //////////
output               ENET0_GTX_CLK;
input                ENET0_INT_N;
input                ENET0_LINK100;
output               ENET0_MDC;
inout                ENET0_MDIO;
output               ENET0_RST_N;
input                ENET0_RX_CLK;
input                ENET0_RX_COL;
input                ENET0_RX_CRS;
input                [3:0]      ENET0_RX_DATA;

```

input		ENET0_RX_DV;
input		ENET0_RX_ER;
input		ENET0_TX_CLK;
output	[3:0]	ENET0_TX_DATA;
output		ENET0_TX_EN;
output		ENET0_TX_ER;
input		ENETCLK_25;

////////// Ethernet 1 //////////

output		ENET1_GTX_CLK;
input		ENET1_INT_N;
input		ENET1_LINK100;
output		ENET1_MDC;
inout		ENET1_MDIO;
output		ENET1_RST_N;
input		ENET1_RX_CLK;
input		ENET1_RX_COL;
input		ENET1_RX_CRS;
input	[3:0]	ENET1_RX_DATA;
input		ENET1_RX_DV;

input		ENET1_RX_ER;
input		ENET1_TX_CLK;
output	[3:0]	ENET1_TX_DATA;
output		ENET1_TX_EN;
output		ENET1_TX_ER;

////////// TV Çözücü //////////

input		TD_CLK27;
input	[7:0]	TD_DATA;
input		TD_HS;
output		TD_RESET_N;
input		TD_VS;

////////// USB 2.0 OTG //////////

output	[1:0]	OTG_ADDR;
output		OTG_CS_N;
output	[1:0]	OTG_DACK_N;
inout	[15:0]	OTG_DATA;
input	[1:0]	OTG_DREQ;

inout		OTG_FSPEED;
input	[1:0]	OTG_INT;
inout		OTG_LSPEED;
output		OTG_RD_N;
output		OTG_RST_N;
output		OTG_WE_N;
////////// IR A1c1 //////////		
input		IRDA_RXD;
////////// SDRAM //////////		
output	[12:0]	DRAM_ADDR;
output	[1:0]	DRAM_BA;
output		DRAM_CAS_N;
output		DRAM_CKE;
output		DRAM_CLK;
output		DRAM_CS_N;
inout	[31:0]	DRAM_DQ;
output	[3:0]	DRAM_DQM;
output		DRAM_RAS_N;
output		DRAM_WE_N;
////////// SRAM //////////		
output	[19:0]	SRAM_ADDR;
output		SRAM_CE_N;
inout	[15:0]	SRAM_DQ;
output		SRAM_LB_N;
output		SRAM_OE_N;
output		SRAM_UB_N;
output		SRAM_WE_N;
////////// Flash //////////		
output	[22:0]	FL_ADDR;
output		FL_CE_N;
inout	[7:0]	FL_DQ;
output		FL_OE_N;
output		FL_RST_N;
input		FL_RY;
output		FL_WE_N;
output		FL_WP_N;

////////// GPIO, GPIO D5M, 5Mpiksellik kameraya bağlanır //////////

```
input      [11:0]      D5M_D;
input      D5M_FVAL;
input      D5M_LVAL;
input      D5M_PIXLCLK;
output     D5M_RESET_N;
output     D5M_SCLK;
inout      D5M_SDATA;
input      D5M_STROBE;
output     D5M_TRIGGER;
output     D5M_XCLKIN;
```

```
//=====
// REG/WIRE bildirimleri
//=====
```

```
wire [15:0] Read_DATA1;
wire [15:0] Read_DATA2;
```

```
wire [11:0] mCCD_DATA;
wire      mCCD_DVAL;
wire      mCCD_DVAL_d;
wire [15:0] X_Cont;
wire [15:0] Y_Cont;
wire [9:0]  X_ADDR;
wire [31:0] Frame_Cont;
wire      DLY_RST_0;
wire      DLY_RST_1;
wire      DLY_RST_2;
wire      DLY_RST_3;
wire      DLY_RST_4;
wire      Read;
reg [11:0] rCCD_DATA;
reg      rCCD_LVAL;
reg      rCCD_FVAL;
wire [11:0] sCCD_R;
wire [11:0] sCCD_G;
wire [11:0] sCCD_B;
wire      sCCD_DVAL;
```

```

wire          sdram_ctrl_clk;
wire [9:0]   oVGA_R;          //   VGA Red[9:0]
wire [9:0]   oVGA_G;          //   VGA Green[9:0]
wire [9:0]   oVGA_B;          //   VGA Blue[9:0]

//güç açılınca başlama
wire        auto_start;
//=====
// Yapısal kodlama
//=====
// D5M
assign D5M_TRIGGER    =    1'b1; // tRIGGER
assign D5M_RESET_N    =    DLY_RST_1;
assign VGA_CTRL_CLK = ~VGA_CLK;

assign LEDR          =    SW;
assign LEDG          =    Y_Cont;
assign UART_TXD = UART_RXD;

//en yüksek 8 bit'i alır
assign VGA_R = oVGA_R[9:2];
assign VGA_G = oVGA_G[9:2];
assign VGA_B = oVGA_B[9:2];

//D5M okuma
always@(posedge D5M_PIXLCLK)
begin
    rCCD_DATA <=    D5M_D;
    rCCD_LVAL <=    D5M_LVAL;
    rCCD_FVAL <=    D5M_FVAL;
end

//güç açıldığında otomatik başlama
assign auto_start = ((KEY[0])&&(DLY_RST_3)&&(!DLY_RST_4))? 1'b1:1'b0;
//Modülü baştan başlatma
Reset_Delay          u2    (    .iCLK(CLOCK2_50),
                                .iRST(KEY[0]),

```



```

        .oRST_0(DLY_RST_0),
        .oRST_1(DLY_RST_1),
        .oRST_2(DLY_RST_2),
        .oRST_3(DLY_RST_3),
        .oRST_4(DLY_RST_4)
    );

//D5M görüntü yakalama
CCD_Capture      u3      (
    .oDATA(mCCD_DATA),
    .oDVAL(mCCD_DVAL),
    .oX_Cont(X_Cont),
    .oY_Cont(Y_Cont),
    .oFrame_Cont(Frame_Cont),
    .iDATA(rCCD_DATA),
    .iFVAL(rCCD_FVAL),
    .iLVAL(rCCD_LVAL),
    .iSTART(!KEY[3]|auto_start),
    .iEND(!KEY[2]),
    .iCLK(~D5M_PIXLCLK),
    .iRST(DLY_RST_2)
);

//D5M ham verisinin RGB verisine dönüşümü
`ifdef VGA_640x480p60
RAW2RGB          u4      (
    .iCLK(D5M_PIXLCLK),
    .iRST(DLY_RST_1),
    .iDATA(mCCD_DATA),
    .iDVAL(mCCD_DVAL),
    .oRed(sCCD_R),
    .oGreen(sCCD_G),
    .oBlue(sCCD_B),
    .oDVAL(sCCD_DVAL),
    .iX_Cont(X_Cont),
    .iY_Cont(Y_Cont)
);

`else
RAW2RGB          u4      (
    .iCLK(D5M_PIXLCLK),
    .iRST_n(DLY_RST_1),
    .iData(mCCD_DATA),
    .iDval(mCCD_DVAL),
    .oRed(sCCD_R),
    .oGreen(sCCD_G),
    .oBlue(sCCD_B),
    .oDval(sCCD_DVAL),
    .iZoom(SW[16]),
    .iX_Cont(X_Cont),

```

```

        .iY_Cont(Y_Cont)
    );
`endif
//Kare sayım görüntüsü
SEG7_LUT_8      u5    (    .oSEG0(HEX0),.oSEG1(HEX1),
                        .oSEG2(HEX2),.oSEG3(HEX3),
                        .oSEG4(HEX4),.oSEG5(HEX5),
                        .oSEG6(HEX6),.oSEG7(HEX7),
                        .iDIG(Frame_Cont[31:0])
    );

s dram_pll      u6    (
                        .inclk0(CLOCK2_50),
                        .c0(s dram_ctrl_clk),
                        .c1(DRAM_CLK),
                        .c2(D5M_XCLKIN), //25M
`ifdef VGA_640x480p60
                        .c3(VGA_CLK) //25M
`else
                        .c4(VGA_CLK) //40M
`endif
    );

//SDRam kare tamponu olarak yazma ve okuma yapar
S dram_Control  u7    (    //    HOST Side

                        .RESET_N(KEY[0]),
                        .CLK(s dram_ctrl_clk),

                        //    FIFO Write Side 1

                        .WR1_DATA({1'b0,sCCD_G[11:7],sCCD_B[11:2]}),
                        .WR1(sCCD_DVAL),
                        .WR1_ADDR(0),
`ifdef VGA_640x480p60
                        .WR1_MAX_ADDR(640*480/2),
                        .WR1_LENGTH(8'h50),
`else
                        .WR1_MAX_ADDR(800*600/2),
                        .WR1_LENGTH(8'h80),

```

```

`endif

.WR1_LOAD(!DLY_RST_0),
.WR1_CLK(D5M_PIXLCLK),

// FIFO Write Side 2

.WR2_DATA({1'b0,sCCD_G[6:2],sCCD_R[11:2]}),
.WR2(sCCD_DVAL),
.WR2_ADDR(23'h100000),
`ifdef VGA_640x480p60
.WR2_MAX_ADDR(23'h100000+640*480/2),
.WR2_LENGTH(8'h50),
`else
.WR2_MAX_ADDR(23'h100000+800*600/2),
.WR2_LENGTH(8'h80),
`endif

.WR2_LOAD(!DLY_RST_0),
.WR2_CLK(D5M_PIXLCLK),

// FIFO Read Side 1
.RD1_DATA(Read_DATA1),
.RD1(Read),
.RD1_ADDR(0),
`ifdef VGA_640x480p60
.RD1_MAX_ADDR(640*480/2),
.RD1_LENGTH(8'h50),
`else
.RD1_MAX_ADDR(800*600/2),
.RD1_LENGTH(8'h80),
`endif

.RD1_LOAD(!DLY_RST_0),

.RD1_CLK(~VGA_CTRL_CLK),

// FIFO Read Side 2
.RD2_DATA(Read_DATA2),
.RD2(Read),
.RD2_ADDR(23'h100000),
`ifdef VGA_640x480p60

```

```

.RD2_MAX_ADDR(23'h100000+640*480/2),
                                                                    .RD2_LENGTH(8'h50),
`else

.RD2_MAX_ADDR(23'h100000+800*600/2),
                                                                    .RD2_LENGTH(8'h80),
`endif

.RD2_LOAD(!DLY_RST_0),

.RD2_CLK(~VGA_CTRL_CLK),

                                                                    // SDRAM Side
                                                                    .SA(DRAM_ADDR),
                                                                    .BA(DRAM_BA),
                                                                    .CS_N(DRAM_CS_N),
                                                                    .CKE(DRAM_CKE),
                                                                    .RAS_N(DRAM_RAS_N),
                                                                    .CAS_N(DRAM_CAS_N),
                                                                    .WE_N(DRAM_WE_N),
                                                                    .DQ(DRAM_DQ),
                                                                    .DQM(DRAM_DQM)
                                                                    );
//D5M I2C control
I2C_CCD_Config      u8      ( // Host Side
                                                                    .iCLK(CLOCK2_50),
                                                                    .iRST_N(DLY_RST_2),
                                                                    .iEXPOSURE_ADJ(KEY[1]),
                                                                    .iEXPOSURE_DEC_p(SW[0]),
                                                                    .iZOOM_MODE_SW(SW[16]),
                                                                    // I2C Side
                                                                    .I2C_SCLK(D5M_SCLK),
                                                                    .I2C_SDAT(D5M_SDATA)
                                                                    );

// kanal girişi (renkli) RGB
wire [9:0] wVGA_R = Read_DATA2[9:0];
wire [9:0] wVGA_G = {Read_DATA1[14:10],Read_DATA2[14:10]};
wire [9:0] wVGA_B = Read_DATA1[9:0];

// sobel çıkışı
wire [9:0] wSobel;

```

```

Sobel sobel0 (
    .iCLK(VGA_CTRL_CLK),
    .iRST_N(DLY_RST_2),
    .iTHRESHOLD(SW[9:2]), // sobel'in eşik değeri
    .iDVAL(Read),
    .iDATA(wVGA_G), // yeşil/gri kanal (eşik değeri ayarlandığında gri)
    .oDATA(wSobel)
);

// SW[15]
// aktif: sobel
// inaktif: renkli
wire [9:0] red_or_sobel = SW[15] ? wSobel : wVGA_G;
wire [9:0] green_or_sobel = SW[15] ? wSobel : wVGA_G;
wire [9:0] blue_or_sobel = SW[15] ? wSobel : wVGA_G;

//VGA DISPLAY
VGA_Controller    u1    ( // Host Side
    .oRequest(Read),
    .iRed(red_or_sobel),
    .iGreen(green_or_sobel),
    .iBlue(blue_or_sobel),
    // VGA Side
    .oVGA_R(oVGA_R),
    .oVGA_G(oVGA_G),
    .oVGA_B(oVGA_B),
    .oVGA_H_SYNC(VGA_HS),
    .oVGA_V_SYNC(VGA_VS),
    .oVGA_SYNC(VGA_SYNC_N),
    .oVGA_BLANK(VGA_BLANK_N),
    // Control Signal
    .iCLK(VGA_CTRL_CLK),
    .iRST_N(DLY_RST_2),
    .iZOOM_MODE_SW(SW[16])
);

endmodule

```

8.2 EK-B

8.2.1.1 FPGA Cihazı

- Cyclone IV EP4CE115F29 cihazı
- 114,480 adet Mantık Elemanı
- 432 adet M9K Hafıza Bloęu
- 3,888 Kilobit gömülü hafıza
- 4 adet PLL

8.2.1.2 FPGA Ayarlama

- JTAG ve AS modlarında ayarlama
- EPCS64 seri ayarlama cihazı
- Board üstünde USB Blaster devresi

8.2.1.3 Hafıza Cihazları

- 128MB (32Mx32bit) SDRAM
- 2MB (1Mx16) SRAM
- 8 bit modlu 8MB (4Mx16) Flash
- 32Kb EEPROM

8.2.1.4 SD Kart Yuvası

- SD kart erişimi için SPI ve 4-bit SD modu sağlar

8.2.1.5 Baęlantılar

- İki Adet Ethernet 10/100/1000 Mbps portu
- High Speed Mezzanine Card (HSMC)
- Ayarlanabilir giriş çıkış standardı (gerilim seviyeleri:3.3/2.5/1.8/1.5V)
- A ve B tipi USB
 - USB 2.0 uyumlu misafir bilgisayar ve cihaz kontrolü saęlar
 - Tam hız ve düşük hızda veri transferini destekler
 - Bilgisayar sürücüsü bulunur
- 40-pin genişletme portu
 - Ayarlanabilir giriş çıkış standardı (gerilim seviyeleri: 3.3/2.5/1.8/1.5V)
- VGA çıkış baęlantısı
 - VGA Dijital Analog Çevirici DAC (yüksek hızlı, üçlü Dijital Analog Çevirici)
- RS-232 portu için akış kontrollü DB9 seri konektörü
- PS/2 fare klavye

8.2.1.6 Saat

- 3 adet 50MHz osilatör saat girişı
- SMA konektörleri (harici saat giriş/çıkışı)

8.2.1.7 Ses

- 24 bit enkoder/dekoder (CODEC)
- Line giriş ve çıkışı ve mikrofon girişı jakları

8.2.1.8 Ekran

- 16x2 LCD modülü

8.2.1.9 Anahtarlar ve Göstergeler

- 18 Adet kaydırılabilir anahtar ve 4 Adet basılabilir düğme
- 18 Adet kırmızı ve 9 Adet yeşil kullanıcı LED'i
- 8 adet yedi parçalı gösterge

8.2.1.10 Diğer Özellikler

- Kızılötesi uzaktan kumanda alıcı modülü
- TV dekoderi (NTSC/PAL/SECAM) ve TV giriş bağlantısı

8.2.1.11 Güç

- Masaüstü DC giriş
- LM3150MH anahtarlama ve düşürücü regülatörleri