



**YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ
ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ BÖLÜMÜ**

Bitirme Tezi/Projesi

**GERÇEK ZAMANLI LİNEER GÖRÜNTÜ İŞLEME
ALGORİTMALARININ FPGA İLE GERÇEKLENMESİ**

Tez/Proje Danışmanı :

Prof. Dr. Vedat TAVŞANOĞLU

Öğrencinin Numarası, Adı, Soyadı

04014086
Ahmet Remzi ÖZCAN

İstanbul, 2009

İÇİNDEKİLER

SİMGE VE KISALTMA LİSTESİ.....	iv
ŞEKİL LİSTESİ.....	v
TABLO LİSTESİ.....	vii
ÖNSÖZ	viii
ÖZET	ix
ABSTRACT.....	x
 BÖLÜM 1	
GİRİŞ	1
 BÖLÜM 2	
DONANIM	2
2.1. FPGA.....	2
2.2. Uygulama Geliştirme Kartı	3
2.2.1. Altera Cyclone II	4
2.3. Görüntü Sensörü.....	6
2.4. Sensör Kartı	7
 BÖLÜM 3	
İŞLEVSEL BİRİMLERİN TASARIMI	9
3.1. Seri Haberleşme Birimi	9
3.1.1. RS232 Protokolü	9
3.1.2. Seri Haberleşme Birimi Tasarımı	10
3.2. Ekran Kontrol Birimi	12
3.2.1. VGA Protokolü	12
3.2.2. Ekran Kontrol Birimi Tasarımı	14
3.3. Görüntü Yakalama Birimi.....	16
3.4. Sensör Kontrol Birimi	18
3.4.1. SCCB Protokolü	18
3.4.2. Sensör Kontrol Birimi Tasarımı	20
3.5. Konvolüsyon Birimi	22
3.5.1. Konvolüsyon Birimi Tasarımı	24

BÖLÜM 4	
SİSTEMLERİN GERÇEKLENMESİ	27
4.1. Seri Port Kaynaklı Basit Konvolüsyon Sisteminin Gerçeklenmesi	27
4.2. Seri Port Kaynaklı Ekran Kontrollü Basit Konvolüsyon Sisteminin Gerçeklenmesi	30
4.3. Ana Sisteminin Gerçeklenmesi	33
 BÖLÜM 5	
SONUÇ	40
 KAYNAKLAR	41
Ek 1 Konvolüsyon Birimi Blok Şeması	43
Ek 2 Uygulama Geliştirme Kartı Üzerinde Sistemin Gerçeklenmesi	44
Ek 3 Sistemle Haberleşmeyi Sağlayan MATLAB Kodu	45
ÖZGEÇMİŞ	46

SİMGE ve KISALTMA LİSTESİ

V	volt
mW	mili Watt
nm	nanometre
s	saniye
us	mikro saniye
Hz	Hertz
KHz	kilo Hertz
Mhz	mega Hertz
Mbps	mega bit per second (saniye başına mega bit)
Fps	frame per second (saniye başına kare)
Kbayt	kilo Bayt
Mbayt	mega Bayt
Kbit	kilo bit
Mbit	mega bit
RAM	Random Access Memory (Rastgele Erişimli Bellek)
SRAM	Statik RAM (Statik Rastgele Erişimli Bellek)
SDRAM	Synchronous Dynamic RAM (Senkron Dinamik Rastgele Erişimli Bellek)
SD	Secure Digital
CMOS	Complementary Metal–Oxide–Semiconductor (Bütünleyici Metal Oksit Yarıiletken)
CCD	Charge-Coupled Device (Yüklenme İliştirilmiş Araç)
LED	Light Emitting Diode (ışık saçan diyod)
MUX	Multiplexer (Çoklayıcı)
LUT	Look-Up Table (Başvuru Tablosu)
LEs	Logic Elements
PLL	Phase-Locked Loop
RS232	Recommended Standard 232
VGA	Video Graphics Array (Video Grafik Dizisi)
RGB	Red-Green-Blue (Kırmızı-Yeşil-Mavi)
YUV	Y: Luminance U: Chrominance1 V: Chrominance2
ADC	Analog Digital Converter (Analog Sayısal Dönüştürücü)
DAC	Digital Analog Converter (Sayısal Analog Dönüştürücü)
SCCB	Serial Camera Control Bus (Seri Kamera Kontrol Hattı)
CS	Chip Select (Çip Seçim)
G/Ç	Giriş/Çıkış
I2C	Inter-Integrated Circuit
FPGA	Field-Programmable Gate Array (Alanda Programlanabilir Kapı Dizileri)

ŞEKİL LİSTESİ

Şekil 2.1	FPGA	2
Şekil 2.2	FPGA gerçekleştirme mimarileri	3
Şekil 2.3	Altera DE1 uygulama geliştirme kartı blok şeması	4
Şekil 2.4	Altera DE1 uygulama geliştirme kartı	4
Şekil 2.5	Altera Cyclone II	5
Şekil 2.6	LE lojik işlem birimi	6
Şekil 2.7	Görüntü sensörü	7
Şekil 2.8	Sensör kartı ve objektifi	7
Şekil 2.9	Görüntü sensörünün blok diyagramı	8
Şekil 3.1	1 baytlık bir verinin gönderilmesi	9
Şekil 3.2	Seri haberleşme alıcı alt birimi zaman durum diyagramı	11
Şekil 3.3	Seri haberleşme birimi blok şeması	11
Şekil 3.4	Seri haberleşme alıcı alt birimi simülasyon sonucu	12
Şekil 3.5	Ekran kontrol sinyallerinin işlev ve zamanlamaları	13
Şekil 3.6	Ekran kontrol sinyalleri boş çevrim zamanlamaları	14
Şekil 3.7	Altera DE1 direnç ağı DAC'ı (Red sinyali için)	15
Şekil 3.8	Ekran kontrol birimi	15
Şekil 3.9	Tasarlanan ekran kontrol biriminin uygulama geliştirme kartı üzerinde denenmesi	16
Şekil 3.10	Ekran kontrol biriminin simülasyon sonucu	16
Şekil 3.11	Görüntü yakalama birimi	17
Şekil 3.12	Görüntü sinyalleri zaman durum diyagramı	18
Şekil 3.13	Görüntü yakalama birimi simülasyon sonucu	19
Şekil 3.14	SCCB haberleşmesi blok şeması	20
Şekil 3.15	SCCB yazma protokolü	20
Şekil 3.16	SCCB protokolünde iletimin başlaması ve sonlanması	21
Şekil 3.17	Sensör kontrol birimi	21
Şekil 3.18	FPGA uygulama geliştirme kartı üzerinde anahtar tanımlamaları	22
Şekil 3.19	Sensör kontrol birimi simülasyon sonucu	22
Şekil 3.20	Görüntü üzerinde konvolüsyon işlemi	23
Şekil 3.21	Görüntü ile laplace matrisinin konvolüsyon sonucu	23
Şekil 3.22	Gelen piksel verilerinin blok ram'e yazılması	24
Şekil 3.23	Görüntünün i(2,2) ve i(2,23) piksel bilgileri geldiğinde kaydedicilerin durumu	25

Şekil 3.24	Görüntünün $i(4,0)$ piksel bilgisi geldiğinde kaydedicilerin durumu	25
Şekil 3.25	Konvolüsyon işlem alt birimi blok şeması	26
Şekil 4.1	Konvolüsyon kontrol verisi	27
Şekil 4.2	Seri port kaynaklı basit konvolüsyon sistemi hiyerarşisi	28
Şekil 4.3	Seri port kaynaklı konvolüsyon sistemine gönderilen görüntü	28
Şekil 4.4	Seri port kaynaklı konvolüsyon sisteminden alınan görüntü	29
Şekil 4.5	MATLAB üzerinde yapılan konvolüsyon sonucu	29
Şekil 4.6	SRAM'a yazma ve SRAM'den okuma zamanlaması	30
Şekil 4.7	Sisteme gönderilen görüntü	31
Şekil 4.8	Konvolüsyon işlemi yapılırken ekran durumu	32
Şekil 4.9	Konvolüsyon sonucu	32
Şekil 4.10	SRAM, ekran ve gelen veri saat işaretlerinin zaman durum diyagramları	34
Şekil 4.11	Önbellek ara birimi blok şeması	34
Şekil 4.12	Ana sistemin giriş ve çıkışları	35
Şekil 4.13	Ana sistemin blok diyagramı	36
Şekil 4.14	Ekran üzerinde konvolüsyon sonucu	36
Şekil 4.15	Sistem hızını sınırlayan kablo ve kapı gecikmelerinden biri	38
Şekil 4.16	FPGA üzerinde sistemin yerleşimi	38
Şekil 4.17	Ana sistemin hiyerarşisi	39

TABLO LİSTESİ

Tablo 3.1	Verilen formatlarda tarama için gerekli frekans ve tarama çevrimleri	14
Tablo 3.2	Seçilen CS değerlerine göre sensörün kimlik değerleri	14
Tablo 4.1	İlk sistem için FPGA kaynaklarının kullanımı	30
Tablo 4.2	İkinci sistem için FPGA kaynaklarının kullanımı	33
Tablo 4.2	Ana sistem için FPGA kaynaklarının kullanımı.....	37

ÖNSÖZ

Öğrenim hayatım boyunca benden maddi manevi desteklerini esirgemeyen ve beni bugünlere getiren aileme çok teşekkür ederim.

Bitirme tezini almamda beni teşvik eden ve destekleyen değerli hocam Prof. Dr. Vedat TAVŞANOĞLU'na, proje çalışması boyunca bana yardımcı olan Arş. Gör. Murathan ALPAY, Arş. Gör. Nerhun YILDIZ ve Arş. Gör. Evren CESUR' a teşekkürlerimi bir borç bilirim.

ÖZET

Bu çalışmada FPGA uygulama geliştirme kartı üzerinde gerçek zamanlı görüntü işleyen bir sistem oluşturulmuştur. Gerçeklenen bu sistem görüntü sensöründen alınan hareketli görüntü üzerinde konvolüsyon işlemi yaparak sonucu ekran üzerinde görüntülemektedir. Yapılan konvolüsyon işlemiyle görüntü üzerinde yumuşatma, keskinleştirme, kenar belirleme gibi filtreleme işlemleri gerçekleştirilebilmiştir.

Çalışma sonunda gerçekleştirilen sistem kendi başına çalışacak biçimde tasarlanmış buna ek olarak konvolüsyon şablonunun dışarıdan seri port ile değiştirilebilmesi sağlanmıştır. Böylece görüntü üzerine uygulanacak filtrenin değiştirilmesi için sistemin tekrar sentezlenmesine gerek kalmamıştır. Çalışmada görüntüyü yakalayan birimden, ekran üzerinde sonucu görüntüleyen birime kadar bütün birimlerin FPGA üzerinde gerçekleştirilmesi sayesinde, sistemi ve bileşenlerini aynı altyapıyı kullanan ileri tasarımlarda kullanmak zor olmayacaktır. Tasarımların VHDL dili ile davranışsal olarak yapılması sistem üzerinde değişiklik yapmayı kolaylaştırmaktadır. Sistemin bu haliyle yüz, plaka tanıma gibi daha üst seviyedeki çalışmalar için temel teşkil edeceği düşünülmektedir.

ABSTRACT

In this project, a system which is able to process images in real time is developed using FPGA development kit. This system that is realized displays the result which is obtained by convolution over the moving image taken from the image sensor. Many kinds of filtering processes can be realized such as smoothing, sharpening and edge detecting owing to this convolution process.

The system that was realized at the end of the study was designed to run itself in addition to this it is provided the convolution mask to be able to change via serial port externally. Thus, there is no need the system to be synthesized in order to change the filter which is applied to the image. With this study, it is not going to be hard to use the system and its components which are designed owing to be realized the whole units from which captures the image to which displays the image on the screen on an FPGA in the next designs use the same infrastructure as it. Designing the system behavioral by using VHDL makes it simple to change something desired.

It is supposed that the present state of the system serves as a basis on the high level applications such as face or metal-tag recognition etc.

1 GİRİŞ

Projede gerçek zamanlı görüntü işleyebilen gömülü bir sistemin oluşturulması amaçlanmıştır. Bu amaca yönelik olarak yapılan çalışma, sistemi oluşturan birimlerin tasarlanması ve sistemin gerçekleştirilmesi aşamalarından meydana gelmektedir. Sistem ve sistemi oluşturan birimler VHDL dili ile davranışsal olarak tasarlanmış, Altera Quartus II [1] yazılımı ile kullanılan FPGA yongasına özgü olarak sentezlenmiştir. Yapılan tasarımların simülasyonları Modelsim [2] üzerinde gerçekleştirilmiş, gerçekleştirilen sistemin bilgisayar ile haberleşmesinde MATLAB yazılımı kullanılmıştır.

Bu tezin ikinci bölümünde projenin gerçekleştirilmesinde kullanılan temel donanımsal aygıtlar olan FPGA, uygulama geliştirme kartı ve görüntü sensörü hakkında bilgiler verilmiş, içyapıları kısaca anlatılmıştır.

Üçüncü bölümde gerçekleştirilen sistemi oluşturan birimlerin nasıl çalıştığı hakkında açıklamalarda bulunulmuş, tasarımları detaylı olarak verilmiştir.

Dördüncü bölümde projenin seyri boyunca tasarlanan birimlerin birlikte ne kadar kararlı çalıştığını anlamak ve varsa hataları düzeltmek üzere gerçekleştirilen ara sistemler ve proje sonunda gerçekleştirilen ana sistem yer almaktadır.

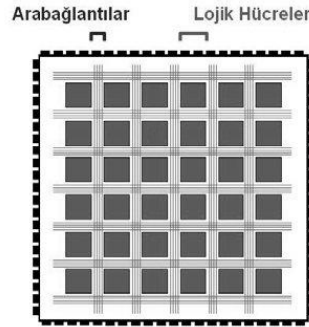
Son bölümde ise sonuç kısmına yer verilmiştir.

2 DONANIM

Bu bölümde proje tasarımının gerçekleştirilmesinde kullanılan donanımsal bileşenler hakkında bilgi verilmektedir.

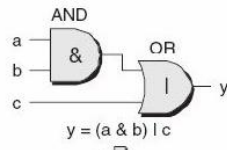
2.1 FPGA

Günümüzde sayısal sistem tasarımlarında çok sık kullanılan FPGA (Field Programmable Gate Array – Alan Programlanabilir Kapı Dizileri) , programlanabilir mantık blokları ve bu blokları birbirine bağlayan ara bağlantılardan oluşan tümdevrelerdir. VHDL (Very high speed integrated circuit Hardware Description Language - Çok Hızlı Tümlşik Devre Donanım Tanımlama Dili) dilinin sistem tasarımına getirdiği esneklik ve özellikle davranışsal (behavioral) tasarım tekniği sayesinde karmaşık sistemleri çok kısa sürede tasarlamak, gelişen FPGA teknolojisi sayesinde de bu tasarımları çok kısa sürede gerçeklemek mümkün olmuştur. FPGA’ın üretici firmalara göre farklı mimarileri olsa da temelde FPGA’lerin büyük çoğunluğu tasarım yapılandırmasını SRAM hücreleri ile sağladığından binlerce kez programlanabilir.



Şekil 2.1 FPGA

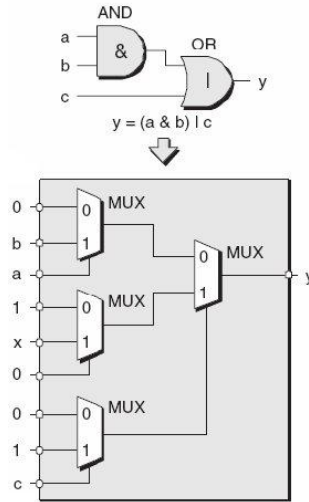
Programlanabilir hücre mimarili FPGA’ler istenen mantıksal yapıyı gerçeklemede farklı yapıları kullanabilir. Bu yapılardan en çok kullanılanları MUX ve LUT tabanlı hücre mimarileridir [3]. LUT tabanlı hücre mimarisinde bir mantık hücresi giriş sinyallerinin alacağı her değer için sonucun ne olacağının tutulduğu bir başvuru tablosu (look-up table) içermektedir. MUX tabanlı hücre mimarisinde ise bir mantık hücresi sadece çoğullayıcılardan oluşur ve istenen fonksiyon bu yapılarla gerçekleştirilir.



Doğruluk Tablosu

a	b	c	y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

a) LUT tabanlı gerçekleştirme



b) MUX tabanlı gerçekleştirme

Şekil 2.2 FPGA gerçekleştirme mimarileri

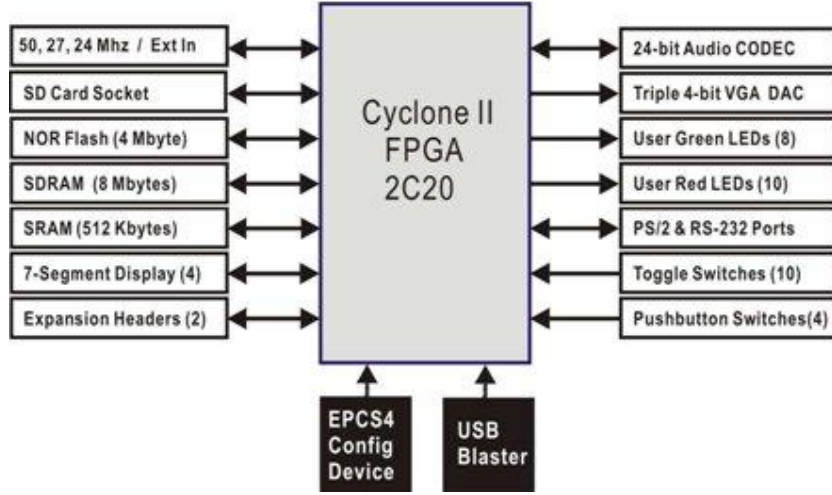
2.2 Uygulama Geliştirme Kartı

Bu projede Terasic firmasının üretmiş olduğu Altera DE1 uygulama geliştirme kartı kullanılmıştır [4]. Bu kart üzerinde;

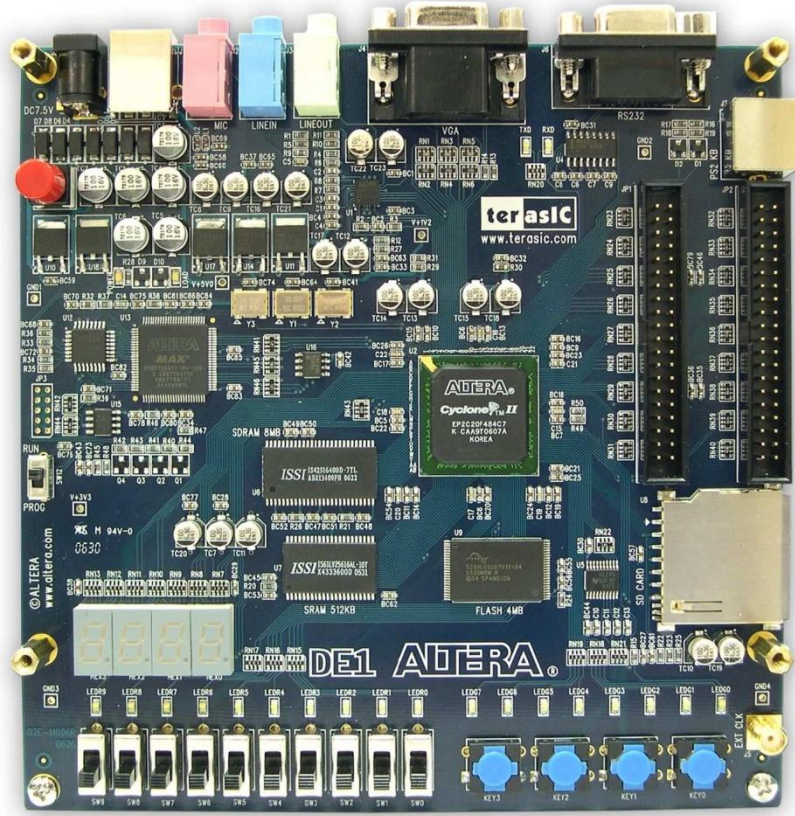
- 20000 lojik işlem birimine sahip Altera Cyclone II 2C20 FPGA tümdevresi,
- Cyclone II 2C20 FPGA'in yapılandırılmasını sağlayan 4 Mbit seri flash hafıza,
- 8 Mbayt (4M x 16) SDRAM,
- 4 Mbayt Flash Hafıza,
- 512 Kbayt (256K x 16) SRAM,
- SD kart soketi
- 10 kaydırmalı anahtar, 4 buton, 18 LED ve 4 adet 7 parçalı gösterge,
- 24-bit Audio CODEC, line-in, line-out, ve mikrofon girişleri,
- RS232, VGA (12 bit), PS/2 portları,
- 2 adet 40 pinlik genişletme portu bulunmaktadır.

Kartın blok şeması Şekil 2.3'de verilmektedir.

Projede kart üzerinde bulunan SRAM, VGA, RS232, 40 pinlik genişletme portları, ledler, kaydırmalı anahtarlar ve butonlar kullanılmıştır. Sistemde saat işareti olarak yine kart üzerinde bulunan 50Mhz'lik kristal kullanılmıştır.



Şekil 2.3 Altera DE1 uygulama geliştirme kartı blok şeması



Şekil 2.4 Altera DE1 uygulama geliştirme kartı

2.2.1 Altera Cyclone II

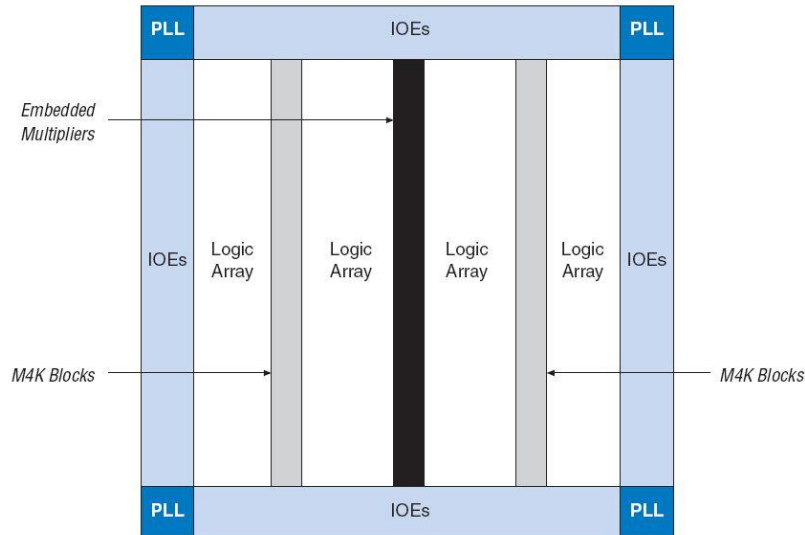
Kullanılan geliştirme kartı üzerinde bulunan FPGA, Altera firması tarafından üretilmiş Altera Cyclone II 2C20'dir [5]. 90-nm üretim teknolojisine sahiptir ve üzerinde 20000 lojik işlem birimi (LEs) bulunmaktadır. Bu lojik birimler dışında 238 Kbit blok ram, 26

tane 18 x 18 çarpıcı ve 4 tane sistem saat yöneticisi (PLL) bulunmaktadır. FPGA'nın blok diyagramı Şekil 2.5'de yer almaktadır.

Tümdevre üzerinde 4Kbit'lik parçalar halinde yer alan blok ramler çift portludur (true dual port). Dolayısıyla bu ramlerde farklı adreslerden aynı anda iki okuma ve yazma işlemi gerçekleştirilebilir. Düşük gecikme süresi dolayısıyla 260 MHz ve üzeri hızlarda çalışabilmektedir. Veri genişliği 1, 2, 4, 8, 9, 16, 18, 32 ve 36 bit olarak yapılandırılabilir.

18x18 çarpma işlemi 250 MHz hızında gerçekleştirebilen çarpıcı bloklar aynı zamanda iki bağımsız 9x9 çarpıcı olarak da ayarlanabilir.

Gelişmiş G/Ç desteğine sahip olan tümdevrede farksal G/Ç standartları olan LVDS (Low-Voltage Differential Signaling), RSDS (Reduced Swing Differential Signaling), mini-LVDS, LVPECL (Low-Voltage Positive Emitter-Coupled Logic), farksal HSTL (High-Speed Transceiver Logic) ve farksal SSTL(Stub Series Terminated Logic) bulunmaktadır. Bu farksal iletim standartlarından LVDS ile 640 Mbps iletişim hızına çıkılabilmektedir. Giriş-çıkış pinleri 1.5V, 1.8V, 2.5V ve 3.3V olmak üzere farklı gerilim seviyelerinde giriş, çıkış veya giriş-çıkış ya da yüksek empedans olarak yapılandırılabilir. Tümdevre üzerinde kullanıcıya ait toplam 315 giriş çıkış pini bulunmaktadır.

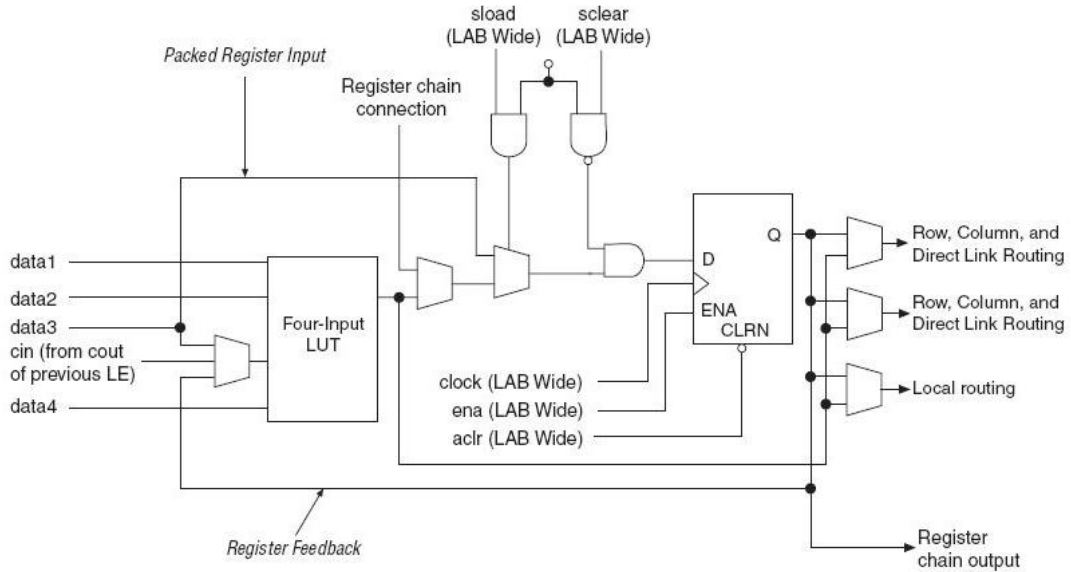


Şekil 2.5 Altera Cyclone II

Altera Cyclone II FPGA'leri LE (Logic Element) olarak adlandırılan lojik işlem birimlerinden oluşmaktadır. Her LE lojik işlem birimi;

- Lojik işlemlerin gerçekleştirildiği 4 girişli bir LUT,
- Bir programlanabilir kaydedici,
- Elde zincir bağlantısı,
- Kaydedici zincir bağlantısı içermektedir.

LE birimleri arasındaki bağlantı ağı sayesinde bu yapılar birbirlerine bağlanıp, aslında basit olan lojik işlem birimlerinden çok karmaşık sistemler meydana getirilebilir. LE lojik işlem birimleri iki farklı modda çalışabilir. Bunlar normal mod ve aritmetik moddur. LE, genel lojik uygulamalar gerçekleştirirken normal modda, sayıcı, akümülatör karşılaştırıcı gibi yapılar gerçekleştirirken ise aritmetik modda çalışmaktadır. LE içinde yer alan LUT normal modda 4 girişli bir LUT olarak atanırken, aritmetik modda 3 girişli bir LUT olarak yapılandırılır. Normal modda çalışan bir LE yapısının blok diyagramı Şekil 2.6'de verilmektedir.



Şekil 2.6 LE lojik işlem birimi

2.3 Görüntü Sensörü

Görüntü sensörleri üzerlerine optik bir mercek yardımıyla düşürülen görüntüyü işleyerek elektriksel sinyallere dönüştüren tümdevrelerdir. Günümüzde dijital kameralar ve diğer görüntüleme aygıtlarında bu sensörler kullanılmaktadır. Görüntü sensörleri temel yapıları itibarıyla CMOS ve CCD sensörler olarak ikiye ayrılmaktadır.



a) CCD Sensor



b) CMOS Sensor

Şekil 2.7 Görüntü sensörü

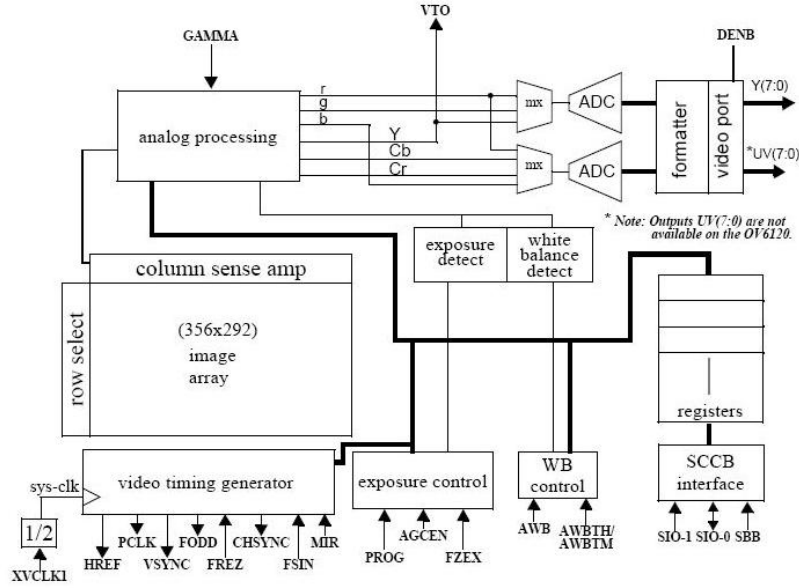
CCD görüntü sensörleri bir tabakanın üstüne dizilmiş ışığa duyarlı foto diyotlardan oluşmaktadır ve bu diyotlar ışığı, düşen ışığın şiddeti oranında elektrik gerilimine çevirirler. Bu sensörler ışığa karşı CMOS sensörlerden daha duyarlıdır ve üretilen görüntünün niteliği daha kalitelidir. Bunun yanında daha pahalıdır ve daha fazla güç harcarlar. Bir CCD sensöründen alınan veri analogdur ve verinin sayısal ortamda işlenebilmesi için öncelikle bir ADC ile işlenmesi gereklidir. Buna karşılık CMOS sensörlerden alınan veri dijitaldir. Hatta çoğu CMOS sensör tümdevresinin üzerinde bir mikroişlemci bulunur ve elde edilen görüntü daha sensör tümdevresi üzerindeyken ön bir işleme tabi tutulur. Projede maliyet, kullanım kolaylığı gibi nedenler dolayısıyla CMOS görüntü sensörü kullanılması kararlaştırılmıştır.

2.4 Sensör Kartı

Projede üzerinde OmniVision firmasının 1/4" CMOS görüntü sensörünü taşıyan bir sensör kartı kullanılmıştır [6]. Kullanılan sensöre uygun objektifiyle beraber gelen kart üzerinde, sensörün çalışması için gerekli olan kapasiteler, dirençler ve sensör için saat işaretini üreten bir kristal de bulunmaktadır. Bu haliyle beslemeleri yapıldığında sensör kartı üzerinden görüntü alınmaya başlanabilir.

**Şekil 2.8** Sensör kartı ve objektifi

Üzerinde bulunan OV6620 CMOS görüntü sensörü maksimum 30 fps hızıyla, 352x288 çözünürlükte görüntü yakalayabilmektedir. Sayısal veri çıkışı sayesinde ek bir dönüştürücü ihtiyacı yoktur. Sensör üzerinde yer alan analog işlemci ile elde edilen görüntü üzerinde pozlama, beyaz dengesi, kazanç, kontrast ayarlamaları yapılır ve yine sensör üzerinde yer alan SCCB seri iletişim birimi ile bu ayarlamalar kontrol edilebilir. Sensör tam çalışma esnasında 80mW güç tüketmektedir. Sensörde sayısal veri çıkışı YUV veya RGB görüntü formatında olabilmektedir. Sensörden çıkan Y (8 bit) ve UV(8 bit) veri sinyalleriyle 16 bitlik piksel bilgisi, PCLK, VSYNC ve HREF kontrol sinyalleriyle görüntü zamanlaması bildirilir. SIO-0 ve SIO-1 ise sırasıyla SCCB portunun veri giriş çıkış ve saat sinyalleridir. Şekil 2.9'de OV6620 sensörünün blok diyagramı yer almaktadır.



Şekil 2.9 Görüntü sensörünün blok diyagramı

Sensörün fiziksel görüntü çıkışı YUV formatına göre ayarlandığından ve projede gri formattaki görüntüler üzerinde işlem yapılacağından YUV formatıyla çalışılması kararlaştırılmıştır.

3 İŞLEVSEL BİRİMLERİN TASARIMI

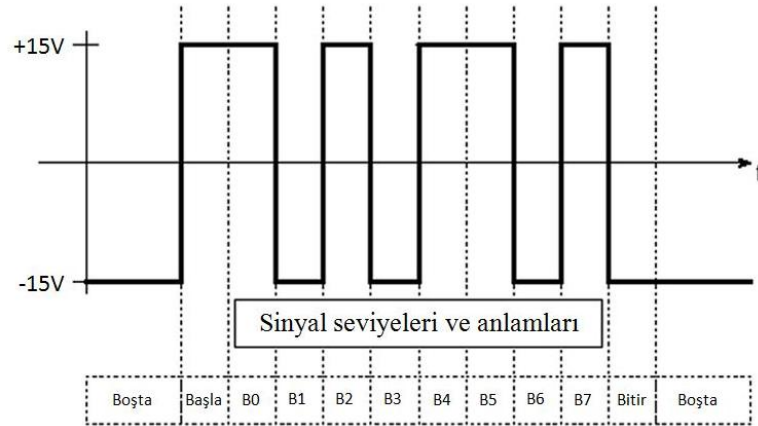
Proje süresinde hedeflenen sistemin gerçekleşmesine yönelik birçok işlevsel birim tasarlanmıştır. Altera Quartus II yazılımı ile sentezlenen bu tasarımların simülasyonları Modelsim yazılımı üzerinde gerçekleştirilmiştir.

3.1 Seri Haberleşme Birimi

Projede tasarlanan sistem ile bilgisayar arasındaki haberleşmeyi sağlamak amacıyla RS232 protokolünü kullanan bir seri haberleşme birimi tasarlanmıştır.

3.1.1 RS232 Protokolü

RS232 seri haberleşmede kullanılan en genel protokoldür [7]. Birbirine yakın birimler arasındaki haberleşmeyi sağlar ve bilgiler +15V ve -15V lojik gerilim seviyeleri ile genelde 10 bitlik parçalar halinde gönderilir. İletimde saat bilgisi gönderilmediği için iletim asenkronudur. Veri bitlerinden önce gönderilen “başla” bitiyle alıcı saatini eşitler ve gönderim başlar. Veri bitlerinin gönderim sırası en düşük ağırlıklı bitten en yüksek ağırlıklı bite doğrudur. Gönderilen 8 veri bitinden sonra gönderilen “bitir” biti ile gönderim tamamlanmış olur. Başlangıç biti lojik 0, bitir biti lojik 1’dir. Gönderilen bu bitler baud olarak adlandırılır. Gerilim seviyesi olarak -15V lojik 1, +15V lojik 0 anlamındadır. Şekil 3.1’de 8 bitlik bir verinin gönderim diyagramı yer almaktadır.



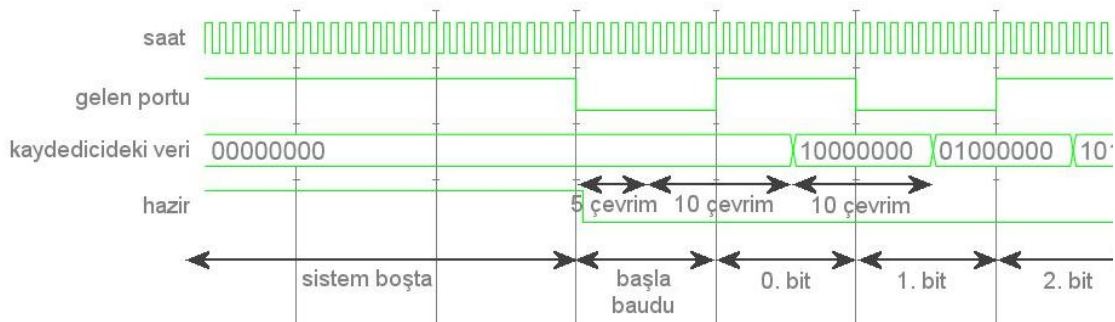
Şekil 3.1 1 baytlık bir verinin gönderilmesi

3.1.2 Seri Haberleşme Birimi Tasarımı

Projede tasarlanan seri haberleşme biriminin iletişim hızı 38400 baud/s olarak seçilmiştir. Bu durumda bir baudun gönderimi yaklaşık 26 us tutacaktır. Seri haberleşme birimi tasarlanırken karşılaşılan en büyük zorluk alıcı durumundayken, verici ile saat senkronizasyonunun sağlanmasıdır. Bu problemi aşmak için seri haberleşme biriminin saat frekansı, gönderim frekansının 10 katı olarak seçilmiştir. Böylece gönderilen bir baudun gönderim çevrimi süresince, gelen portu 10 kez taranmış olacaktır. Kullanılan sistem saat frekansı 50 MHz'dir. Seri haberleşme birimi için kullanılması gereken saat frekansı ise 384 KHz olmalıdır. Seri haberleşme birimi için gerekli olan saat frekansı, sistem saati 130'a bölünerek elde edilmiştir.

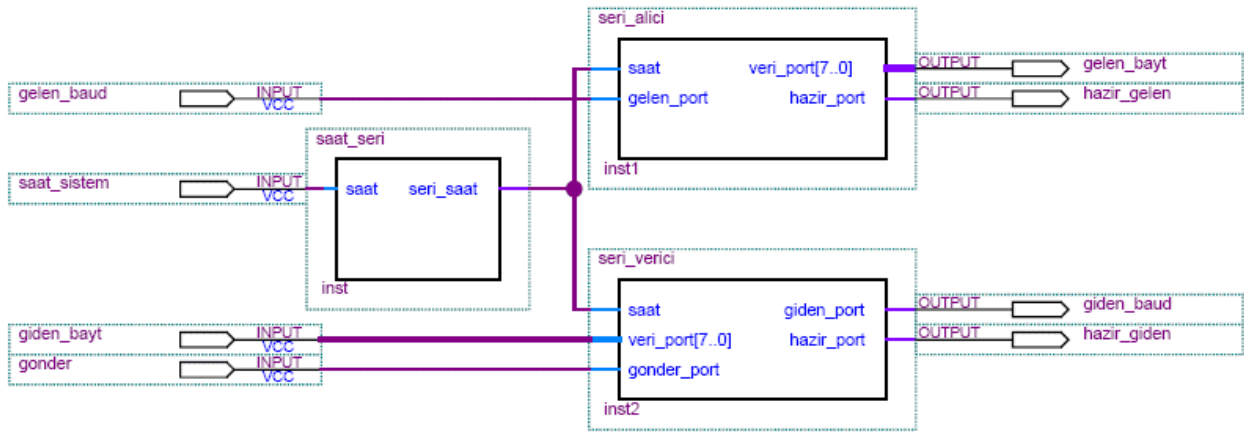
$$bölén_{seri} = \frac{50 \times 10^6}{384 \times 10^3} \cong 130 \quad (3.1)$$

İletim hattı boşta iken gelen portu lojik 1 seviyesindedir. Hat bu durumdayken seri haberleşme birimi de boştaadır. Gelen portunun seviyesi lojik 0'a düştüğünde, seri haberleşme birimi meşgul durumuna girer ve 5 çevrim bekler. Seri haberleşme biriminin saat frekansı haberleşme hızının 10 katı olarak ayarlandığı için bu 5 çevrimin sonunda ilk baud çevriminin ortasına gelmiş olur. Burada "başla" baudundan sonra gelecek baud, verinin en düşük değerklikli biti olacaktır. Bu yüzden gelen bu ilk bauddan sonra 10'ar çevrimlik beklemlerle pin üzerindeki lojik seviye okunarak, bilgi 8 bitlik bir kaydedici üzerinde sağdan sola doğru kaydırılır. 85. çevrim sonunda gelen bilgi bu kaydedici üzerinden okunacak duruma gelmiş olur ve seri haberleşme biriminin "gelen hazır" bayrağı lojik 1 yapılır. Şekil 3.2'de alıcı alt birimin zaman durum diyagramı yer almaktadır.

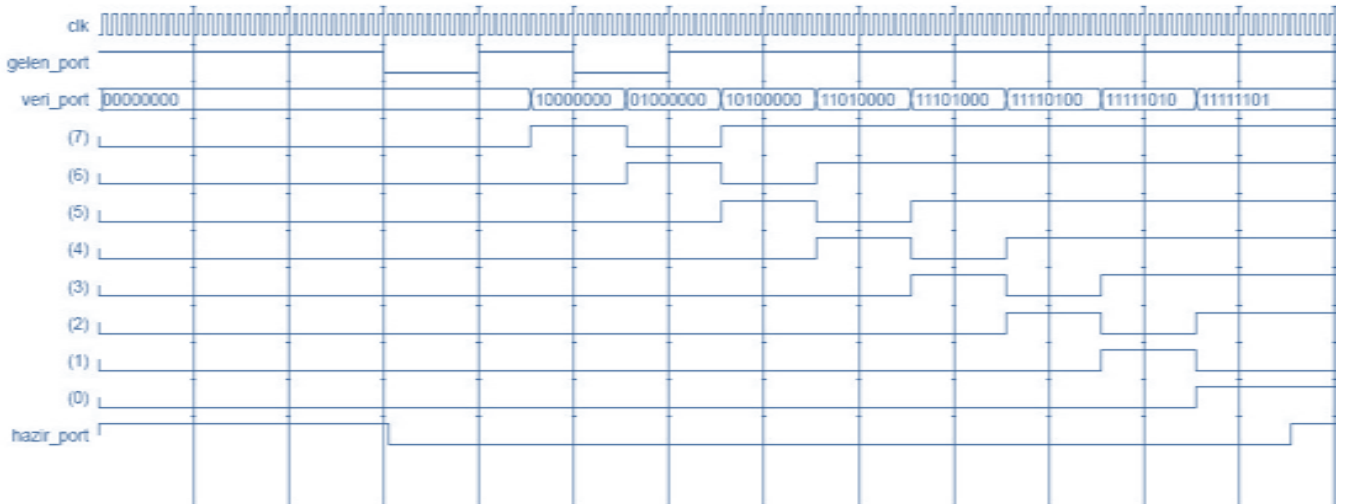


Şekil 3.2 Seri haberleşme alıcı alt birimi zaman durum diyagramı

Seri haberleşme biriminde verinin gönderimi alımına benzer şekilde olmaktadır. Gönderilen veri için saat ayarlaması alıcı tarafında yapılacağından gönderici tasarımı nispeten daha kolaydır. Gönderici birim gönderme durumundayken yeni veri gönderilmesini engellemek amacıyla, ilk olarak “hazır” bayrağı lojik 0 yapılır. Başla komutu olan lojik 0 gönderildikten sonra 10’ar çevrimlik beklemelemlerle 8 veri biti ve son olarak “bitir” biti gönderilir. Son bit gönderildikten sonra “hazır” bayrağı tekrar lojik 1’e çekilir. Şekil 3.3’de seri haberleşme biriminin blok şeması yer almaktadır.



Şekil 3.3 Seri haberleşme birimi blok şeması



Şekil 3.4 Seri haberleşme alıcı alt birimi simülasyon sonucu

3.2 Ekran Kontrol Birimi

Projenin aşamalarından biri olan elde edilen görüntünün ekranda gösterilmesi işleminde görüntünün ekrana aktarılması için FPGA uygulama geliştirme kartı üzerinde bulunan VGA portu kullanılmış ve dolayısıyla tasarım da VGA protokolüne göre yapılmıştır.

3.2.1 VGA Protokolü

VGA bilgisayarlardaki analog görüntü standardını ifade etmektedir ve ilk defa 1988 yılında IBM tarafından piyasaya sürülmüştür. Bu protokolün temel görüntü çözünürlüğü 640x480 olduğu için VGA bu çözünürlüğün kendisini de ifade etmektedir [8].

VGA iletiminde 5 aktif sinyal bulunur. Bunlar;

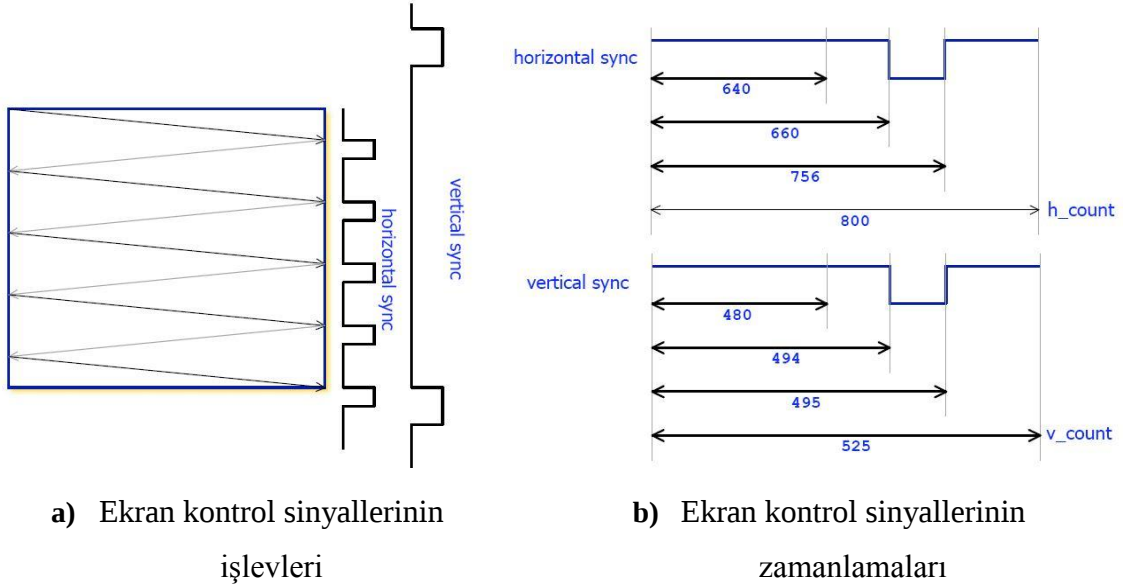
- horizontal sync: Satır eşlemesini sağlayan dijital sinyal,
- vertical sync: Kare eşlemesini sağlayan dijital sinyal,
- red(R): Kırmızı renk bilgisini taşıyan analog sinyal,
- green(G): Yeşil renk bilgisini taşıyan analog sinyal,
- blue(B): Mavi renk bilgisini taşıyan analog sinyaldir.

Burada yer alan renk sinyallerinin gerilim seviyelerinin değişmesiyle diğer bütün renkler elde edilmiş olunur. Ekran üzerinde tarama işlemi yukarıdan aşağıya doğru satır satır yapılır. Bu satırlarda da ekran soldan sağa doğru piksel piksel taranmış olur. Görüntünün hareketli olarak algılanması için ekranın bir saniyede onlarca kez taranması gerekmektedir. Bir saniyede yapılan tarama sayısına tazeleme oranı denir ve bir bilgisayar ekranında bu oran 60 veya daha fazla olmaktadır. 640x480 çözünürlüğünde ve 60 Hz tazeleme oranına sahip bir görüntüde, kontrol sinyalleriyle birlikte düşünülecek olursa bir piksel çevrimi 40 ns’de tamamlanır. Bunun için ekranı sürecektir. Bu çevrim süresini sağlayacak 25 MHz’lik bir frekansa ihtiyacı vardır.

$$t_{pclk} = \frac{1}{[(640 + 160) \times (480 + 45)] \times 60} = 40 \times 10^{-9} \quad (3.2)$$

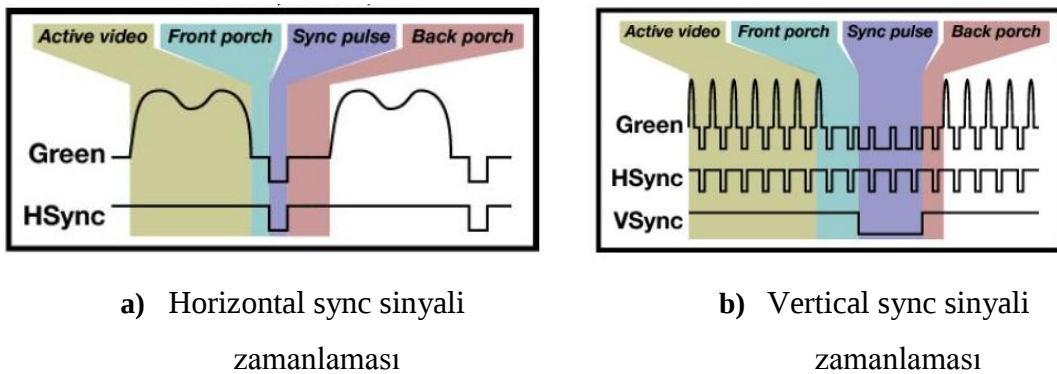
Vertical sync (Dikey eşleme) sinyali ekrana yeni bir karenin başladığını bildirmektedir. Bu sinyal alındığında tarama işlemi sol üst köşeden (0,0) itibaren başlar.

Horizontal sync (Yatay eşleme) sinyali ekrana yeni bir satırın başladığını bildirmektedir. Bu sinyal alındığında tarama işlemine bir alt satırdan devam edilir. 480 satırlık taramadan sonra yeni bir vertical sync sinyali alınır ve tarama yine üst sol köşeden itibaren devam eder. Şekil 3.5’de ekran kontrol sinyallerinin işlev ve zamanlamaları yer almaktadır.



Şekil 3.5 Ekran kontrol sinyallerinin işlev ve zamanlamaları

Bir satırın ve tüm karenin taranmasının öncesinde ve sonrasında kontrol amaçlı boş çevrimler yer almaktadır. Bu boş çevrimler esnasında ekrana vertical sync, horizontal sync sinyalleri gönderilerek görüntünün ekrana doğru basılması sağlanmış olunur. Bu boş çevrimlerin sayısı çözünürlüğe ve tazeleme oranına göre değişmektedir. Şekil 3.6’da bir kontrol sinyalinden önce ve sonraki boş çevrimler görülmektedir.



Şekil 3.6 Ekran kontrol sinyalleri boş çevrim zamanlamaları

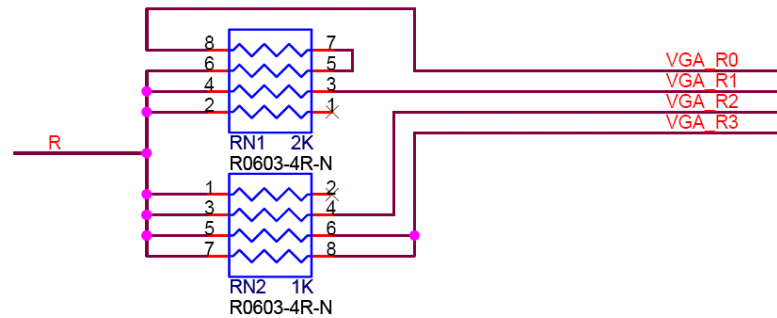
Tablo 3.1 Verilen formatlarda tarama için gerekli frekans ve tarama çevrimleri [9]

Format	Frekans (MHz)	Yatay (Piksel)				Dikey (Satır)			
		Aktif Video	Ön Boşluk	Eşleme Süresi	Arka Boşluk	Aktif Video	Ön Boşluk	Eşleme Süresi	Arka Boşluk
640x480, 60Hz	25.175	640	16	96	48	480	11	2	31
640x480, 72Hz	31.500	640	24	40	128	480	9	3	28
640x480, 75Hz	31.500	640	16	96	48	480	11	2	32
640x480, 85Hz	36.000	640	32	48	112	480	1	3	25
800x600, 56Hz	38.100	800	32	128	128	600	1	4	14
800x600, 60Hz	40.000	800	40	128	88	600	1	4	23
800x600, 72Hz	50.000	800	56	120	64	600	37	6	23

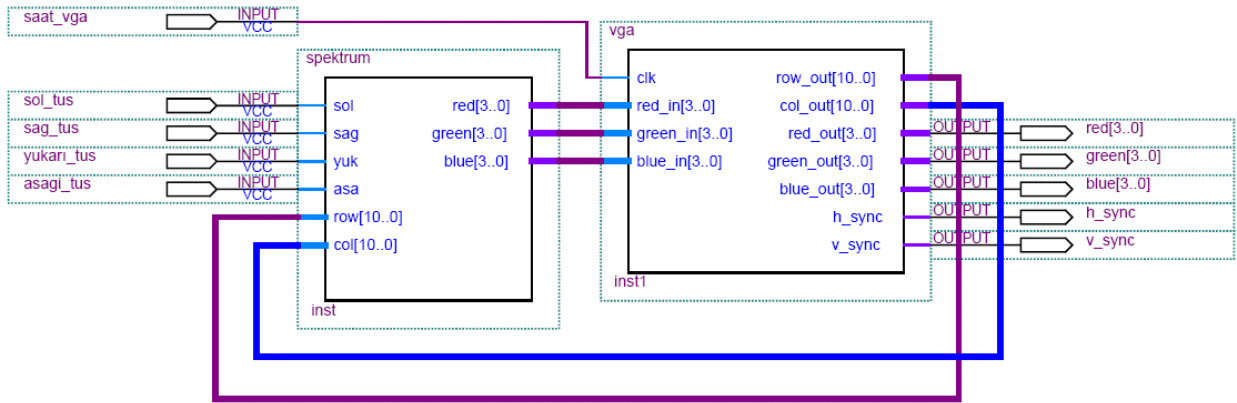
3.2.2 Ekran Kontrol Birimi Tasarımı

Projede kullanılmak üzere tasarlanan Ekran Kontrol Birimi 640x480 piksel çözünürlükte 60 Hz tazeleme oranı ile çalışacak şekildedir. Tasarlanan sistemde birkaç değişkenin değeri değiştirilerek istenilen çözünürlükte ve tazeleme oranında görüntü üretilebilmektedir.

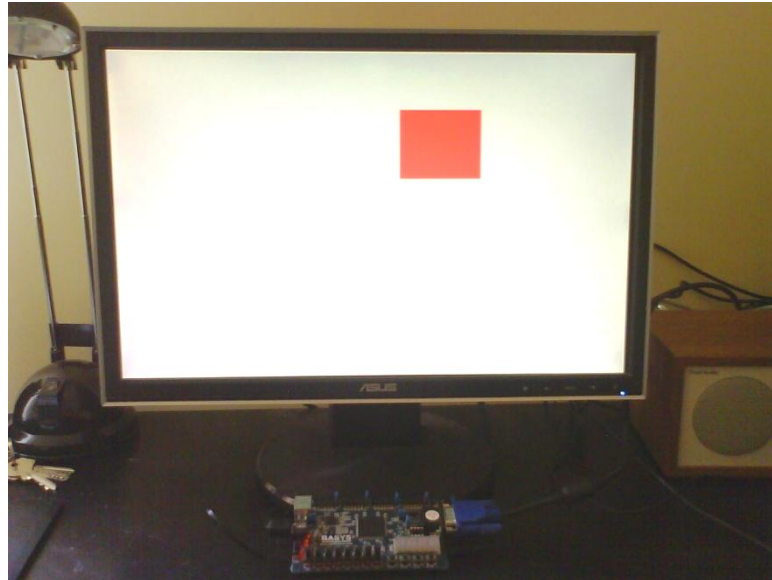
Ekran kontrol biriminin ihtiyacı olan 25 MHz saat frekansı sistemin saat frekansı ikiye bölünerek elde edilmiştir. Kontrol sinyalleriyle birlikte yatay tarama 800 çevrim dikey tarama 525 çevrimden oluşur. Bu yüzden birimde dikey ve yatay çevrimi sayan iki sayıcı tanımlanmıştır. Dikey sayıcı 0 ile 639 ve yatay sayıcı 0 ile 479 değerleri arasındayken, girişten alınan piksel bilgisi Red, Green, Blue çıkışlarına aktarılmakta diğer durumlarda bu çıkışlar lojik sıfıra çekilmektedir. FPGA uygulama geliştirme kartı üzerinde bulunan direnç ağı ile oluşturulan VGA DAC ile 4'er bitlik Red, Green, Blue sinyalleri, değeri 0 ile 0.7V arasında değişen analog sinyallere dönüştürülür.

**Şekil 3.7** Altera DE1 direnç ağı DAC'ı (Red sinyali için)

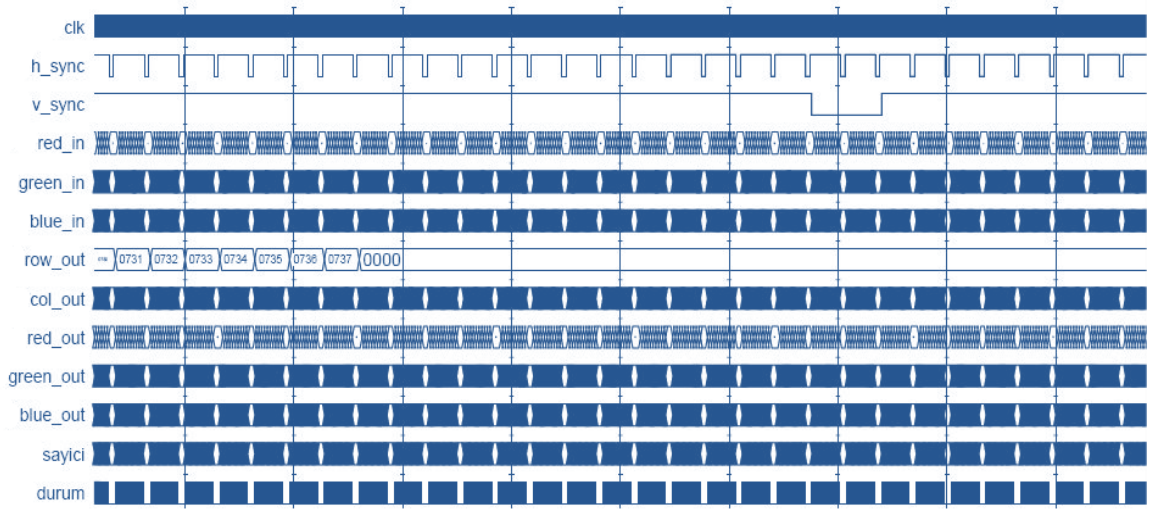
Yatay sayıcı 660 ile 756 değerleri arasında iken horizontal sync, dikey sayıcı 494 ile 495 değerlerinde iken vertical sync çıkışları lojik 0 yapılmakta diğer durumlarda bu iki çıkış da lojik 1 olarak kalmaktadır. Taranmakta olan pikselin konumu row out ve col out çıkışları ile blok dışına bildirilmektedir. Şekil 3.8’de tasarlanan ekran kontrol biriminin blok şeması yer almaktadır. Burada test amacıyla oluşturulan spektrum adlı blok, ekran kontrol biriminden aldığı konum bilgisinden yararlanarak düz beyaz ekran üzerinde kırmızı bir kare oluşturmaktadır, sol, sağ, yukarı, aşağı girişleri ile bu karenin ekran üzerindeki konumu değiştirilebilmektedir.



Şekil 3.8 Ekran kontrol birimi



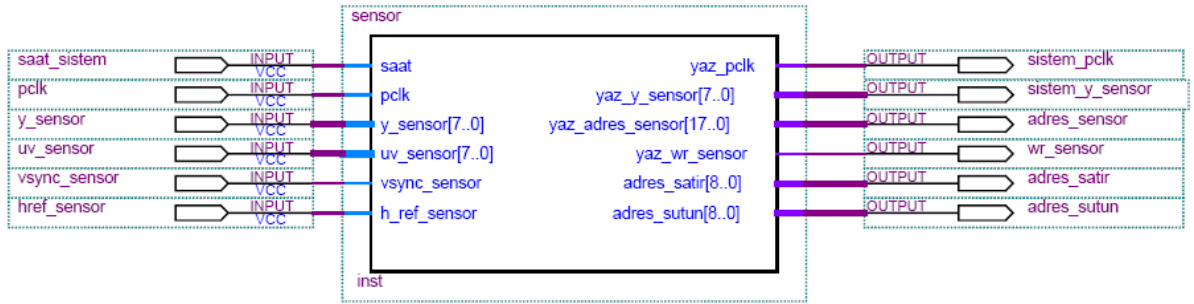
Şekil 3.9 Tasarlanan ekran kontrol biriminin uygulama geliştirme kartı üzerinde denenmesi



Şekil 3.10 Ekran kontrol biriminin simülasyon sonucu

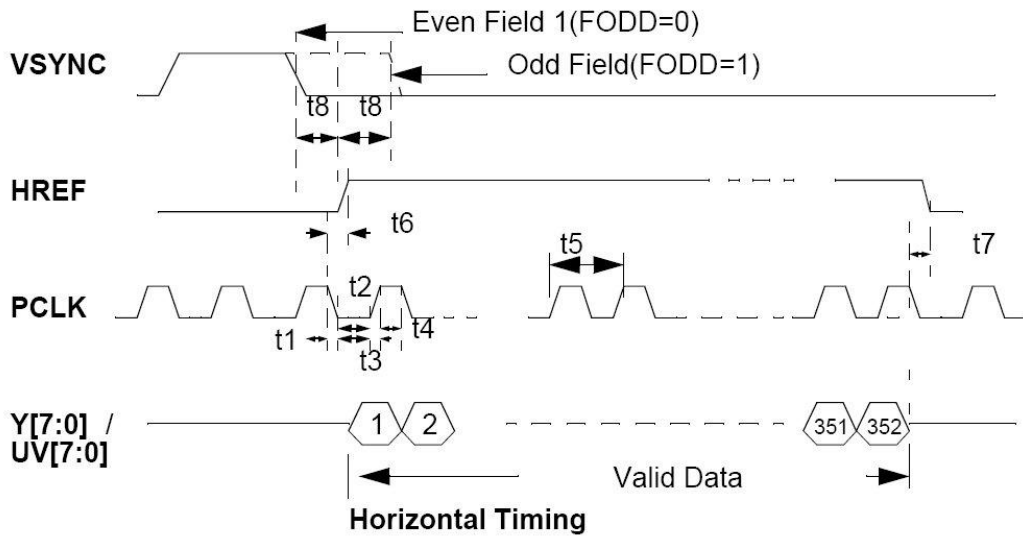
3.3 Görüntü Yakalama Birimi

Görüntü yakalama birimi sensör kartından sisteme görüntü girişini sağlamak üzere tasarlanmıştır. Görüntü sensörünün çalışması için gerekli olan saat işaretini sensör kartı üzerinde bulunan bir kristal üretmektedir. Sensör bu kristalle üretilen 17.73 MHz saat işaretini giriş katında ikiye bölerek, PCLK olarak adlandırılan 8.865 MHz frekansında sensör saat işaretini oluşturmaktadır. Dolayısıyla sensör, 50MHz frekansında çalışan ana sistemden farklı frekansta ve asenkron çalışmaktadır. Görüntü sensöründen gönderilen piksel bilgisinin sistem tarafından alınabilmesi için, sistemde PCLK girişi için atanan pin 50 MHz hızıyla taranmakta, bu tarama esnasında pinin lojik seviyesinin 0'dan 1'e çıktığı an Y ve UV girişlerindeki bilgi sisteme alınmaktadır. Yine bu tarama yapılırken pinin lojik seviyesinin 0'dan 1'e çıktığı an sistem için PCLK işaretinin yükselen kenarı, 1'den 0'a düştüğü an düşen kenarı olarak kabul edilir. Sistem tarafından içeride oluşturulan PCLK saat işareti değişken bir frekansa sahip olsa da sistem saatiyle eşzamanlı çalışmaktadır. Bu yüzden sisteme alınan Y ve UV girişlerindeki bilgi sistem tarafından sorunsuz bir şekilde kullanılabilir. Şekil 3.11'de görüntü yakalama birimi blok yapısı yer almaktadır.



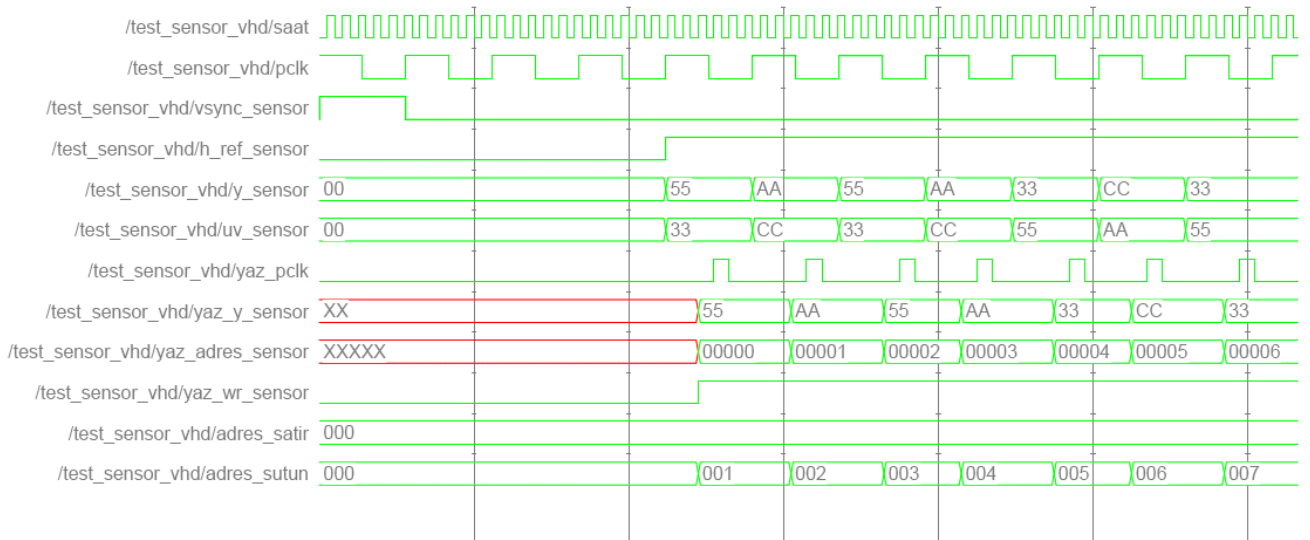
Şekil 3.11 Görüntü yakalama birimi

Sensörün PCLK saat, Y ve UV piksel bilgisi sinyalleri dışında görüntü karesinin ne zaman başladığı ve ne zaman bittiğini bildiren VSYNC ve HREF kontrol sinyalleri bulunmaktadır. Bu sinyallerden VSYNC yeni bir görüntü karesinin başladığını, HREF ise bir satırın gönderilmekte olduğunu bildirmektedir. Yeni bir görüntü karesi başlamadan VSYNC çıkışı lojik 1 olur ve bir süre bu durumda kaldıktan sonra lojik 0'a çekilir. HREF çıkışı lojik 0'dan lojik 1'e çekildiğinde satırın ilk piksel bilgisi de Y ve UV çıkışlarına verilmiş olur. Toplam 352 PCLK darbesiyle ilk satır bilgisi gönderildikten sonra HREF çıkışı tekrar lojik 0 yapılır. Bu şekilde 288 satır bilgisi gönderilerek görüntünün bir karesi tamamlanmış olur. Sensörün görüntü sinyalleri için zaman durum diyagramı Şekil 3.12'de yer almaktadır.



Şekil 3.12 Görüntü sinyalleri zaman durum diyagramı [10]

Görüntü yakalama biriminde ilk olarak gönderilmekte olan satır ve sütunun yerini belirlemek için “satır sayaç” ve “sütun sayaç” adlı iki sayıcı oluşturulmuştur. HREF sinyali lojik 1 olduğunda “sütun sayaç”ın değeri algılanan her PCLK için bir arttırılır ve görüntü verisi okunur. “Sütun sayaç”ın değeri 352’ye geldiğinde “satır sayaç”ın değeri bir arttırılır ve bu şekilde tüm piksel konumları için piksel bilgisi sisteme alınmış olur. Görüntü yakalama birimi simülasyon sonucu Şekil 3.13’de verilmiştir. Burada da görüldüğü gibi sistem içerisinde üretilen PCLK işareti düzensiz bir işarettir fakat dikkat edilirse bu işaretin yükselen kenarı ile sistem saatinin yükselen kenarı örtüşmektedir. Bu sayede alınan veri herhangi bir kayıp olmaksızın sistem içerisinde işlenebilecek durumdadır.



Şekil 3.13 Görüntü yakalama birimi simülasyon sonucu

3.4 Sensör Kontrol Birimi

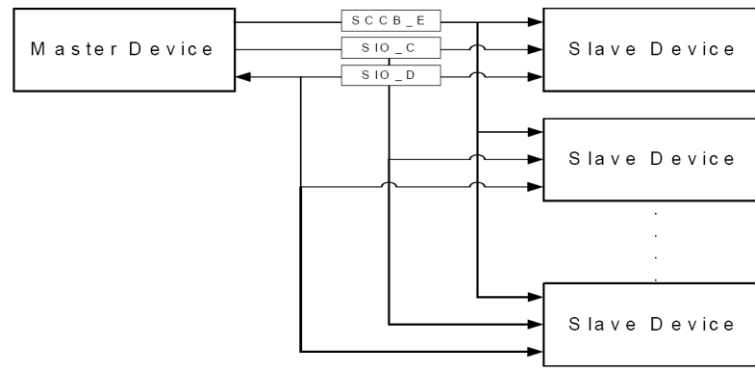
Projede görüntü sensörünün kontrolünün sağlanması için bir seri haberleşme birimi tasarlanmıştır. Bu birim sensöre özgü olan SCCB protokolünü kullanmaktadır.

3.4.1 SCCB Protokolü

SCCB OmniVision firması tarafından üretilen görüntü sensörlerinde yer alan bir I2C portudur [11]. Bu port sayesinde görüntü sensörünün diğer tümdevrelerle seri haberleşmesi sağlanır.

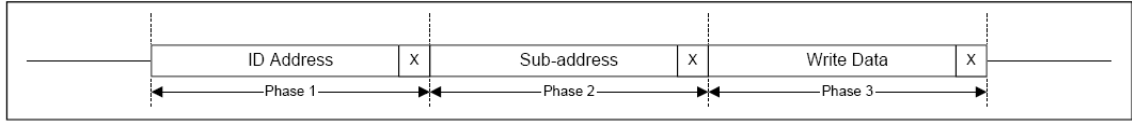
SCCB protokolünde haberleşme veriyi ve saat işaretini taşıyan iki hat ile sağlanır. Klasik I2C haberleşmesinde olduğu gibi ana (master) – uydu (slave) ilişkisi bu protokolda de bulunur.

Saat işareti ana cihaz tarafından üretilir ve veri hattı ile uydu cihaza veri yazılır ya da uydudan veri okunur. Hem saat hattı hem de veri hattı pull-up dirençleri ile besleme gerilimine çekilmektedir. Bu sayede haberleşme olmaması halinde her iki hat da lojik 1 seviyesindedir. Haberleşmeyi ana birim başlatır ve bitirir. SCCB haberleşmesinde saat hattı SIO_C, veri hattı SIO_D olarak isimlendirilir. Klasik I2C haberleşmesinden farklı olarak SCCB haberleşmesinden SIO_E olarak isimlendirilen aktifleştirme (enable) sinyali de bulunur fakat ikili haberleşmede bu sinyal kullanılmaz. SCCB haberleşmesi blok şeması Şekil 3.14’de verilmektedir.



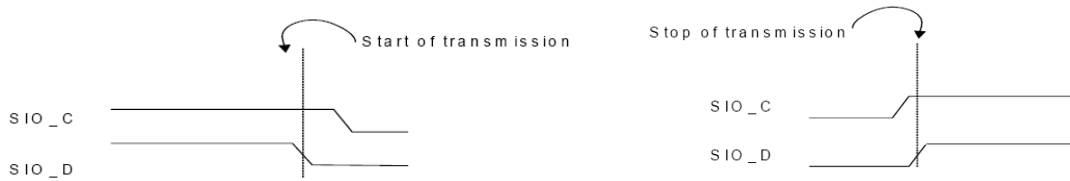
Şekil 3.14 SCCB haberleşmesi blok şeması

SCCB haberleşme protokolü 3 veya 2 baylık bir koddan meydana gelir. Eğer yazma yapılacaksa gönderilecek kod genelde 3 bayttan meydana gelir. İlk bayt haberleşilecek uydu cihazın kimliği, ikinci bayt bu uydu cihazda değeri değiştirilecek kaydedicinin adresi, üçüncü bayt ise yazılacak veridir. Eğer okuma yapılacaksa gönderilecek kod 2 bayttan meydana gelir. Bu durumda ilk bayt haberleşilecek uydu cihazın kimliği, ikinci bayt ise bu uydu cihazda okuma yapılacak kaydedicinin adresidir. Okuma işleminde üçüncü baytı uydu cihazdan ana cihaza gönderilen veri oluşturur. Her gönderilen bayttan sonra 1 bitlik kontrol biti gönderilir. Bu bit önemsizdir (don't care). Şekil 3.15’de SCCB yazma protokolü yer almaktadır.



Şekil 3.15 SCCB yazma protokolü

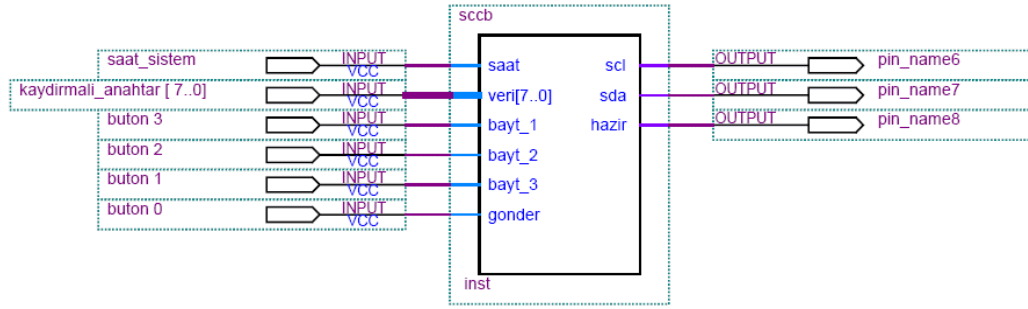
SCCB haberleşmesinde iletim başlamadan önce yani hat boştayken SIO_C saat ve SIO_D veri sinyalleri lojik 1 durumundadır. İletimin başlaması için önce SIO_D veri sinyali sonra SIO_C saat sinyali lojik 0'a çekilir ve SIO_C saat sinyalinin yükselen kenarıyla beraber ilk baytın ilk biti gönderilir. İletim biterken ise üçüncü baytın önemsiz biti gönderildikten sonra SIO_D veri sinyali lojik 0 yapılır. SIO_C saat sinyali lojik 1 yapıldıktan sonra SIO_D veri sinyalinin tekrar lojik 1 yapılmasıyla iletim sonlandırılmış olur. 3.16'de iletimin başlama ve bitmesi esnasında hatların sinyal seviyelerinin değişimi görülmektedir.



Şekil 3.16 SCCB protokolünde iletimin başlaması ve sonlanması

3.4.2 Sensör Kontrol Birimi Tasarımı

Projede görüntü sensörünün beyaz dengesi, kazanç, parlaklık gibi görüntü özelliklerini değiştirmek amacıyla SCCB seri haberleşme hattı kullanılmıştır. Sensörün kaydedici içeriği okunmaya gerek duyulmadığından, birim sadece yazma özelliğini kullanır. FPGA uygulama geliştirme kartı üzerinde bulunan 8 kaydırmalı anahtar ve 3 butonla gönderilecek veri ayarlandıktan sonra dördüncü buton ile veri gönderilir. Verinin gönderimi seri haberleşme biriminde veri gönderimine benzer şekilde olduğu için oluşturulan algoritma da çok farklı olmamıştır. Yalnızca burada bilgi 3'er baytlık çevrimlerle gönderilmekte, başlatma ve sonlandırma durumları yukarıda anlatıldığı şekilde olmaktadır. Sensör kontrol birimi blok şeması 3.17'de yer almaktadır.



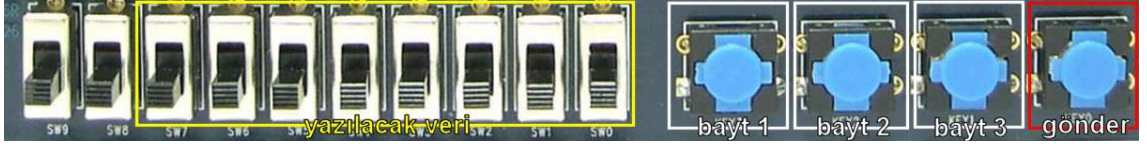
Şekil 3.17 Sensör kontrol birimi

Bilgi gönderilmeden önce birim içerisinde yer alan bayt 1, bayt 2, bayt 3 kaydedicilerinin doldurulması gereklidir. Kullanıcı buton 3, buton 2 ve buton 1'e basarak, kaydırmalı anahtarlar üzerinde oluşturulan ikilik değeri kaydedicilere yüklemiştir. Buton 0'a basıldığında ise veri ilgili protokole uygun olarak sensöre gönderilir. Sistem bir veri gönderiliyorken başka bir verinin gönderimine izin vermeyecektir. Buton 3'e basıldığında bayt 1 kaydedicisine yüklenen veri cihaz kimliği olacaktır. OV6620 görüntü sensörünün birden çok cihaz kimliği bulunabilir. Cihazın kimliği sensör üzerinde bulunan 3 CS pini ile belirlenmektedir. Bu kimliklerin son bitinin değerine göre sensör okuma veya yazma yapılacağını anlar. Eğer kimliğin son biti 0 ise yazma, 1 ise okuma yapılacağı anlamına gelir. Sensör kimlik değerleri Tablo 3.2'de verilmektedir.

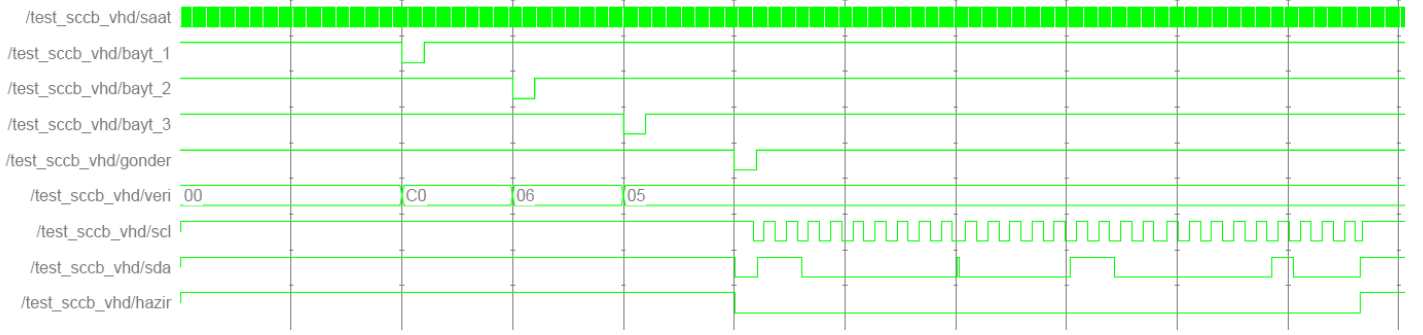
Tablo 3.2 Seçilen CS değerlerine göre sensörün kimlik değerleri

CS[2:0]	000	001	010	011	100	101	110	111
WRITE ID (hex)	C0	C4	C8	CC	D0	D4	D8	DC
READ ID (hex)	C1	C5	C9	CD	D1	D5	D9	DD

Sensöre yazma yapılacağından bayt 1 kaydedicisi "C0" olarak atanmalıdır. (Sensörün CS değeri '000' olarak ayarlanmıştır) Buton 2'ye basıldığında bayt 2 kaydedicisine yüklenen veri sensörde değiştirilmek istenen kaydedicinin adresi olacaktır. Buton 1'e basıldığında bayt 3'e yüklenen veri ise sensörde kaydedicilere yazılacak veri olacaktır. Sensör kontrol biriminin simülasyonu Şekil 3.19'da verilmektedir.



Şekil 3.18 FPGA uygulama geliştirme kartı üzerinde anahtar tanımlamaları



Şekil 3.19 Sensör kontrol birimi simülasyon sonucu

3.5 Konvolüsyon Birimi

Projede görüntü sensöründen alınan görüntü üzerinde filtreleme işlemlerini gerçekleştirmek üzere konvolüsyon birimi tasarlanmıştır. Bu birim görüntü üzerinde 2 boyutlu konvolüsyon işlemi gerçekleştirmektedir. Görüntü üzerinde konvolüsyon işleminin matematiksel ifadesi Eşitlik 3.3’de verilmektedir.

$$g(x, y) = \sum_{j=-n}^n \sum_{i=-m}^m k(i, j) \cdot f(x - i, y - j) = k * f \quad (3.3)$$

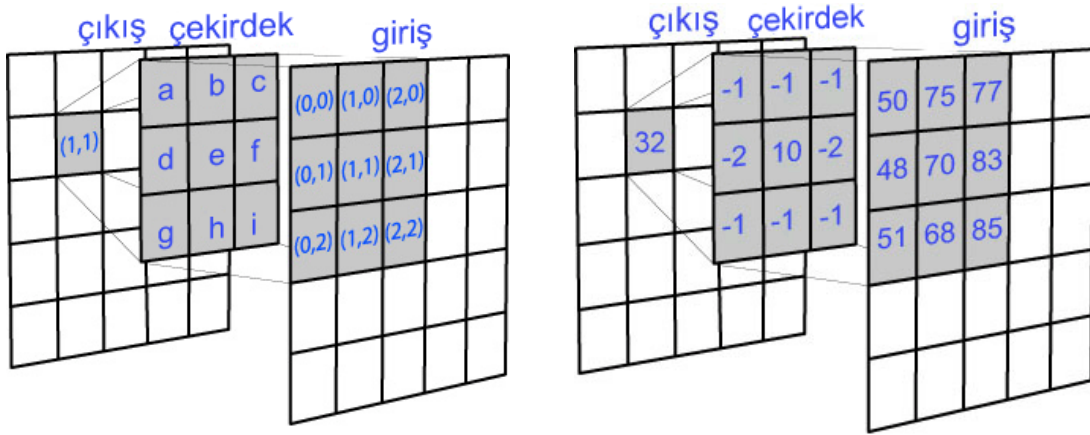
$$m = \frac{w - 1}{2} \quad n = \frac{h - 1}{2} \quad \begin{array}{ll} k: \text{konvolüsyon kerneli} & f: \text{işlenecek imge} \\ w, h: \text{imge boyutları} \end{array}$$

Konvolüsyon, yumuşatma, keskinleştirme, kenar belirleme gibi görüntü işleme fonksiyonlarını gerçekleştirmede çok sık kullanılmaktadır. Konvolüsyonda bir pikselin yeni değeri kendisinin ve çevresindeki piksellerin ağırlıklı ortalaması ile bulunmaktadır. Piksellerin ağırlıkları konvolüsyon şablonu (kerneli) olarak adlandırılan bir matris ile belirlenir. Konvolüsyon şablonu uygulamaya göre farklı boyutlarda olabilmekle beraber genelde 3x3 lük bir matristir. Şekil 3.20’de görüntü üzerindeki bir piksel için konvolüsyon işlemi görülmektedir. Şablon matris ve matrisin üzerinde bulunduğu

görüntü matrisinin konvolüsyonu sonucu elde edilen piksel değeri Eşitlik 3.4'de verilmektedir. Bu işlem görüntü üzerinde tüm piksellere uygulanarak görüntü üzerinde 2 boyutlu konvolüsyon işlemi yapılmış olur.

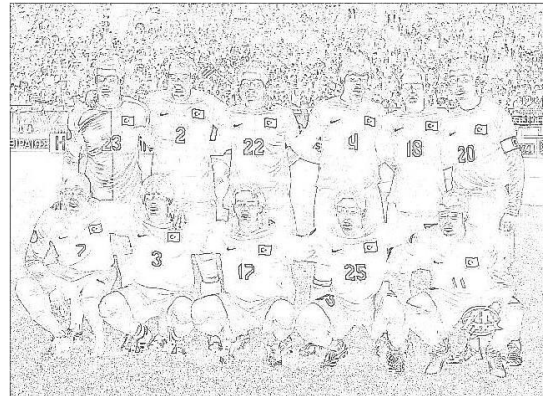
$$imge = \begin{bmatrix} i(0,0) & i(0,1) & i(0,2) \\ i(1,0) & i(1,1) & i(1,2) \\ i(2,0) & i(2,1) & i(2,2) \end{bmatrix} = \begin{bmatrix} 55 & 75 & 77 \\ 48 & 70 & 83 \\ 51 & 68 & 85 \end{bmatrix} \quad çekirdek = \begin{bmatrix} -1 & -1 & -1 \\ -2 & 10 & -2 \\ -1 & -1 & -1 \end{bmatrix}$$

$$i(1,1) = (-1 \times 55) + (-1 \times 75) + (-1 \times 77) + (-2 \times 48) + (10 \times 70) + (-2 \times 83) + (-1 \times 51) + (-1 \times 68) + (-1 \times 85) = 32 \quad (3.4)$$



Şekil 3.20 Görüntü üzerinde konvolüsyon işlemi

$$laplace = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$



Şekil 3.21 Görüntü ile laplace matrisinin konvolüsyon sonucu

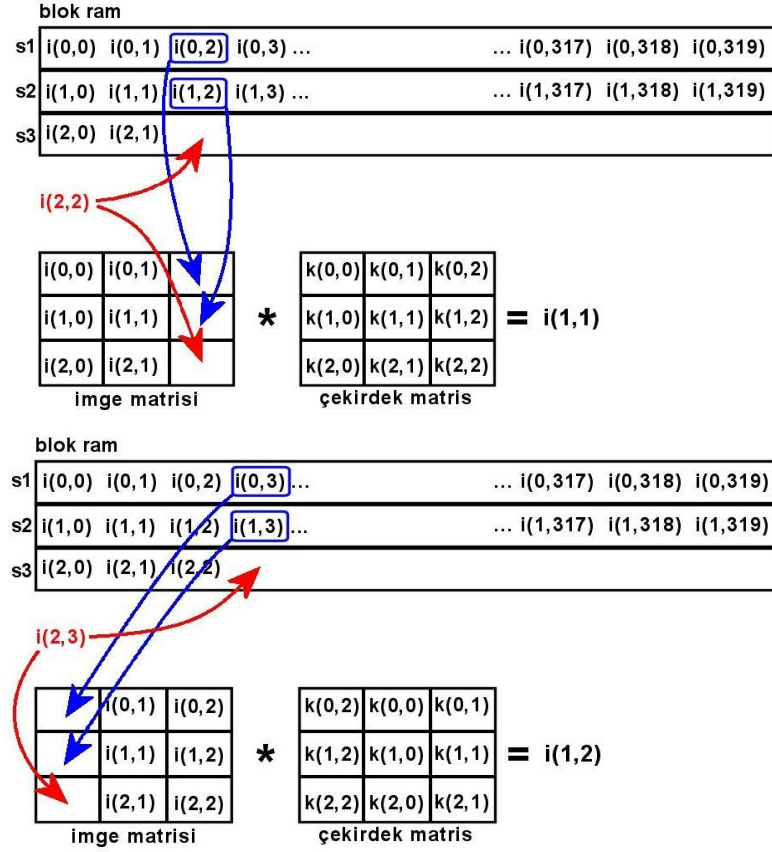
3.5.1 Konvolüsyon Birimi Tasarımı

Görüntü üzerinde filtreleme işlemleri yapmak ve özellikle kenar belirlemek amacıyla tasarlanan konvolüsyon birimi 3x3 boyutundaki bir şablon matris kullanmaktadır. Birimde 8 bit renk derinliğine sahip gri formattaki görüntüler üzerinde konvolüsyon yapılmaktadır. Her saat çevriminde bir pikselin değeri hesaplanmak istenirse, şablon matris 3x3 boyutlarda seçildiğinden dolayı değeri hesaplanacak pikselle birlikte o piksele komşu 8 pikselin de değerinin bilinmesi gereklidir. Bu yüzden sistemde görüntünün 3 satır bilgisini tutacak boyutta bir blok ram bulunmaktadır. Dışarıdan birime gelen piksel verisi bu blok ram'a sırasıyla yazılmakta, blok ram'da adres sonuna gelindiğinde bu yazma işlemi tekrar ilk adresten itibaren başlamaktadır. Görüntünün $i(2,2)$ piksel bilgisi blok ram'a yazıldığında ilk konvolüsyon yapılır ve yeni $i(1,1)$ değeri çıkış olarak verilir.



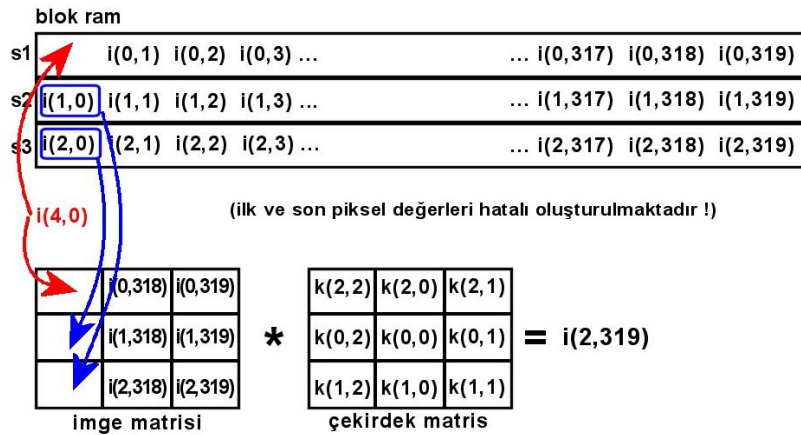
Şekil 3.22 Gelen piksel verilerinin blok ram'a yazılması

Konvolüsyonu gerçekleştirmek için birimde 9 tane 9x9 bitlik çarpıcı bulunmaktadır. Bu çarpıcılarda dokuz çarpımın aynı anda yapılabilmesi için girişlerinde 9 bayt şablon ve 9 bayt görüntü matrisinin hazır bulunması gerekmektedir. Bundan dolayı çarpılacak 9 görüntü pikselini ve şablonu saklayan kaydediciler oluşturulmuştur. Bu kaydedicilerden görüntü piksellerini saklayan görüntü matrisine her saat çevriminde ikisi blok ram'dan biri dışarıdan olmak üzere toplam 3 piksel değeri sütun sütun yazılır. Görüntü matrisindeki görüntünün sütun sıralamasına göre şablon matrisindeki sütun sıralaması da değiştirilmektedir. Şekil 3.23'de konvolüsyon birimine gelen bir piksel için kaydedicilerde meydana gelen değişiklikler görülmektedir.



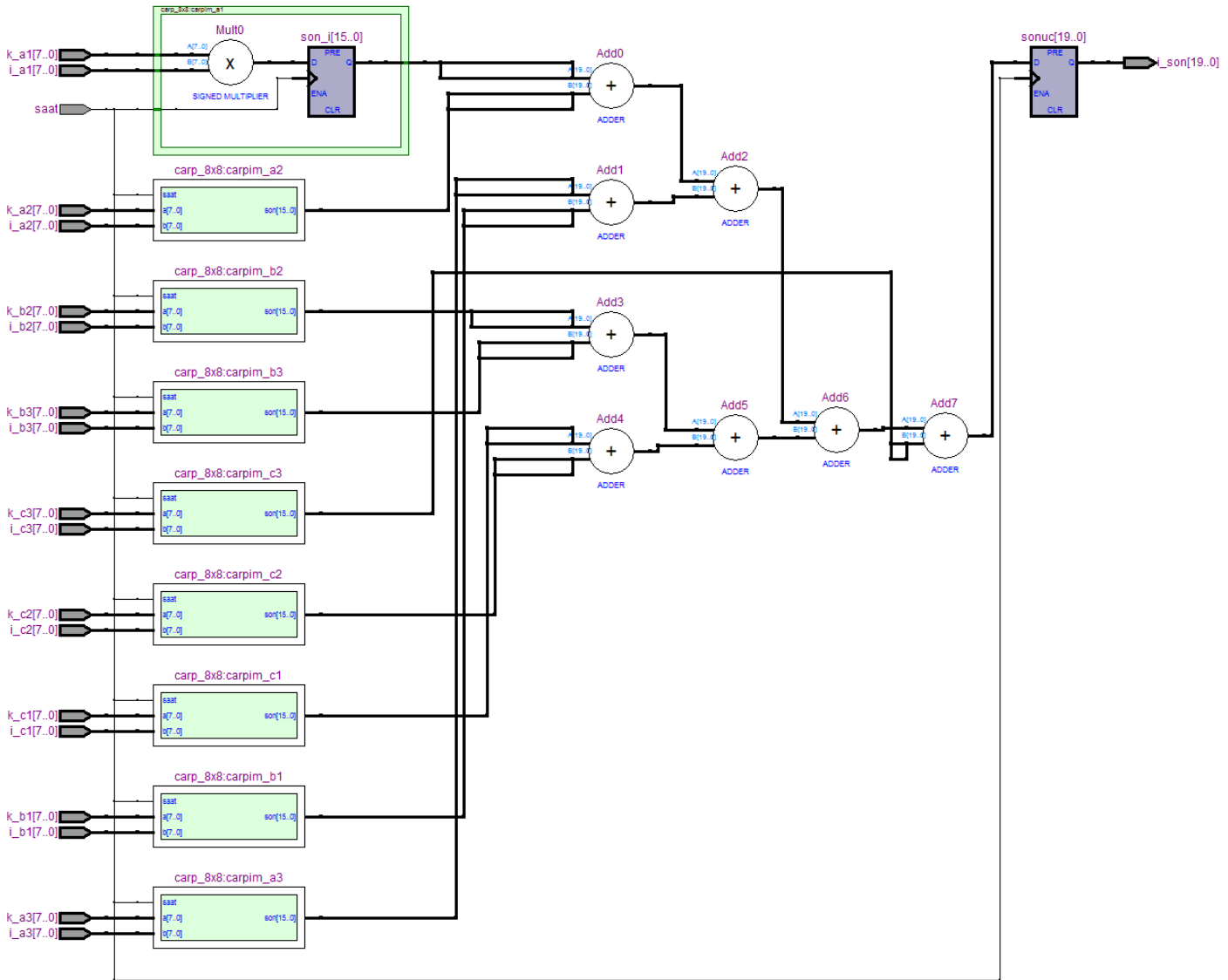
Şekil 3.23 Görüntünün i(2,2) ve i(2,23) piksel bilgileri geldiğinde kaydedicilerin durumu

Görüntünün i(4,0) piksel bilgisinin gelmesiyle sistem biraz daha karmaşık hale gelmektedir. Gelen bu piksel bilgisi görüntü matrisinde ilk satıra, blok ram'dan gelen iki piksel bilgisi ise ikinci ve üçüncü satıra yazılır. Şablon matrisindeki satır sıralaması da buna bağlı olarak değişir.



Şekil 3.24 Görüntünün i(4,0) piksel bilgisi geldiğinde kaydedicilerin durumu

Oluşturulan algoritma ile her saat darbesinde bir piksel değerinin hesaplanabilmesi mümkün olmuştur. Birimde konvolüsyon şablonu sabit olmayıp dışarıdan alınmaktadır. Bu nedenle 3x3 lük bütün filtreleme işlemleri görüntü üzerine uygulanabilir. Konvolüsyon biriminde çarpımlar toplamını yapan alt birimin blok şeması Şekil 3.25’de yer almaktadır. Bu birimde kapı gecikmesi 9.22 ns olmaktadır. Böylece konvolüsyon birimi 100 MHz ve üstü frekanslarda çalışabilmektedir. Ek 1’de konvolüsyon birimi blok şeması yer almaktadır.



Şekil 3.25 Konvolüsyon işlem alt birimi blok şeması

4 SİSTEMLERİN GERÇEKLENMESİ

Projenin devamı esnasında tasarlanan birimlerle ara sistemler oluşturulmuş, oluşturulan bu sistemler sentezlenerek FPGA uygulama geliştirme kartı üzerinde fiziksel olarak gerçekleştirilmiştir. Böylece tasarlanan birimlerin bir sistem içinde ne kadar doğru çalıştığı tespit edilerek hatalar düzeltilmeye çalışılmıştır. İlk tasarlanan sistemde görüntü verisi seri port üzerinden alınıp konvolüsyon sonucu yine seri port ile gönderilmektedir. İkinci sistemde de görüntü verisi seri port üzerinden alınmakta fakat konvolüsyon sonucu ekran üzerinde görüntülenmektedir. Son tasarlanan sistemde ise görüntü verisi CMOS görüntü sensöründen alınmakta, konvolüsyon sonucu ekran üzerinde görüntülenmektedir.

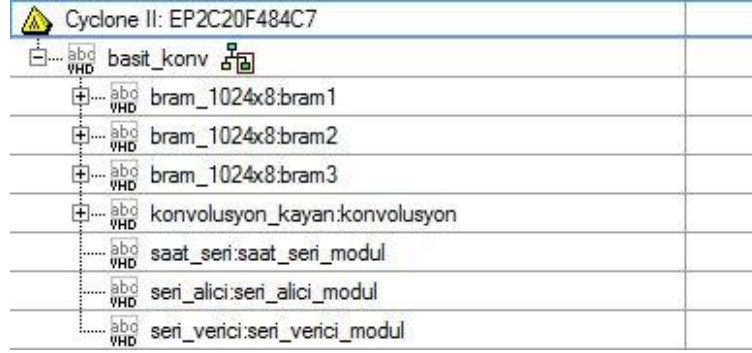
4.1 Seri Port Kaynaklı Basit Konvolüsyon Sisteminin Gerçeklenmesi

Bu sistemde görüntü verisi seri port ile bilgisayar üzerinden alınmakta, üzerinde konvolüsyon işlemi yapıp sonuç tekrar seri port ile bilgisayara gönderilmektedir. Görüntü verisi seri port ile alındığından sistemin hızını seri port'un hızı belirlemektedir. Veri FPGA geliştirme kartına MATLAB aracılığı ile gönderilmektedir. Görüntü verisi gönderilmeden önce 14 baytlık kontrol verisi gönderilir. Bu verinin ilk baytı başla komutu, sonraki 9 baytı konvolüsyon şablonu, son 4 baytı ise gönderilecek görüntünün boyutlarıdır. Sisteme veriyi gönderen ve sonucu alan MATLAB kodu Ek 2'de verilmektedir. Şekil 4.1'de konvolüsyon kontrol verisi yer almaktadır.

01	255	255	255	255	08	255	255	255	255	00	239	01	63
başla	konvolüsyon şablonu									boyut - y		boyut - x	

Şekil 4.1 Konvolüsyon kontrol verisi

Komut verisinden sonra görüntü verisi sisteme gönderilir. Sistem her aldığı bir piksel bilgisinde, bir piksel bilgisi üretip bu sonucu PC'ye yollar. Gönderilen konvolüsyon şablonu işaretli sayı olarak algılanır. Görüntü boyutları için ikişer bayt ayrılmıştır. Gönderilecek piksel bilgisi 8 bit derinliğinde gri formatta olmalıdır. Sistemde seri haberleşme birimi ve konvolüsyon birimi kullanılmıştır. Sistem hiyerarşisi Şekil 4.2'de verilmiştir.

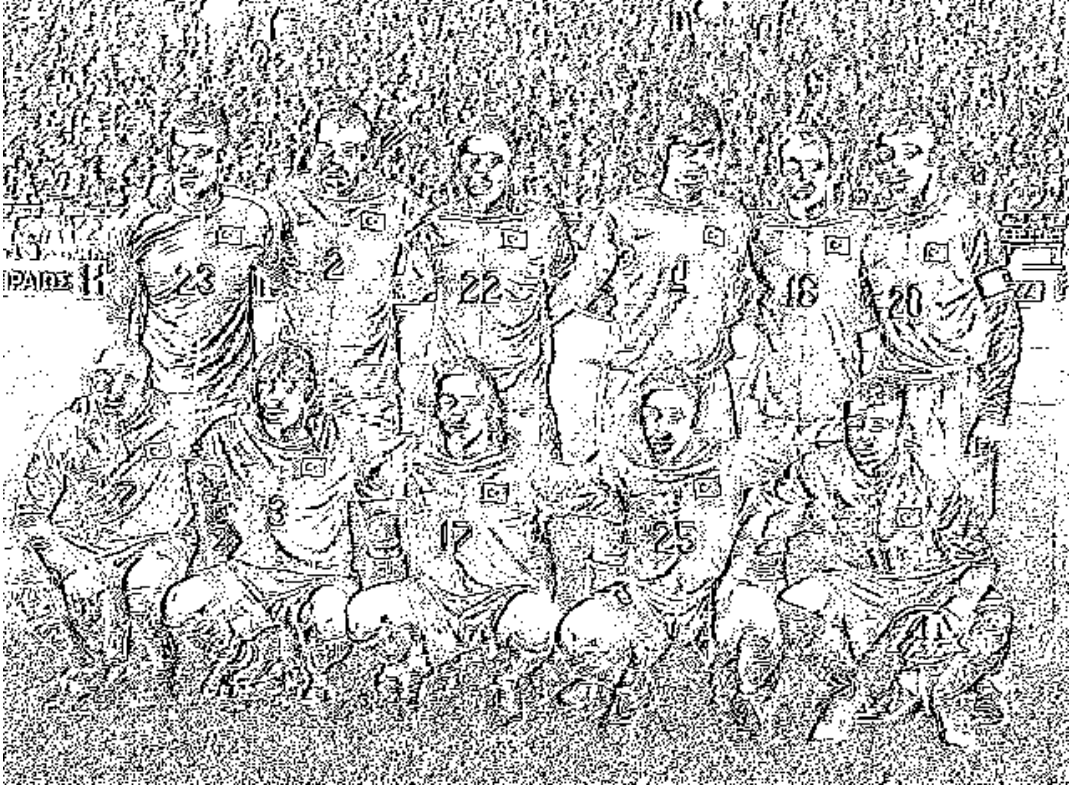


Şekil 4.2 Seri port kaynaklı basit konvolüsyon sistemi hiyerarşisi

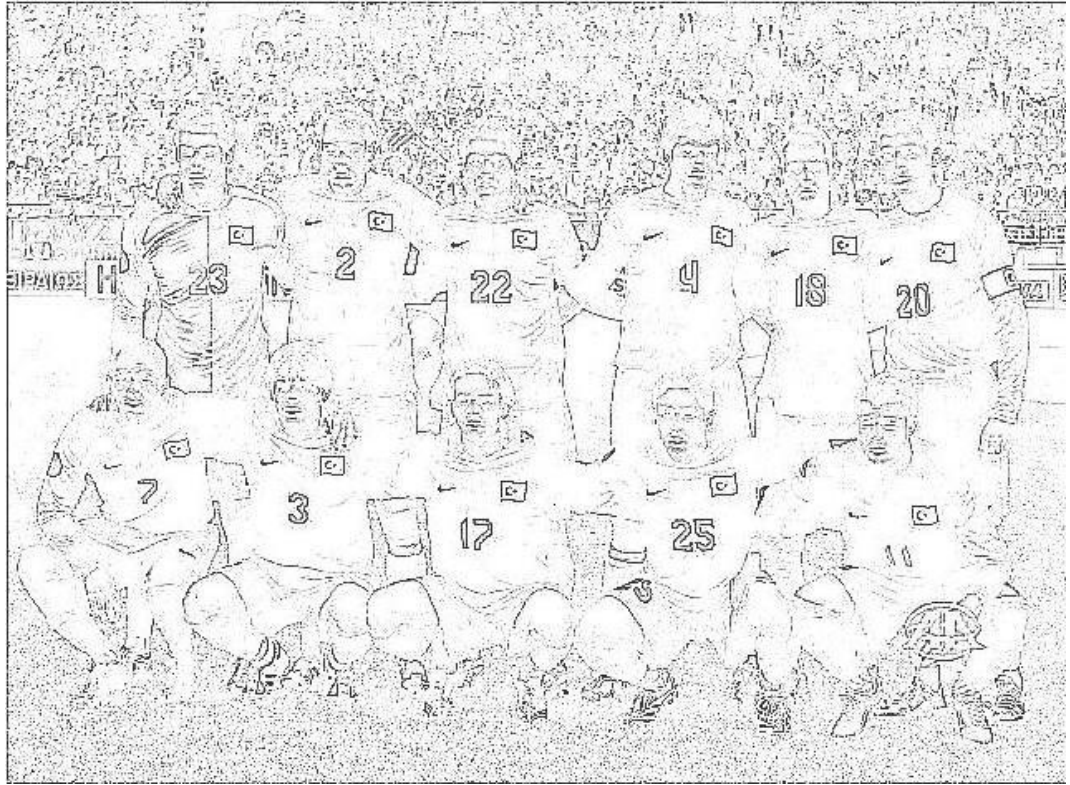
Şekil 4.3’de sisteme gönderilen görüntü yer almaktadır. Bu veri laplace matrisi ile konvolüsyondan geçirilmiştir. Şekil 4.4 ve Şekil 4.5’de sistemden elde edilen sonuç ile MATLAB üzerinde elde edilen sonuç yer almaktadır. Alınan görüntü üzerinde, bilgisayar seri port ön belleğinde meydana gelen yığılmadan kaynaklanan bozulmalar mevcuttur. İletişim hızı düşürülerek bu problem yüksek oranda çözülebilir. FPGA kaynaklarının kullanımı Tablo 4.1’de yer almaktadır.



Şekil 4.3 Seri port kaynaklı konvolüsyon sistemine gönderilen görüntü



Şekil 4.4 Seri port kaynaklı konvolüsyon sisteminden alınan görüntü
(Laplace matrisi ile konvolüsyondan geçirilmiştir)



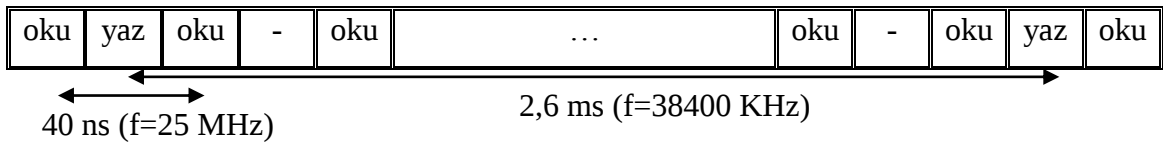
Şekil 4.5 MATLAB üzerinde yapılan konvolüsyon sonucu

Flow Status	Successful - Wed May 20 12:10:00 2009
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	adim4
Top-level Entity Name	basit_konv
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	Yes
Total logic elements	1,028 / 18,752 (5 %)
Total combinational functions	1,018 / 18,752 (5 %)
Dedicated logic registers	326 / 18,752 (2 %)
Total registers	326
Total pins	12 / 315 (4 %)
Total virtual pins	0
Total memory bits	21,504 / 239,616 (9 %)
Embedded Multiplier 9-bit element	18 / 52 (35 %)
Total PLLs	0 / 4 (0 %)

Tablo 4.1 İlk sistem için FPGA kaynaklarının kullanımı

4.2 Seri Port Kaynaklı Ekran Kontrollü Basit Konvolüsyon Sisteminin Gerçeklenmesi

İlk sistemden ayrı olarak bu sistemde üretilen sonuç ekran üzerinde görüntülenmektedir. Sistemde seri port hızı 38400 baud/s olarak ayarlanmıştır. Ekran kontrol birimi ise 50 Hz tazeleme oranında 640x480 çözünürlüğünde 25 MHz saat frekansı ile çalışmaktadır. Burada gelen veri ile gönderilmesi gereken veri hızlarında çok büyük bir uçurum bulunmaktadır. Dolayısıyla gelen verinin geldiği gibi işlenip ekranın sürülmesi imkansızdır. Bu nedenle sistemde ekran kontrol birimi, verisini FPGA uygulama geliştirme kartı üzerinde bulunan SRAM üzerinden alacak şekilde tasarlanmıştır. Kart üzerinde bulunan SRAM tek portlu olduğundan dolayı konvolüsyon sonucunun SRAM'e yazılmasıyla, ekrana sürülecek verinin SRAM'dan alınması belirli bir sıra ile olmaktadır. Ekran kontrol birimi 25 MHz'de çalıştığından dolayı SRAM kontrol ara biriminin saat frekansı 50Mhz olarak ayarlanmıştır. Böylece her iki saat darbesinin birinde SRAM'dan okuma yapılmaktadır. SRAM'a yazma işlemi ise konvolüsyon sonucu değer üretildiğinde olmaktadır. Şekil 4.6'da SRAM'a yazma ve SRAM'dan okuma zamanlaması yer almaktadır.



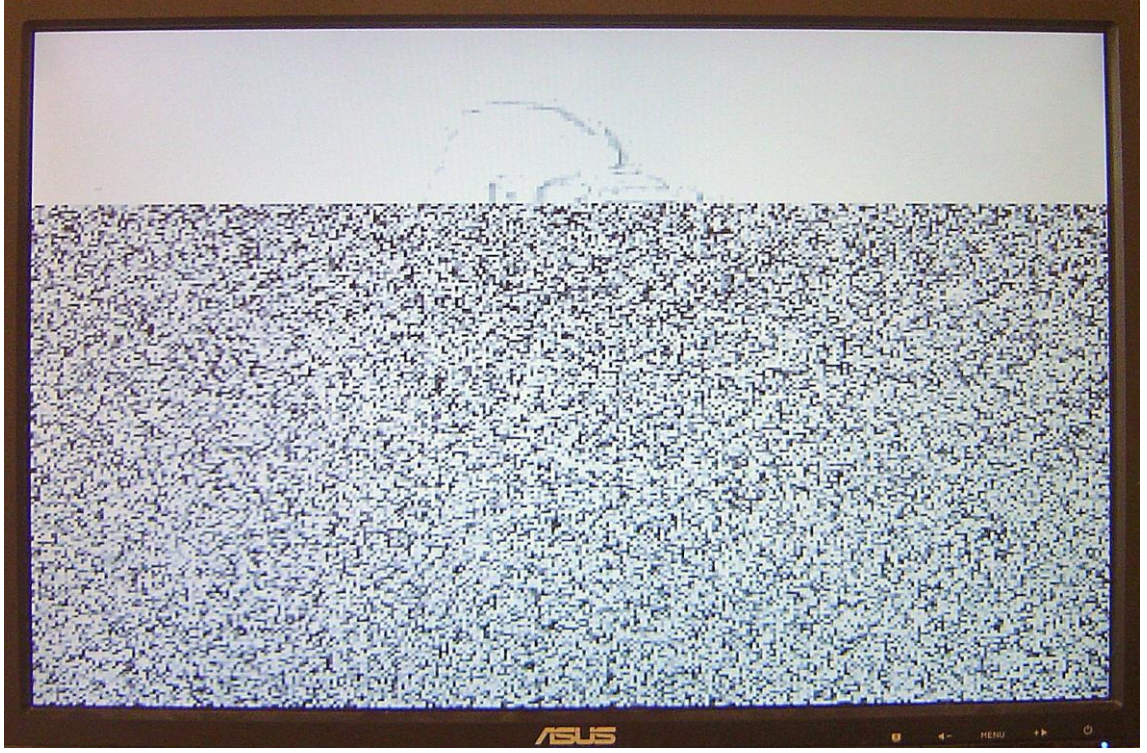
Şekil 4.6 SRAM'a yazma ve SRAM'dan okuma zamanlaması

Sistemde ekran kontrol birimi SRAM'daki veriyi saniyede 50 kere ekrana gönderir. Konvolüsyon işleminin hızı seri portun hızına bağlı olduğundan bir karenin konvolüsyonu yaklaşık 20 saniye sürmektedir. Bu nedenle konvolüsyon işleminin yapılışı ekranda izlenebilir. Şekil 4.7'de sisteme gönderilen görüntü yer almaktadır. Bu veri laplace matrisi ile konvolüsyondan geçirilmiştir. Şekil 4.8'de konvolüsyon yapılırken, Şekil 4.9'da konvolüsyon sonunda ekran durumları görülmektedir.

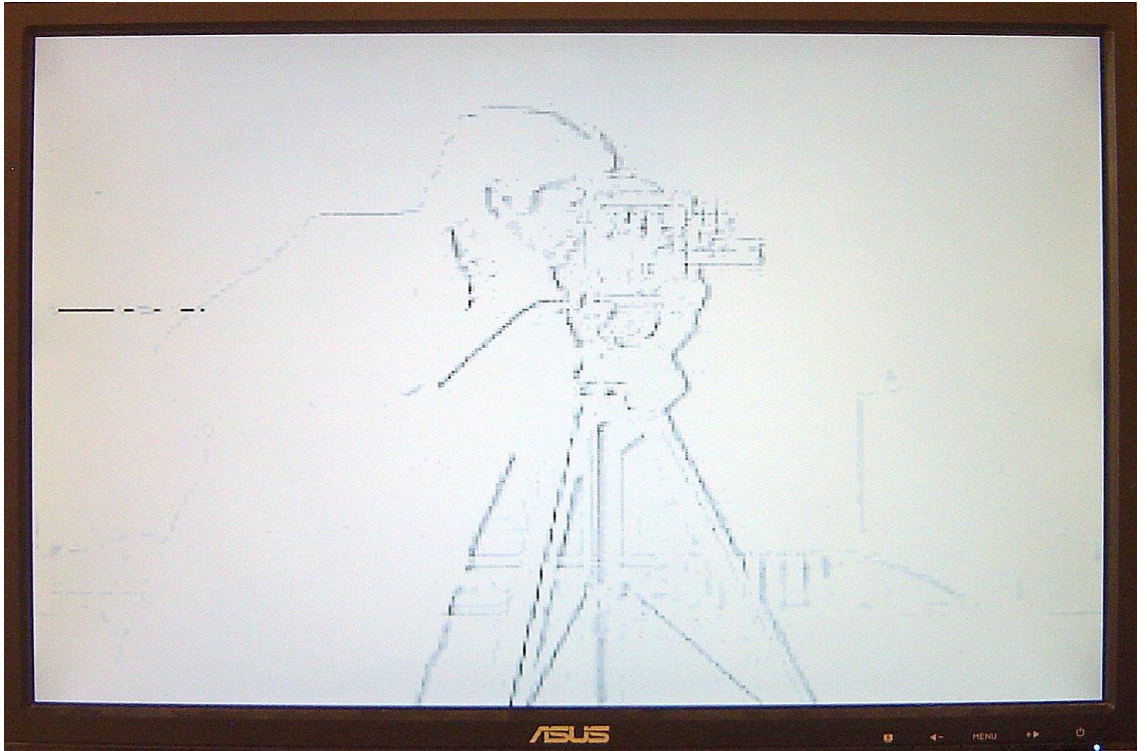
$$t_{konv} = \frac{(320 \times 240) \times 8 \text{ bit}}{\frac{38400 \text{ bit/sn}}{10 \text{ bit}} \times 8 \text{ bit}} = 20 \text{ sn} \quad (5.1)$$



Şekil 4.7 Sisteme gönderilen görüntü



Şekil 4.8 Konvolüsyon işlemi yapılırken ekran durumu



Şekil 4.9 Konvolüsyon sonucu

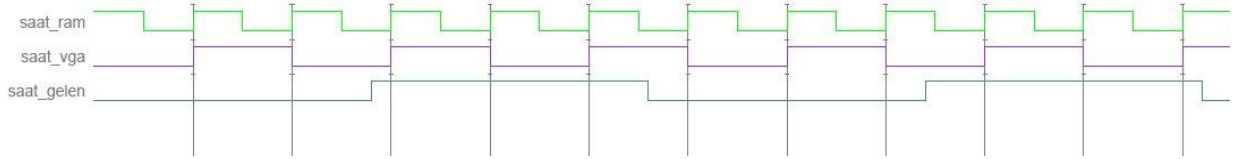
Flow Status	Successful - Wed May 20 02:36:19 2009
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	adim_5
Top-level Entity Name	basit_konv
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	No
Total logic elements	1,360 / 18,752 (7 %)
Total combinational functions	1,237 / 18,752 (7 %)
Dedicated logic registers	633 / 18,752 (3 %)
Total registers	633
Total pins	72 / 315 (23 %)
Total virtual pins	0
Total memory bits	21,504 / 239,616 (9 %)
Embedded Multiplier 9-bit element	18 / 52 (35 %)
Total PLLs	0 / 4 (0 %)

Tablo 4.2 İlk sistem için FPGA kaynaklarının kullanımı

4.3 Ana Sistemin Gerçeklenmesi

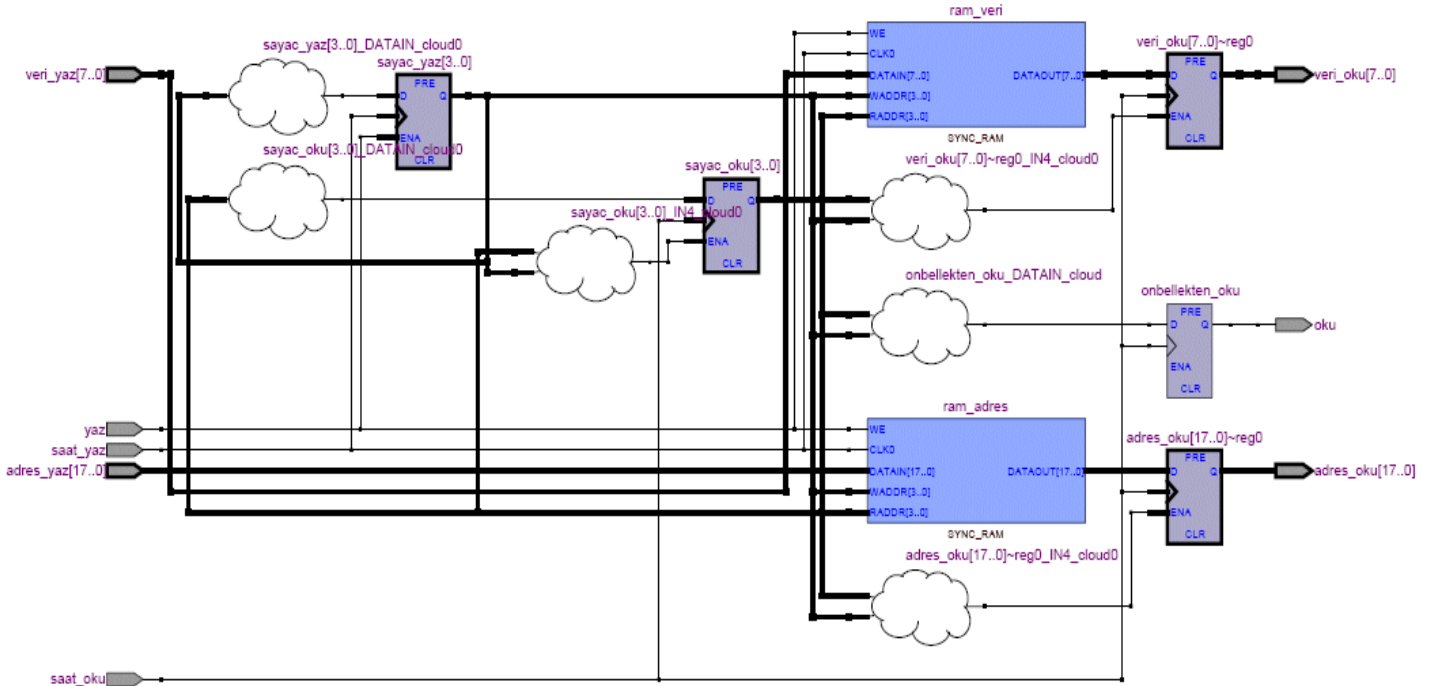
Proje seyri boyunca oluşturulan ara sistemlerle, tasarlanan birimlerin birbirleriyle uyumu gözlenmiş ve karşılaşılan uyumsuzluklar büyük ölçüde giderilmiştir. Böylelikle yapılan yanlış tasarımların çok geç olmadan fark edilmesi mümkün olmuştur. Tasarlanan bu sistem projenin mihenk taşı olarak kabul edilebilir. Daha önce tasarlanan bütün birimlerin kullanıldığı bu sistem, bilgisayardan tamamen bağımsız çalışabilen bir SoC (System-on-a-Chip) olarak kabul edilebilir. Sistem, görüntü sensöründen alınan veri üzerinde konvolüsyon işlemi yapmakta ve konvolüsyonun sonucunu VGA ekran üzerinde görüntülemektedir. Sistemde bilgisayara olan bağımlılık, konvolüsyon şablon matrisinin bilgisayar üzerinden seri port ile değiştirilebiliyor olmasından kaynaklanmaktadır. Aslında bu bağımlılık sisteme esneklik sağlamakta, konvolüsyon şablonunun istenilen an değiştirilebilmesini mümkün kılmaktadır. Her sistem tasarımında olduğu gibi bu sistem de tasarlanırken çeşitli zorluklarla karşılaşmıştır. En büyük zorluk görüntü sensörünün farklı frekansta ve asenkron çalışmasından dolayı yaşanmıştır. Görüntü yakalama birimi tasarımında daha önce karşılaşılan bu durum bir bakıma örnekleme yapılarak çözülmüştür. Bu sistemin tasarımındaki problem ise SRAM'a yazma sırasında meydana gelmektedir. Daha önce tasarlanan sistemlerde SRAM'a yazılacak verinin saat frekansı SRAM ve VGA kontrol ara biriminin saat frekansından farklı olsa da, üç işaret de aynı harici kristal ile üretildiğinden birimler sıralı çalışacak şekilde ayarlanabilmiştir.

Oysa görüntü sensörü saat frekansını, görüntü sensör kartı üzerindeki kristalden almaktadır. Bu sorun SRAM ile yazılacak veri arasına bir önbellek koyularak çözülmüştür. Bu önbelleğin gerçek çift portlu olması dolayısıyla, alınan verinin önbelleğe yazılması ile önbellekten SRAM'a yazılma işlemi aynı anda mümkün olmaktadır. Şekil 4.10'da SRAM, ekran ve gelen veri saat işaretlerinin zaman durum diyagramları yer almaktadır. Gelen veri, saat frekansının asenkronluğu dolayısıyla SRAM'a yazılamamaktadır.



Şekil 4.10 SRAM, ekran ve gelen veri saat işaretlerinin zaman durum diyagramları

Önbellek ara biriminde SRAM'a yazılacak veriyi ve adresini saklayan iki çift port bellek bulunmaktadır. Veri belleği 16 x 8 bit, adres belleği 16 x 18 bit olarak seçilmiştir. Gelen veri önbelleğin ilk adresinden itibaren yazılmaya başlanıp, adres sonuna gelindiğinde tekrar ilk adresten yazılmaya devam edilmektedir. Gelen veri bir adrese yazılırken aynı adresten veri okunmasını engellemek amacıyla iki adres sayıcısı oluşturulmuştur. Şekil 4.11'de önbellek ara birimi blok şeması yer almaktadır.

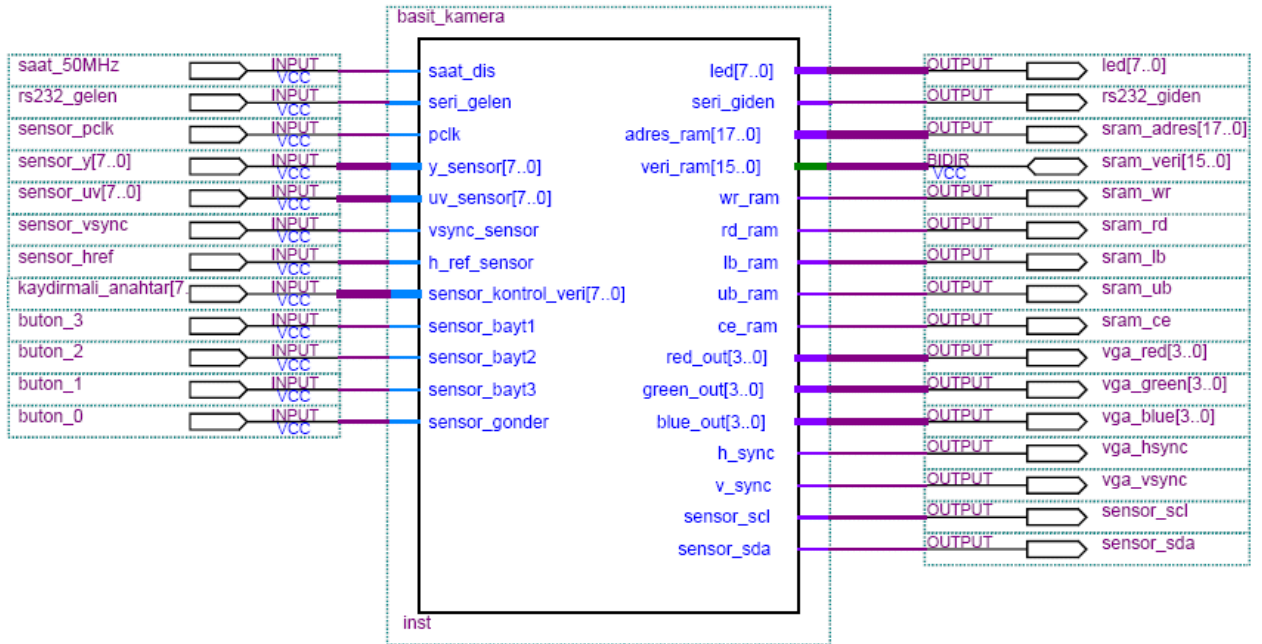


Şekil 4.11 Önbellek ara birimi blok şeması

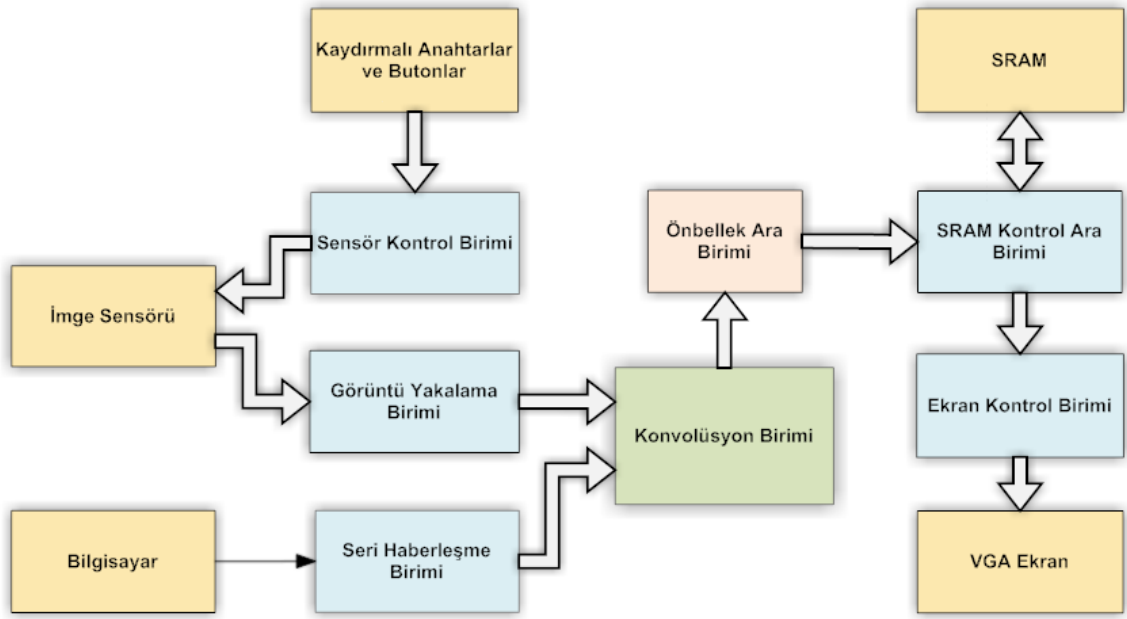
SRAM'a veri yazılmasını kontrol eden sayıcı, gelen verinin yazıldığı önbellek adresini sayan sayıcıdan geride olduğunda SRAM'a veri yazılmasına izin verir. Diğer durumlarda beklemeye geçer.

Önbellek biriminde veri yazılacağı adresle beraber saklanmaktadır. Veri SRAM'da hangi adrese yazılacağı bilgisiyle beraber bulunduğundan, önbelleğe gönderilen verinin zamanı geldiğinde doğru adrese yazılacağından emin olunur. Tasarlanan ana sistemde kameradan alınan veri üzerinde konvolüsyon işlemi yapılmakta ve SRAM önbellek arabirimine gönderilmektedir. Buradan veri SRAM'a yazılarak ekran kontrol birimi tarafından ekranda görüntülenmektedir. Konvolüsyon birimi görüntü sensörünün saat frekansı olan 8.865 MHz hızında çalışmaktadır. Dolayısıyla ekranda saniyede 50 kare görüntülense de efektif görüntüleme hızı yaklaşık 30 kare/s olmaktadır.

$$hız_{sistem} = \frac{1}{satir_{saat} * sütun_{saat} * t_{clk}} = \frac{1}{625 * 472 * \frac{1}{17.73 * 10^6}} \cong 30 \text{ kare/s}$$



Şekil 4.12 Ana sistemin giriş ve çıkışları



Şekil 4.13 Ana sistemin blok diyagramı



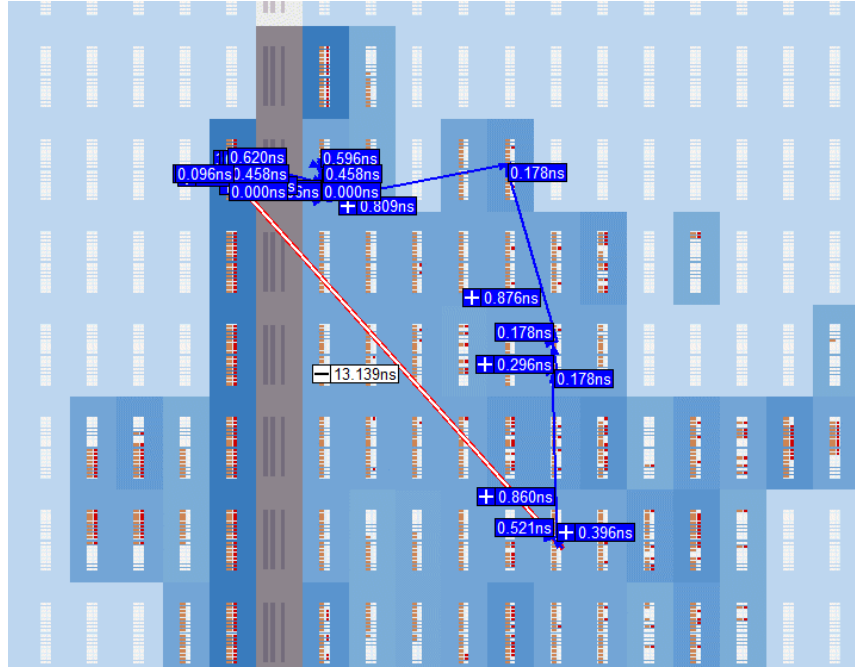
Şekil 4.14 Ekran üzerinde konvolüsyon sonucu

Flow Status	Successful - Wed May 20 16:34:45 2009
Quartus II Version	9.0 Build 132 02/25/2009 SJ Web Edition
Revision Name	adim_5
Top-level Entity Name	basit_kamera
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Met timing requirements	No
Total logic elements	1,441 / 18,752 (8 %)
Total combinational functions	1,373 / 18,752 (7 %)
Dedicated logic registers	567 / 18,752 (3 %)
Total registers	567
Total pins	105 / 315 (33 %)
Total virtual pins	0
Total memory bits	11,168 / 239,616 (5 %)
Embedded Multiplier 9-bit element	18 / 52 (35 %)
Total PLLs	0 / 4 (0 %)

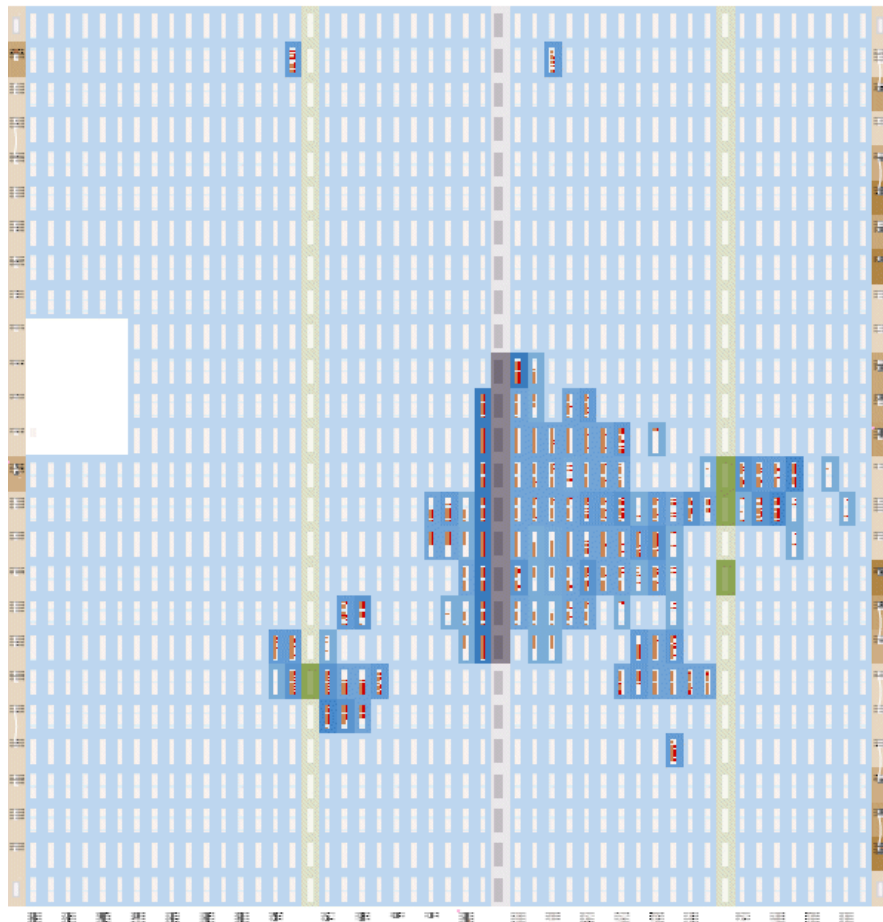
Tablo 4.3 Ana sistem için FPGA kaynaklarının kullanımı

Ana sistem sentezlendikten sonra FPGA üzerinde yerleşimi kullanılmakta olan Quartus II yazılımı tarafından otomatik olarak yapılmaktadır. Yapılan bu yerleştirme işleminde amaç sistemde kablo gecikmelerini en aza indirmektir. Sistemin maksimum hızını belirleyen etmenler kapı gecikmeleri ve özellikle kablo gecikmeleridir.

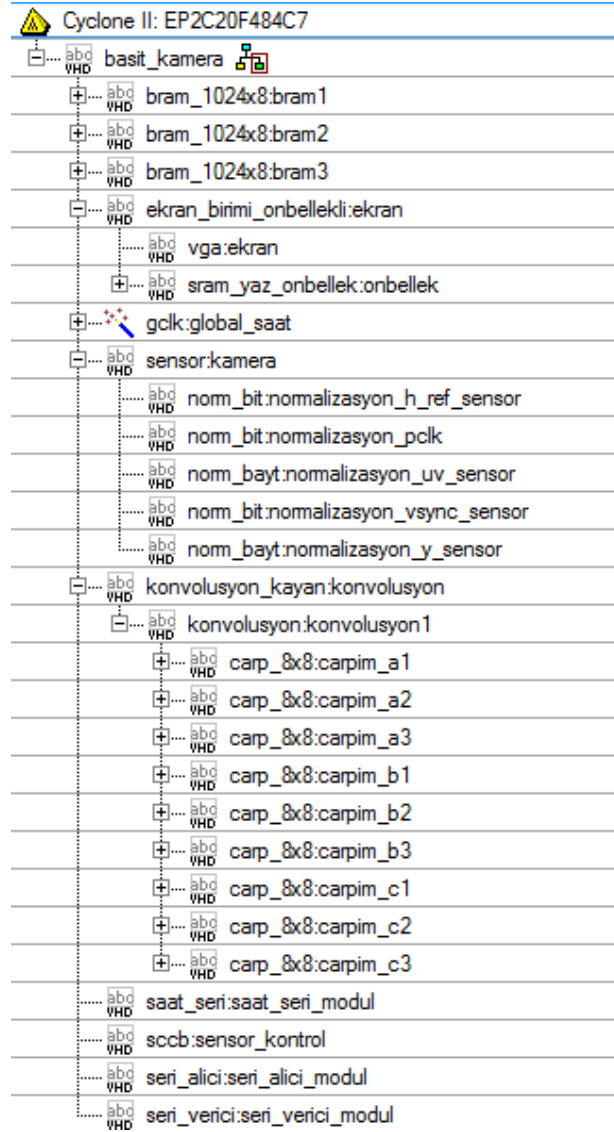
Birbiriyle ilintili bir lojik işlem biriminden diğer lojik işlem birimine veri aktarılırken bir miktar süre geçmektedir. Bu gecikme iki birim arasındaki uzaklığa bağlı olarak artmaktadır. Eğer bu gecikme bir saat periyodundan daha fazla ise diğer lojik işlem birimine daha veri gitmeden ikinci bir saat darbesi uygulanır ve lojik birim hattaki eski veri ile işlem yapar. Bu nedenle FPGA üzerinde tasarlanan birimlerin yerleşimi yapılırken kablo gecikmelerine çok dikkat edilmelidir. Şekil 4.15’de tasarlanan ana sistemin hızını sınırlayan kablo ve kapı gecikmelerinden biri görülmektedir. Yaklaşık 13 ns’lik gecikme dolayısıyla sistem maksimum 75 MHz’de çalışabilecektir. Bu hız tasarım algoritmasının ya da yerleşimin düzgünleştirilmesiyle artırılabilir fakat sistemin çalışma frekansı zaten 50 MHz olarak ayarlandığından buna gerek görülmemiştir.



Şekil 4.15 Sistem hızını sınırlayan kablo ve kapı gecikmelerinden biri



Şekil 4.16 FPGA üzerinde sistemin yerleşimi



Şekil 4.17 Ana sistemin hiyerarşisi

5 SONUÇ

Bu çalışmada gerçekleştirilen sistem ile görüntü sensöründen alınan görüntü üzerinde gerçek zamanlı konvolüsyon işlemi yapılmış ve sonuç ekran üzerinde görüntülenmiştir. Yapılan konvolüsyon işlemiyle görüntü üzerinde yumuşatma, keskinleştirme, kenar belirleme gibi filtreleme işlemleri gerçekleştirilebilmiştir.

Gerçeklenen sistem kendi başına çalışacak biçimde tasarlanmış olmakla beraber sistemin konvolüsyon şablonu sistem çalışırken seri port ile dışarıdan değiştirilebilmektedir. Böylece görüntü üzerine uygulanacak filtrenin değiştirilmesi için sistemin tekrar sentezlenmesine gerek kalmamıştır.

Gerçeklenen sistemin FPGA'ye ait kaynakların yaklaşık %10'unu kullanması aynı donanım üzerinde daha ileri seviyede tasarımların gerçekleştirilebilmesini mümkün kılmaktadır. Örneğin tasarlanacak yeni algoritmalarla kenarları belirlenmiş görüntü üzerinde yüz ve göz bölgeleri belirlenip bu bölgelerin takibi yapılabilir. Yine kenarları belirlenmiş görüntü üzerinde plaka bölgesini tespit edecek ve plakayı tanıyacak bir algoritma oluşturularak bir gömülü plaka tanıma sistemi gerçekleştirilebilir.

KAYNAKLAR

- [1] Altera Corporation, (2009), Quartus II Subscription Edition Software, <http://www.altera.com/products/software/quartus-ii/subscription-edition/qts-se-index.html>.
- [2] Mentor Graphics, (2009), ModelSim, <http://www.mentor.com/products/fv/modelsim/>.
- [3] Aydın, Atilla, (2005), "FPGA Yonga Mimarisi ve Kullanımı", Elektronik ve Haberleşme Müh. Süleyman Demirel Üniversitesi.
- [4] Altera Corporation, (2009), Altera DE1 Development and Education Board User Manual, http://www.terasic.com.tw/attachment/archive/83/DE1_UserManual_v1018.pdf.
- [5] Altera Corporation, (2004), Cyclone II Device Family Data Sheet, http://www.altera.com/literature/hb/cyc2/cyc2_cii5v1_01.pdf.
- [6] Electronics123 Inc., (2009) C3088 Specification Sheet, <http://www.electronics123.net/amazon/datasheet/c3088.pdf>.
- [7] Strangio, Christopher E., (2006) "The RS232 Standard", Cami Research, http://www.camiresearch.com/Data_Com_Basics/RS232_standard.html.
- [8] Wikimedia Foundation, Inc, (2009), Video Graphics Array, http://en.wikipedia.org/wiki/Video_Graphics_Array.
- [9] Ickes, Nathan, (2004) "VGA Video", Introduction to Digital Systems, <http://web.mit.edu/6.111/www/s2004/NEWKIT/vga.shtml>.
- [10] School of Computer Science, (2009), OV6620 Datasheet, Carnegie Mellon University, <http://www.cs.cmu.edu/~cmucam/Downloads/ov6620DSL.F.PDF>.
- [11] OmniVision Technologies Inc., (2002), Serial Camera Control Bus Functional Specification, http://www.ovt.com/products/sccbspec_an_2_1.pdf.
- [12] C. T. Johnston, K. T. Gribbon, D. G. Bailey, (2004),"Implementing Image Processing Algorithms on FPGAs", New Zealand, Proceedings of the Eleventh Electronics.
- [13] Ridha Djemal, Didier Demigny,Rached Tourki, (2009) "A Real-Time Image Processing with a Compact FPGA-Based Architecture", <http://www.design->

reuse.com/articles/10943/a-real-time-image-processing-with-a-compact-fpga-based-architecture.html.

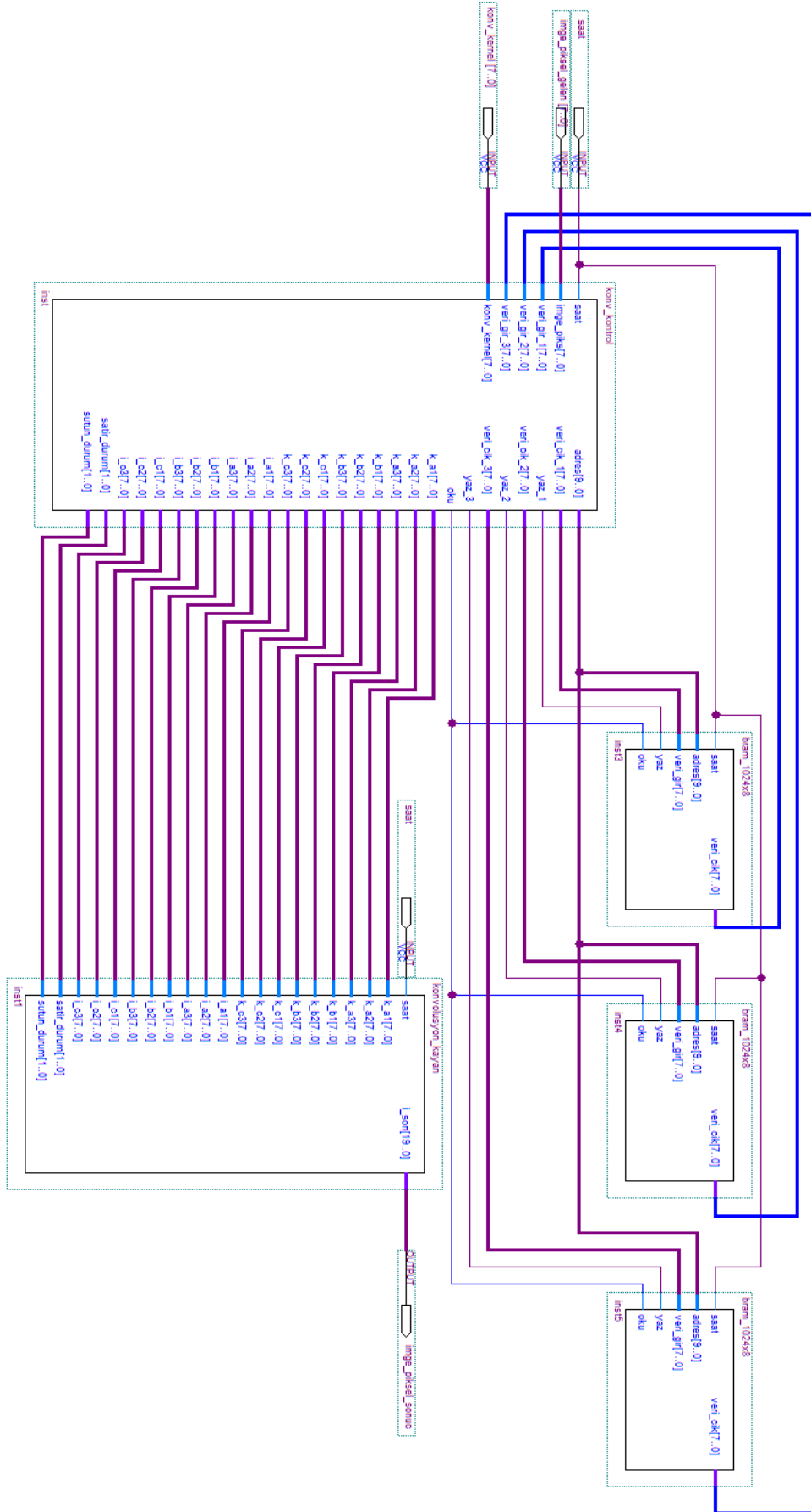
[14] L. Kessal, N. Abel, D. Demigny, (2003), "Real-time image processing with dynamically reconfigurable architecture", Real-Time Imaging.

[15] Nelson, Anthony Edward, (2000), " Implementation of Image Processing Algorithms On Fpga Hardware", Graduate School of Vanderbilt University.

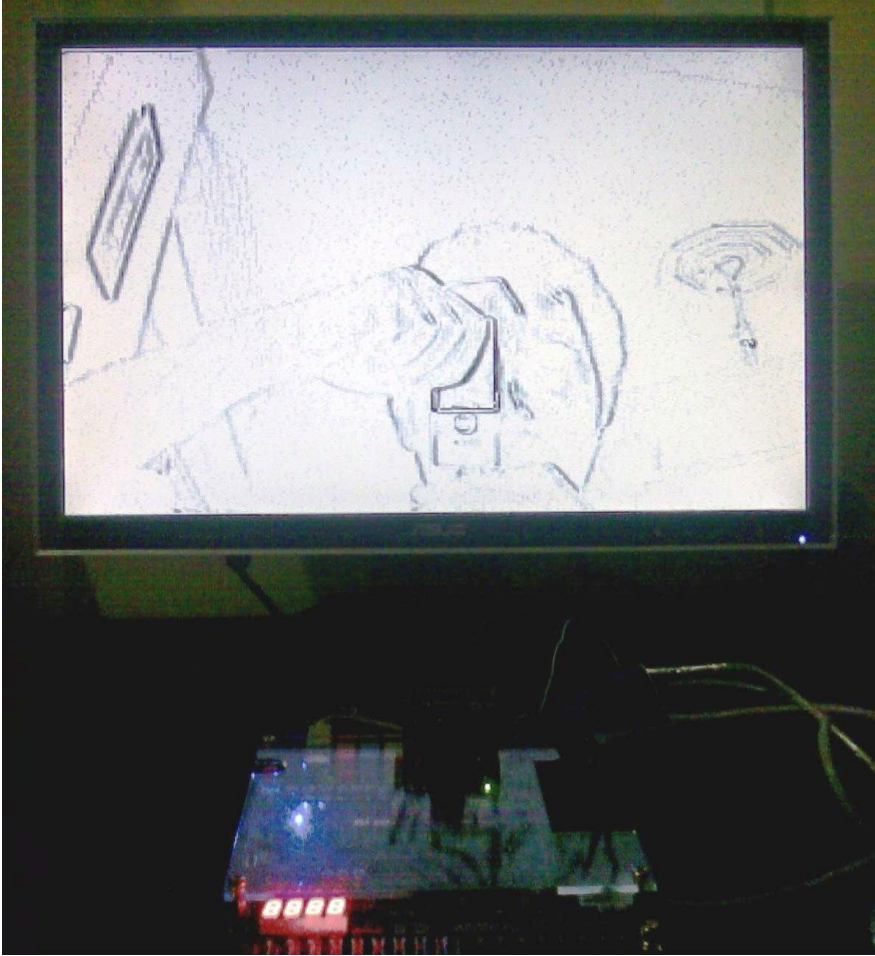
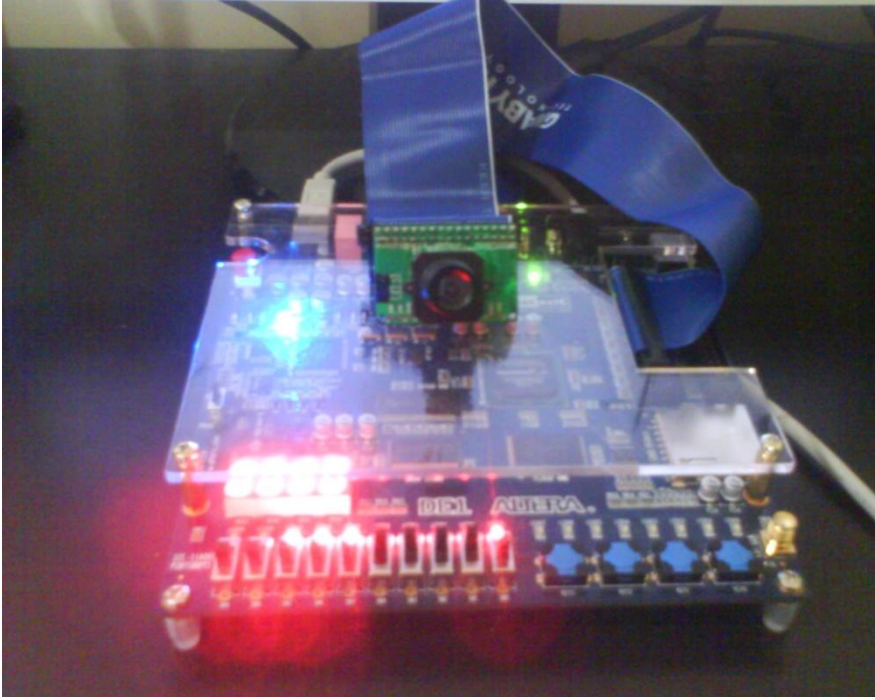
[16] Yeniçeri, Ramazan, (2007), "Cmos Görüntü Sensörü ve Fpga İle Sayısal Fotoğraf Makinesi Gerçeklenmesi", Bitirme Projesi, İstanbul Teknik Üniversitesi Elektronik Mühendisliği Bölümü.

[17] Image Processing Toolbox, (2009),The MathWorks, Inc., <http://www.mathworks.com/access/helpdesk/help/toolbox/images/>.

Ek 1 Konvolüsyon Birimi Blok Şeması



Ek 2 Uygulama Geliştirme Kartı Üzerinde Sistemin Gerçeklenmesi



Ek 3 Sistemle Haberleşmeyi Sağlayan MATLAB Kodu

seri.m

```

x=imread('cameraman.jpg');
    % verisi alınacak imgenin yolu
s2 = serial('COM2', 'BaudRate',38400,'timeout',1,'Terminator','');
    % burada uygun olan COM portu yazılmalı
    % baud rate ayarlanmalı default 9600
    % timeout alınan ya da gönderilen veri için bekleme süresi
x=rgb2gray(x);
    % imge gri formata çevrildi
xters=x';
    % trazozezi alındı
xvektor=xters(:);
    % bir sütunluk matris olarak biriktirildi
xvektor=xvektor/2;
    % değer aralığı 0 ile 127 arasında olacak şekilde ayarlandı
xvektor=int8(xvektor');
    % tranzpozezi alınarak int8 formatında bir satırlık matris olarak
    ayarlandı
komut=uint8([ 1 8 8 8 8 8 8 8 8 8 0 239 1 63 ]);
    % '1' başlama komutu
    % 9 bayt konvolüsyon matrisi sırasıyla k11, k12, k13, k21, k22, k23,
    k31, k32, k33
    % imge çözünürlüğü 2 bayt satır 2 bayt sütun genişliği
    % burada 240x320
gelen=[];
    % alınacak verinin biriktirileceği dizi oluşturuldu
[size_x,size_y]=size(xvektor);
    % imgenin toplam boyutu
fopen(s2);
    % seri port açıldı
fwrite(s2,komut(:));
    % komut gönderildi
for i=1:320:size_y;
    fwrite(s2,xvektor(i:i+319));
        % veri 320 baytlık parçalar halinde gönderiliyor
    z=fread(s2,320);
        % ve konvolüsyon sonucu geri alınıyor
    z=z';
    gelen=[gelen,z];
        % gelen sonuc gelen matrisine yazılıyor
end
fclose(s2);
gelenresim=parcala(gelen,219,320);
    % gelen veri 240*320 lik matris yapılıyor
gelenresim=int8(gelenresim);
figure,imshow(gelenresim);

function [imge]= parcala(gelen,sat,sut)
    imge=[];
    for i=0:1:(sat-1)
        imge=[imge;gelen((i*sut+1):((i+1)*sut))];
    end
end
end

```

ÖZGEÇMİŞ

Ad-Soyad	Ahmet Remzi ÖZCAN
Doğum Tarihi	25.11.1986
Doğum Yeri	Trabzon
Orta Öğrenim	2000-2004 Bursa A.O.S. Fen Lisesi
Staj Yapılan Yerler	2007 Tübitak UEKAE – MTRD (Multimedya Teknolojileri Değerlendirme, Araştırma ve Geliştirme Laboratuvarı) 2008 Delphi Automotive. Sys. Ltd. Şti. 2008 Thy Teknik A.Ş. Era Rev. Atl. Radyo Atelyesi
İletişim Bilgileri	
GSM	0 (544) 7403272
e-mail	ozcanar@hotmail.com