



**FPGA KULLANARAK GERÇEK ZAMANLI GÖRSEL
NESNE ALGILAMA**

Mehmet Rıza SARAÇ

**Yüksek Lisans Tezi
Mekatronik Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Zühtü Hakan AKPOLAT
Ağustos-2016**

**T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

FPGA KULLANARAK GERÇEK ZAMANLI GÖRSEL NESNE ALGILAMA

YÜKSEK LİSANS TEZİ

Mehmet Rıza SARAÇ

121134108

Tezin Enstitüye Verildiği Tarih : 20 Temmuz 2016

Tezin Sunulduğu Tarih : 03 Ağustos 2016

Danışmanı : Prof. Dr. Zühtü Hakan AKPOLAT (F.Ü.)

Diğer Jüri Üyeleri : Doç. Dr. Celaleddin YEROĞLU (İ.Ü.)

Doç. Dr. Ömür AYDOĞMUŞ (F.Ü.)

İÇİNDEKİLER

Sayfa No

İÇİNDEKİLER	II
ÖZET	III
SUMMARY	IV
ŞEKİLLER LİSTESİ	VI
TABLolar LİSTESİ.....	VII
KISALTMALAR LİSTESİ	VIII
1. GİRİŞ	1
1.1 Görme.....	1
1.2 Bilgisayar Görmesi.....	4
1.3 Daha Önce Yapılmış Çeşitli Çalışmalar	5
2. GÖRÜNTÜNÜN ELDE EDİLMESİ.....	7
2.1 Kameralar	7
2.1.1 CCD Algılayıcılar	7
2.1.2 CMOS Algılayıcılar	8
2.1.3 Renkli Kameralar	9
3. FPGA’NIN NESNE TANIMADA KULLANIMI	12
3.1 FPGA.....	12
3.2 FPGA ile Yapılmış Çeşitli Uygulamalar	16
4. FPGA ile NESNE ALGILAMANIN GERÇEKLEŞTİRİLMESİ	21
4.1 Kullanılan Donanım Birimleri.....	21
4.1.1 Terasic DE2-115 FPGA geliştirme ve eğitim kartı.....	21
4.1.2 Terasic D5M CMOS kamera kiti.....	25
4.2 Kullanılan Referans Görüntüleme Sistemi.....	29
4.2.1 Referans sistemin genel yapısı	29
4.2.2 Referans sistemin genel çalışma prensibi	30
4.3 Gerçekleştirilen Sistem ve Önemli Bazı Yapı Blokları	34
4.3.1 Satır tamponlama.....	34
4.3.2 FPGA sisteminde gecikmeler ve bunların kontrolü.....	37
4.3.3 Medyan filtre	40
4.4 Sistemin Modüllerinin Birleştirilmesi ve Çalıştırılması	42
5. SONUÇLAR ve ÖNERİLER	45
5.1 Sonuçlar	45
5.2 Öneriler	47
KAYNAKLAR.....	48
EKLER	50
A. Mevcut Sistem	50
A.1 Mevcut sistemin FPGA yüzey yerleşimi.....	50
A.2 FPGA dahili M9K RAM bloklarından bir örneği	51
A.3 Mevcut görüntüleme sistemi	52
A.4 Mevcut sistemin RTL blok şeması.....	53

B.	Gerçekleştirilen Sistem	54
B.1	Gerçekleştirilen sistemin FPGA yüzey yerleşimi	54
B.2	Gerçekleştirilen sistemin RTL blok şeması	55
B.3	Eklenen modülün RTL blok şeması	56
B.4	fcross modülün RTL blok şeması	57
B.5	Displays U3 RTL blok şeması	58
B.6	hclock modülü RTL şeması	59
B.7	Mask U4 RTL blok şeması	60
B.8	Tamponda kullanılan RAM hafıza RTL blok şeması	61
B.9	Mask process modülü RTL blok şeması	62
B.10	Colour modülü RTL blok şeması	63
B.11	Median modülü RTL blok şeması	64
B.12	Çalışma esnasında sistemin görüntüleri	65
C.	Median Modülünün Verilog Kodları	68
	ÖZGEÇMİŞ	72

ÖZET

FPGA kullanımı, görüntü işleme operasyonu sırasında veri boyutunun büyük olmasından dolayı birçok uygulamada tercih edilmektedir. Bu tezde, Terasic DE2-115 FPGA seti ve onunla uyumlu TRDB-D5M kamera modülü incelenmiş ve firma tarafından sağlanan Verilog dilindeki kamera uygulama modülü, geliştirmeye uygun referans bir sistem olarak kullanılmıştır. Sisteme farklı fonksiyonlar ve özellikler eklenerek bir nesne algılama sistemi gerçekleştirilmiştir.

Nesne algılama sistemi temel olarak 3x3 pencere boyutuna sahip bir medyan filtresinin donanımsal olarak optimize edilmiş bir versiyonunu içermektedir.

İkili imge oluşturmak için filtrelenmiş görüntüdeki kırmızı, yeşil ve mavi (RGB) renkler ve gri renk için farklı eşik değerleri kullanılarak segmentasyon yapılır. Sahnede bulunan nesneyi kabaca gösteren bu ikili imge sayesinde nesneye ait olduğu düşünülen pikseller için video tamponuna sıfırdan farklı bir veri değeri yazılmaktadır.

Elde edilen ikili imgeden nesnenin ağırlık merkezi ve bu merkezinin koordinatları belirlenmiştir. Belirlenen merkezin koordinatları onaltılık sayı sistemi (hexadecimal) olarak FPGA setinin üzerinde bulunan 7-segment göstergeye aktarılmıştır.

Tamamen gerçek zamanlı çalışan sistem üzerinde farklı aydınlanma koşulları ve eşik değerlerle deneysel çalışmalar yapılmıştır. Deneyler sırasında gözlemlenen, yayılma gecikmelerine bağlı problemler, kodlarda yapılan düzeltmelerle giderilmiştir.

Anahtar Kelimeler: FPGA, Gerçek Zamanlı Nesne Algılama, Medyan Filtre

SUMMARY

REAL-TIME VISUAL OBJECT DETECTION USING FPGA

The use of FPGA is preferred in many applications because of the massive amount of data processed by image processing operation. In this thesis, Terasic DE2-115 FPGA Kit and a compatible camera module TRDB-D5M has been analyzed. The application module written in Verilog provided from Terasic is used as the base system for the development of the image processing system. An object detection system has been created by adding various functions and features to the given system.

The object detection system comprises a hardware optimized version of median filter with a 3x3 window size.

The segmentation is performed to generate binary image using different threshold values for RGB and grey color levels in the filtered image. A data value different from zero is written to the video buffer for pixels considered to be object thanks to this binary image which shows roughly the object in the scene.

Coordinates of the centroid have been determined by using the obtained binary image of the object. The coordinates of the centroid have also been transferred as hexadecimal number to 7-segment display which is available on the FPGA kit.

The experimental implementation has been realized under different illumination conditions and thresholds values as a real-time system. The observed problems depending on the propagation delays have been compensated by improving the codes.

Key Words: FPGA, Real-Time Object Detection, Median Filter

ŞEKİLLER LİSTESİ

Sayfa No

Şekil 2.1	Vidicon kamera ve şeması	7
Şekil 2.2	Bazı CCD görüntü algılayıcılar	8
Şekil 2.3	Bazı CMOS görüntü algılayıcılar	9
Şekil 2.4	3 sensör ile renkli görüntü oluşturma	10
Şekil 2.5	Bayer filtresi ile renkli görüntü oluşturma	10
Şekil 2.6	Görüntüde eşik seviye uygulaması	11
Şekil 3.1	Programlanabilen bazı mantık devreleri	12
Şekil 3.2	Altera firmasının kullandığı çeşitli mantık birimleri (LE)	14
Şekil 3.3	Altera firmasının kullandığı LE şeması	15
Şekil 3.4	Normal konfigürasyona sahip bir LE	16
Şekil 3.5	Matematiksel işlem için konfigüre edilmiş bir LE	16
Şekil 4.1	Terasic DE2-115 eğitim kartı ön görünüşü	21
Şekil 4.2	Terasic DE2-115 eğitim kartı blok diyagramı	22
Şekil 4.3	Terasic DE2-115 eğitim kartı SDRAM bağlantısı	23
Şekil 4.4	Terasic DE2-115 eğitim kartı VGA bağlantısı	24
Şekil 4.5	Terasic DE2-115 eğitim kartı GPIO bağlantısı	24
Şekil 4.6	Terasic TRDB-D5M 5 megapiksel kamera kartı	25
Şekil 4.7	Terasic TRDB-D5M 5 megapiksel kamera sinyal hattı	26
Şekil 4.8	Terasic TRDB-D5M 5 megapiksel kamera kiti PLL sistemi	27
Şekil 4.9	DE2-155 ve TRDB-D5M karşılıklı GPIO bağlantıları	27
Şekil 4.10	Terasic TRDB-D5M 5 megapiksel kamera sensör detayı	28
Şekil 4.11	Terasic TRDB-D5M 5 megapiksel kamera sensörü mozaik filtresi	28
Şekil 4.12	Terasic TRDB-D5M 5 megapiksel kamera ham görüntü sinyali	29
Şekil 4.13	Referans Görüntüleme sistemi blok diyagramı	29
Şekil 4.14	Yatay ve dikey bindirme ile pixel okuma	33
Şekil 4.15	Görüntü işleme pencereciği ve satır tamponlaması	35
Şekil 4.16	Bir pencere işlemi için uygulanan paralel satır tamponlama	35
Şekil 4.17	Altera alt_shift_taps bloğu	36
Şekil 4.18	Altera alt_shift_taps bloğu RTL şeması	37
Şekil 4.19	Matematiksel işlem için konfigüre edilmiş bir LE	38
Şekil 4.20	Mantıksal işlem için konfigüre edilmiş bir LE	38
Şekil 4.21	Düşük gecikmeli bir sistemde işlem örneği	39
Şekil 4.22	Ortalama değerinde gecikmeye sahip bir sistemde işlem örneği	39
Şekil 4.23	Yüksek gecikme değerine sahip bir sistemde işlem örneği	40
Şekil 4.24	9 değer içinden medyan değerini bulmak için gereken işlem yığını	41
Şekil 4.25	Optimize edilmiş bir medyan bulma sistemi	41
Şekil 4.26	Donanım için optimize edilmiş bir medyan bulma sistemi	42
Şekil 4.27	Mask ön tamponu u0 RTL blok şeması	44
Şekil A.1	Mevcut D5M sistemi FPGA yüzey yerleşimi	50

Şekil A.2	M9K hafıza bloğu konfigürasyon örneği	51
Şekil A.3	Referans olarak kullanılan sistem	52
Şekil A.4	Mevcut D5M sistemi RTL blok şeması	53
Şekil B.1	Gerçekleştirilen D5M sistemi FPGA yüzey yerleşimi	54
Şekil B.2	Değiştirilmiş sistemin RTL blok şeması	55
Şekil B.3	Eklenen modülün RTL blok şeması	56
Şekil B.4	fcross modülün RTL blok şeması	57
Şekil B.5	Displays U3 RTL blok şeması	58
Şekil B.6	hclock modülü RTL şeması	59
Şekil B.7	Mask U4 RTL blok şeması	60
Şekil B.8	Tamponda kullanılan RAM hafıza RTL blok şeması	61
Şekil B.9	Mask process modülü RTL blok şeması	62
Şekil B.10	Colour modülü RTL blok şeması	63
Şekil B.11	Median modülü RTL blok şeması	64
Şekil B.12	Nesne algılama (çoklu nesne)	65
Şekil B.13	Nesne algılama (tek nesne)	66
Şekil B.14	Nesne merkezinin göstergeden okunması (tek nesne için)	67

TABLÖLAR LİSTESİ

Sayfa No

4.1	Terasic TRDB-D5M 5 megapiksel kamera temel özellikleri	26
4.2	VGA yatay ve dikey senkronizasyon değerleri	31
4.3	Renk bilgisinin hafıza hatlarına dağılımı	31



KISALTMALAR LİSTESİ

ALM	: Adaptive Logic Module
ASICs	: Application-Specific Integrated Circuits (for specific customs)
ASSPs	: Application-Specific Standard Parts (for multiple customs)
CCD	: Charge-Coupled Device
CMOS	: Complementary Metal-Oxide-Semiconductor
CPLDs	: Complex Programmable Logic Devices
DSP	: Digital Signal Processor
CPU	: Central Processing Unit
FBGA	: FineLine Ball Grid Array
FIFO	: First In First Out
FOV	: Field of View
FPGA	: Field-Programmable Gate Array
FVAL	: Frame Valid
GPB	: Global Probability of Boundary
GPIO	: General Purpose I/O
HSMC	: High Speed Mezzanine Card
İHA	: İnsansız Hava Aracı
IP	: Intellectual Property - Fikri Mülkiyet
LCD	: Liquid Crystal Display
LC	: Logic Cells
LE	: Logic Elements
LED	: Light Emission Diode
LVAL	: Line Valid
PCIe	: Peripheral Component Interconnect Express
PLL	: Phase Locked Loop
RAM	: Random Access Memory
RANSAC	: Random Sample Consensus
RGB	: Red Green Blue
SIMD	: Single Instruction, Multiple Data
SMA	: SubMiniature version A
SOC	: System On Chip
SRAM	: Static Random Access Memory
SSE	: Streaming SIMD (Single Instruction, Multiple Data) Extension
USB	: Universal Serial Bus
VGA	: Video Graphics Array
VHDL	: VHSIC (Very High Speed Integrated Circuit) Hardware Description Language
VHSIC	: Very High Speed Integrated Circuit
YCbCr	: Luma - Chroma(blue-dif) - Chroma (red - dif)

1. GİRİŞ

Mühendisliğin amaçlarından biri yapılan işlerin daha verimli ve etkili yapılmasını sağlamaktır. Bir iş için en uygun çözüm bazen basit olanı kullanmak olsa da çoğu zaman sistemler gittikçe karmaşıklaşmaktadır. Bunun en büyük sebebi gittikçe daha fazla çeşitte imkan ve esnekliğin talep edilmesidir.

Çalışma hayatından köleliğin kaldırılmasının ardından yerlerine geçecek şeyler için imkanların araştırılması bizi gittikçe daha fazla biyolojik sistemlerde bulunan fonksiyonları suni olarak sağlamaya zorlamaktadır. Her ne kadar bazı yerlerde insan kaynaklarının daha ucuz olması bu isteği bastırabilmekte ise de refah arttıkça bu yerler azalmakta ve daha insan-cıl yöntemlere geçilmektedir.

Her ne şekilde olursa olsun kullanılan insan kaynağının bizi cezbeden özelliklerinin, mali veya insani sebeplerle gittikçe daha az insan kaynağı kullanacağımız gelecekte yerinin doldurulması gerekmektedir. Bunun için gittikçe gelişen robot teknolojisinin yanında biraz geç de olsa yarışa katılan yapay görme becerisi, teknolojinin desteğiyle pazar payını arttırmaktadır.

Kabul etmek gerekir ki çoğu mühendislik uygulaması bir başkasının hayatını kolaylaştırmayı kendi işlerini zorlaştırarak yapabilmektedir. Nöbetçi eczane yazısını tahtaya yazmak yerine bunu elektronik olarak sağlamak arasında kıyas yaptığımızda, kullanıcı açısından kolaylıklar olsa da mühendislik açısından zorlukları arttıran bir farklılık barındırmaktadır.

Görme sistemlerinde popüler olarak kullandığımız yazılımsal çözümler esneklik ve kullanım kolaylıkları sayesinde çoğu geliştiriciyi cezbetmekte ancak bazıları da bu yöntemlere saplanıp kalmaktadır. Görme sistemi çoklu bir sistem olarak optiği, donanım ve yazılımı barındırmaktadır. Bunlardan bir tanesine saplanıp kalmak bizi muhtemel çözümlerden alıko-yabilmektedir.

1.1. Görme

Çoğu lisanda anlamak veya farkına varmak gibi anlamlara da gelen görme, insanlardaki muhtemelen en önemli duyu sistemidir [1]. Bu görme yetisini mühendisler olarak makinelere taşımak gerçekten zorlu bir iştir. Dikkat edilirse geleneksel cihazlarımız çevreleri ile etkileşimde ön safları işgal eden algılayıcılar grubunda genelde manyetik, kapasitif; temaslı temassız bir çok sensör ile birlikte hali hazırda optik sensörler barındırır da görme bunlardan

çok farklı olmaktadır.

Temelde bizim görmeden kastımız; bir sistemin etkileşimde bulunacağı çevresiyle ilgili önemli pozisyon, şekil ve çeşitli fiziksel özellik bilgilerinin yüksek duyarlılıkta edinilmesidir. Diğer duyu sistemleri bu amaca hizmet etseler de genelde kısıtlı mesafe ve alanlarda çalışabildiklerinden iş yükü belli duyularda odaklanmaktadır. Işığın olmadığı veya yüksek oranda soğurulduğu bazı ortamlarda kullanılan sonar prensibine dayanan eko-lokasyon haricinde çoğunlukla bu iş elektromanyetik tayfın belli bölgelerine hassas olan gözler ile yapılmaktadır. Bu iş yükü sebebiyle gözlerde bulunan algılayıcı sayısı geri kalan tüm duyu sistemlerinden fazladır [2]. Çevrenin 2 ya da 3 boyutlu bir modelinin sistemde bilgiyi işleyen biriminde oluşturulmasına yardımcı olmak için elektromanyetik tayfın, belli gölge ya da gölgelerine duyarlı ve yeterli uzaysal çözünürlüğü sağlamak amacıyla gerekli algılayıcı ağına sahip yapılar olan gözler, ilgili verileri alarak bunları merkezi bilgi işleme biriminde gönderirler. Burada aktarılan bilgi işe yarar anlamlı sonuçlar üretmek için işlenmektedir. Bu sürecin bilgiyi alan, işleyen ve anlamlandıran kısımlarının tamamı görme sistemi olarak adlandırılmaktadır.

Görme olayının gerçekleşebilmesi için öncelikle bilgi edinilecek nesnelerin bulunduğu ortamın aydınlatılması gerekmektedir. Bu aydınlatmada kullanılacak ışığın ortamın boşluk olarak adlandırılacak kısmından en azından kabul edilebilir bir oranda geçebilmesi gerekmektedir. Bunun için uygun tayf bölge ya da bölgelerinden seçilmeli veya bu bölgeleri de ihtiva etmelidir. Örneğin elektromanyetik tayfın görünür bölgesinde çalışılırsa yoğun duman ya da sisli bir ortamda görme gerçekleşmezken kızılötesi bölgede gerçekleşebilmektedir. Endüstride bir mamulün normal gözle görünemeyen iç kısımlarının muayenesinde kızılötesi ya da X ışını bölgesinde işlem yapmak gerekmektedir. Başka bir örnekte aksine saydam görünen bir nesnenin muayenesi için bu kez morötesi bölgesine geçilmesi gereksinimi ışık kaynağının ve çalışılacak tayf bölgesi seçiminin önemini göstermektedir.

Canlılarda ki görme sisteminin aksine makine görmesinde ışık kaynağının gücü çok önemlidir. Öyle ki çoğu zaman aynı değerleri garanti etmek için aydınlatma suni olarak elde edilmektedir. Fabrika ortamında bu kolay olurken robot görmesi gibi konularda mevcut dış durumlara adaptasyon istenebilmektedir.

Işık kaynağından çıkan radyasyonun nesneye varıncaya kadar katettiği yol ışık kaynağının soğurulması ve dağılması açısından önemlidir. Yukarıdaki örneklerden birinde de bahsedildiği gibi sisli bir ortamda görünür bölgede belli bir mesafeden sonrası için opak iken,

kızılötesi bölge için hala saydamlığını korumakta böylece görme işlevi sürdürüle bilmektedir. Su altı sistemlerinde ise bu durum daha belirgin olmaktadır. Her tayf bölgesi su altında aynı derinliğe inemez ve su ışık kaynağının radyasyonunu mesafe ilerledikçe soğurup engellemektedir. Bu sebeple havada olduğundan çok daha kısa bir mesafede ışık soğurulmaktadır. Bir başka problem ise su ortamında nesnenin normal hava ortamında görülen renkleri farklı görülebilmesidir.

Ortamın kırma indisi ayrıca önemlidir. Hava ortamında ışığın sensör ağının üzerinde gerçek bir görüntü oluşturması için odaklamada kullanılan optik sistemde ki merceklerin odak uzaklıkları havaya göre hesaplanmıştır. Su altında bu değer farklıdır, dolayısıyla sensörün optik kısmının en azından giriş kısmı için ortam farklılıklar oluşturacaktır. Elbette ışığın optik yolunu kontrol eden eleman mercek yerine ayna olduğunda işler değişir, aynaların odak uzaklıkları ortamdaki ortama değişmeyecektir.

Odaklamada ışık toplamada aynaların kullanıldığı sistemler omnivision görüntüleme sistemleri ve teleskoplardır. Aynalı teleskoplarda merceklerin ağırlık, kırma kusurları ve ışık soğurma özelliklerinden sistemi kurtarmak için ayna kullanılmaktadır.

Kaynaktan cisme ulaşan ışık nesneden yansıyarak, içinden geçerek ya da soğurulmadan sonra tekrar ışınarak yoluna devam eder. Bu noktada artık ışık cisimle ilgili bilgilerde barındırmaktadır. Göze veya kameraya ulaşınca kadar mevcut ortama maruz kalır, içinden geçilen nesnelerde nesnenin kendisi de içinden geçilen bir ortamdır. Nesne ışığı soğurup tekrar ışımaya yaparsa kendisi ışık kaynağı gibi davranmaktadır.

Bu adımları geçerek göze veya kameraya ulaşan ışık artık kamerada uygun bir şekilde optik bir sinyalden elektriksel sinyallere dönüştürülmektedir. Burada kameranın optik aksamı halen ışığın geçtiği bir ortamdır. Örnek olarak uzak morötesi ışık için normal cam opaktır dolayısıyla bu ışığa hassas bir alıcının optik sisteminin de buna uygun olması gereklidir.

Zahiri görüntüleri sensör algılayamamaktadır. Bu sebeple alıcıda bulunan sensör ağı üzerine kırıcı veya yansıtıcı düzeneden geçen ışık *gerçek görüntü* oluşturmaktadır. Sensör ağı üzerine gerçek görüntünün oluşabilmesi için odaklama ayarlarının yapılmış olması gerekmektedir. Sensör ağında ki sensör sayısı çözünürlüğü belirlemektedir. Makine görmesi sisteminde sensörler ağı her yerine homojen dağılmışken, ilham almaya çalıştığımız biyolojik sistemlerde dağılım heterojendir [2].

Işığın sensör üzerinde oluşturduğu değişimler yardımcı devreler ile güçlendirilip uygunlaştırılarak merkezi işlem ünitesine gönderilmektedir. Ancak hem biyolojik hemde elektronik sistemlerde aktarılacak bilgi miktarını düşürebilmek için bilgi kısmen de olsa işlenebilmektedir. Çoğu elektronik sistemde bilgi belli standartlara getirilerek sevk edilir.

Merkezi sisteme gelen bilgi gerekirse tekrar açılarak işlenmeye başlamaktadır. Çoğu zaman bu süreç farklı basamaklardan oluşur ve özelleşmiş donanımlar bulundurabilmektedir. Biyolojik sistemlerde neredeyse her aşamanın kendine has merkezi sinir dokusu bulunmakta ve işlemler neredeyse tamamen paralel olarak gerçekleştirilmektedir. Makine veya bilgisayar görmesi sistemlerinde özelleşmiş donanımların kullanıldığı alanlar olsa da henüz merkezi işlemci üzerinde bu işlemler seri olarak zaman paylaşımli bir şekilde yapılmaktadır. Bunun sebep olduğu gecikmeler bazen görme sisteminin sistemin kalanına entegre edilmesini engellemekte bazen de sistemi görme sistemine uyacak şekilde yavaşlatmak ya da görme sisteminin belli özelliklerini kısıtlayarak hızlı olmasını sağlamak şeklinde, problemi çözmek yerine kaçınmak şeklinde halledilmektedir.

Makine görmesi uygulamalarının çoğunda görme sistemi bir bilgisayarın kontrolünde de çalışsa, gereken hızı sağlamak için kendi donanımsal görüntü işleme birimlerini kullanmaktadırlar. Bu donanımlar üst seviye görüntü işleme aşamalarından daha ziyade alt ve temel seviye işlemleri gerçekleştirip sistemin yükünü azaltmaktadırlar. Örnek olarak bir balık kesme makinesinde kullanılan görme sisteminde CCD (Charge-Coupled Device) kameranın bağlandığı GPB (Global Probability of Boundary) temelli görüntü işleme kartı görüntüyü işlemektedir [3].

1.2. Bilgisayar Görmesi

Biyolojik olmayan görme sistemleri için “suni” kelimesi kullanıldığında gerçekte biyolojik bir sisteme, gözündeki problem sebebiyle sağlanan yapay görme anlaşılmaktadır. Makine görmesi dendiğinde ise yeterli kapsayıcılık olmamaktadır. Makine görmesi ile bilgisayar görmesi ilişki ya da farklarından bahsedilebilecekse de temelde biyolojik olmayan görme sistemlerini *Bilgisayar Görmesi* başlığı altında incelemek doğru olacaktır.

Temelde bu konu başlığı altında *Görüntü İşleme*, *Makine Görmesi* ve *Robot görmesi* konu başlıklarının da işlenmesinin temel sebebi; görüntü işleme sistemlerinin bir aşamada bilgisayarın esneklik ve gücüne ihtiyaç duyması sebebiyle sistemde hep bir bilgisayarın bulunmasıdır. Çok geniş bir yelpazeye dağılmış olan bilgisayar görmesi kendi içinde görme

dışında temel alt başlıkları da bulundurmaktadır. Matematik, yapay zeka, makine öğrenmesi ve istatistik gibi kendi başlarına birer konu başlığı halinde bulunan disiplinleri barındırması bilgisayar görmesi konusunu daha da genişletmektedir.

Temel konu başlıklarının bazıları ise belli noktalarda özelleşmektedirler. Örnek olarak makine görmesi daha çok gerçek zamanlı olarak yapılır, kontrollü ışık gerektirir ve daha çok endüstride makinelerin kontrolü için kullanılmaktadır. Robot görmesinde makine görmesine nazaran kontrolsüz ve farklı aydınlatma değerlerinde çalışması istenir, çünkü robot dış ortamda da bulunabilmektedir. Öte yandan robot görmesinde bazen kameralar robotun kendisiyle birlikte hareket ederek ego-motion oluşturabilmektedir.

Biyolojik sistemlerden önemli farkları burada sayacak olursak, canlılarda ki dikkat sisteminden neredeyse tamamen yoksun olarak sensördeki tüm reseptörler dikkate alınmaktadır. Sensör ağında algılayıcı birimler çoğunlukla homojen dağılmıştır. Bu gibi etkenler makinelerin işlem yükünü biyolojik sistemlerden farklı bir şekilde arttırmaktadır.

1.3. Daha Önce Yapılmış Çeşitli Çalışmalar

Konu ile ilgili geçmiş çalışmaların anlaşılabilmesi için lisans bilgilerimizin üzerine bu alana has donanım ve yazılımlara ait temel bilgileri eklememiz gerekmektedir. Kullanılacak donanımların her biri için üreticinin sağladığı dokümanlar da büyük ölçüde başvurulması gereken kaynaklardır.

Özellikle FPGA (Field-Programmable Gate Array) ile gömülü görüntü işleme sistemlerine iyi bir başlangıç olması için Donald G. Bailey tarafından yazılmış FPGA’larda gölümü görüntü işleme sistemlerinin tasarımı ile ilgili kitap, kullanılan FPGA donanımları ve bunlar üzerinde uygulanan yöntemler açısından temel ve zengin bir kitap olarak karşımıza çıkmaktadır [1].

Kaynak olarak literatür incelendiğinde FPGA sistemlerinin çok çeşitli şekillerde görsel işlemler için kullanıldığı görülmektedir.

Gösterilebilecek çok sayıda örnek arasından; FPGA ile hızlandırma kullanarak trafik levhalarını normal PC sistemlerinden daha etkili tespit eden sistem örneği verilebilmektedir [4]. Başka bir uygulama örneğinde FPGA hızlandırma sayesinde geleneksel sistemlere göre daha hızlı ve daha az enerji sarf eden sistemler önerilmiştir [5]. Ölçeklenebilir nesne algılama hızlandırıcılarının farklı FPGA yongaları için otomatik kod üreten örnek yazılımlar gerçekleştirilmiş ve başarımları incelenmiştir [6]. Trafik gözetleme sistemleri için FPGA

temelli hareketli nesne yakalama algoritması oluşturulan farklı bir örnekte alt seviye FPGA uygulamalarına örnek verilmiştir [7].

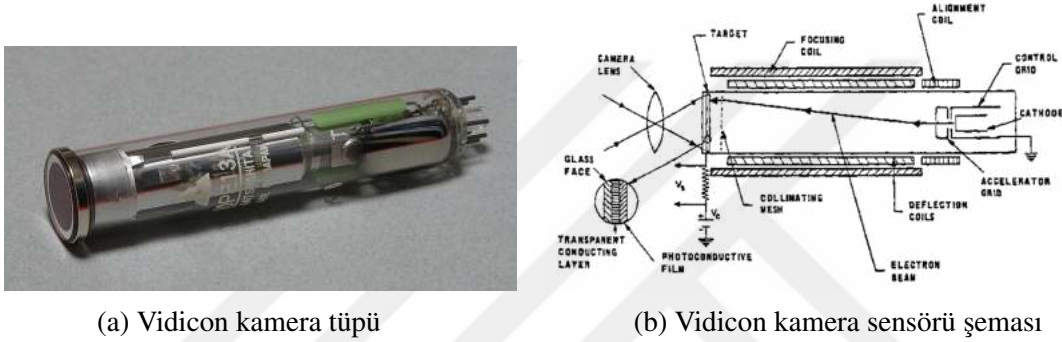
Daha gelişmiş sistem örneklerinden nesne bulma uygulamaları için FPGA ile yapılmış pixel işleme ünitesi için DSP (Digital Signal Processor) ve FPGA karşılaştırması yapılmıştır [8]. Bu tezde kullanılan FPGA geliştirme kartı ve kamerası ile gerçekleştirilen başka bir örnekte, İHA (İnsansız Hava Aracı) sistemleri için görüntü işleme sistemi tasarlanmıştır [9].

Çok kullanılan filtrelerden medyan filtresinin FPGA sistemlerinde gerçekleştirilmesine yönelik çeşitli çalışmalar yapılmıştır. Öncelikle medyan filtresinin optimizasyonunu sağlamak için yöntem geliştirilmiştir [10]. Pencere boyutunun sebep olduğu sistem kompleksliğini azaltabilmek için girişimlerde bulunulmuş ve 5x5 boyutundan sonra avantaj sağlanabilmiştir [11]. Bunun dışında görüntü işlemede çeşitli hafıza düzenlemeleri ile enerji verimliliği sağlamaya yönelik medyan uygulamaları yapılmış olup başarılı sonuçlar alınmıştır [12]. FPGA kullanılmasına rağmen bazı çalışmalarda soft-processor kullanılmış ve kaynak harcamasını kısıtlayabilmek adına medyan filtre RGB (Red Green Blue) sinyaline değil, YCbCr (Luma - Chroma (blue-dif) - Chroma (red-dif)) sinyallerinden Y bileşenine uygulanmıştır [13]. Kendi yaptığımız sistemde ise kameradan alınan görüntü işlenerek doğrudan monitörde gösterilecektir.

2. GÖRÜNTÜNÜN ELDE EDİLMESİ

2.1. Kameralar

Kameralar elektronik sistemde göz gibi görev yapmaktadırlar. Bu başlık altında daha çok kameralarda kullanılan temel algılayıcı çeşitleri ele alınacaktır. Tipik kamera algılayıcı çeşitlerinden ilki Şekil 2.1 de görülen analog olan vidicon kamera tüpleridir [14]. CCD kameralar çıkmadan önce yaygın olarak kullanılmaktaydı. Işığın düştüğü yüzeyde oluşturduğu direnç değişiminin taranarak ölçülmesi ile görüntü elektronik ortama aktarılmaktadır.



Şekil 2.1: Vidicon kamera ve şeması

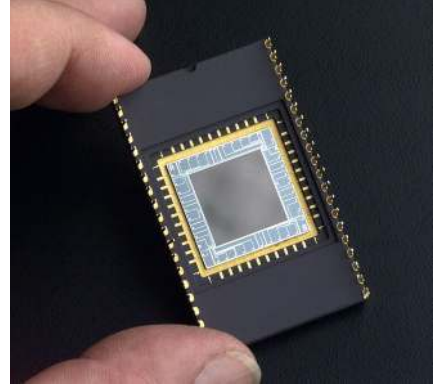
Günümüzde vidicon tüplü kameralar çok az kullanılmakta olup artık yerlerine yarı iletken temelli CMOS (Complementary Metal-Oxide-Semiconductor) ya da CCD kameralar kullanılmaktadır.

2.1.1. CCD Algılayıcılar

CCD (Charge Coupled Device) ile ilgili ilk çalışmalar 1969 da AT&T Bell laboratuvarlarında Willard Boyle ve George E. Smith tarafından yapılmıştır [15]. Bu çalışmada CCD daha ziyade yeni bir kaydırmalı kaydedici olarak düzenlenmişken 1960ların sonlarında Michael Tompsett tarafından görüntü elde etmek için de patent alınmıştır [16]. Bu sistemde ışığın etkisiyle yarı iletkende oluşan elektriksel yükler, yarı iletkene uygulanan sıralı gerilim darbeleri yardımıyla bir sonraki komşu hücreye aktarılacak en sonunda çıkışa iletilir ve çıkışta yük miktarı ölçülmektedir.



(a) CCD satır tarayan sensör



(b) CCD alan tarayan algılayıcı

Şekil 2.2: Bazı CCD görüntü algılayıcılar

İki boyutlu görüntünün elde edilebilmesi için nesne ya mekanik olarak satırlar halinde taranır, ya da tek pozda bir çerçeve taranmaktadır. Birinci yönteme örnek olarak fotokopi, tarayıcı ve fax makineleri ile endüstride özellikle gazetecilikte kullanılan ve baskıdan çıkan gazetelerin üretim bandından akarken tarayan sensörler Şekil 2.2a da görülen hat sensörleridir [17]. Diğer tür ise Şekil 2.2b deki örnekte görüntüyü bir seferde alıp dahili olarak satır satır tarama yapan sensörlerdir [17]. Bunlarda video kameralarda ve fotoğraf makinelerinde kullanılmaktadırlar. Hangisi olursa olsun iki boyutlu bilgi tek boyutlu hale getirilerek aktarılmaktadır. Alan tarayan sensörlerde çerçevenin okunmasında pozlama ve transfer sürecinde farklı yöntemler kullanılabilmektedir.

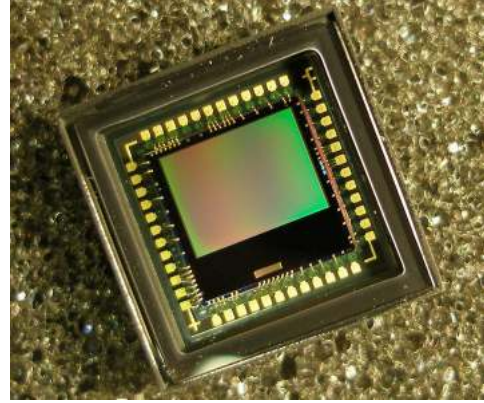
Öncelikle pozlama süresince yarı iletken yüzeyinin algılayıcı kısmı ışığa maruz bırakılmaktadır. Pozlama süresince oluşan elektriksel yükler daha sonra sıralı elektriksel darbeler ile tayin edilmiş yol boyunca ilerleyerek ölçüm birimine iletilmektedir. Bu ilerleme süresi boyunca yükün algılayıcı hücreleri ne kadar meşgul edeceği, başka bir alternatif yol olup olmadığı her yöntemle değişmektedir.

2.1.2. CMOS Algılayıcılar

Katıhal elektroniğin kazandırdığı bir başka algılayıcı türü ise CMOS algılayıcılarıdır. Bu algılayıcılarda bilgi CCD yongalarında olduğu gibi katarlar halinde çekilmemektedir.



(a) CMOS satır tarayan sensör



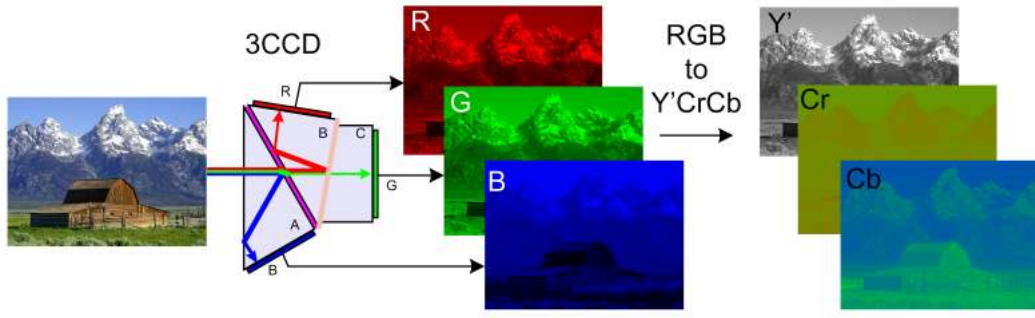
(b) CMOS alan tarayan algılayıcı

Şekil 2.3: Bazı CMOS görüntü algılayıcılar

CMOS sensörlerde her bir algılama hücreciği kendi başına bir devre içermektedir. Verinin alınması için gerekli hücreyi belirtmek üzere satır ve sütun bilgisi gönderilir ve ilgili hücreden gelen veri okunmaktadır. CCD algılayıcılara göre büyük oranda enerji tasarrufu sağlayan ve daha ucuza üretilebilen bu algılayıcı çeşidi gürültü seviyesi daha yüksek olmasına rağmen çoğu pille çalışan cihazda tercih edilmektedir. Hatta aslında piyasada ki çoğu cihazda bu algılayıcı çeşidi mevcuttur. CCD sensörler bazı deney setleri, yüksek kaliteli kameralar ve astronomi gibi uygulamalarda kullanım alanı bulabilmektedir. Şekil 2.3 te bazı CMOS sensör çeşitleri görülmektedir [18, 19].

2.1.3. Renkli Kameralar

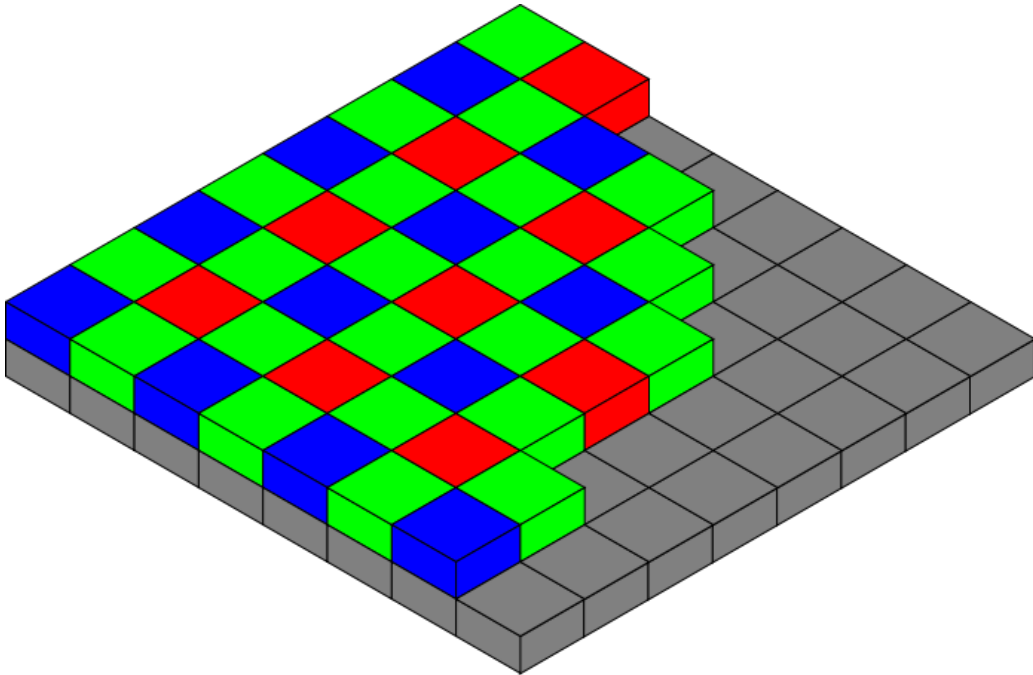
Yukarıda bahsi geçen sensörlerin algılayıcı hücrelerinin tayf duyarlılıkları bir yonga için tamamen aynı olmaktadır. Renkli görüntü elde edebilmek için kullanılan iki ayrı sistem bulunmaktadır. Bunlardan ilki üç ayrı sensör ile renkli görüntü yakalamaktadır. Bu yöntemde objektiften gelen ışık optik düzenele renklerine ayrılmakta ve Şekil 2.4 te görüldüğü gibi her renk için ayrı bir sensör yongasını etkilemektedir [20].



Şekil 2.4: 3 sensör ile renkli görüntü oluşturma

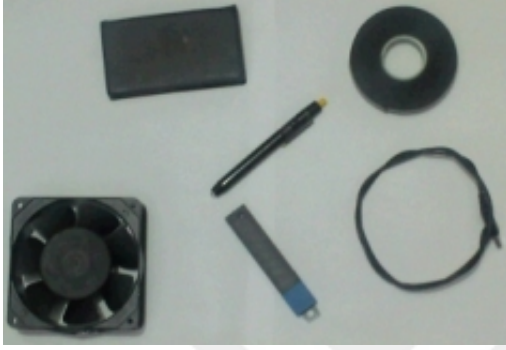
Bu sistemin avantajı her renk için ayrı yonga kullanıldığından çözünürlüğü yüksektir, ancak 3 resmin de birbiri üzerine oturması için hassas düzeyde mekanik ve elektronik ayarlamalar gerektirmektedir. Daha çok sabit kameralarda ve yüksek kaliteli profesyonel kameralarda kullanılmaktadır.

Diğer bir yöntem ise Şekil 2.5 te görüldüğü gibi tek bir sensör yongası kullanılmaktadır [21]. Yonga üstünde ki farklı hücrelere farklı renkler atanabilmesi için mozaik şeklinde bir renk filtresi kullanılmaktadır. İlk defa 1976 yılında Bryce E. Bayer tarafından patenti alınan bu yöntem bayer filtresi olarakta bilinmektedir [22]. Ancak neredeyse her firma kendisi için bu sistemin bir uyarlamasını yapıp patentini almıştır.

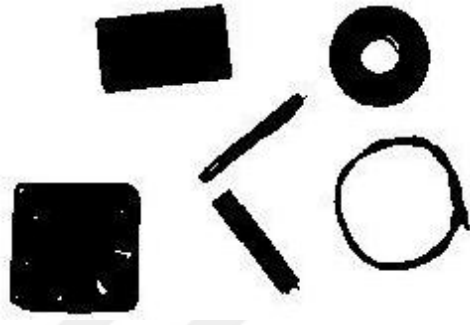


Şekil 2.5: Bayer filtresi ile renkli görüntü oluşturma

Alınan görüntüler binary(ikili) olarak işlenerek ya parlaklık ya da şekilsel özellikler ile belirli görüntü bölgeleri nesne olarak tanımlanmaktadır. Bu görüntü işlemleri için kullanılan yaygın yazılım çözümleri yerine bu tezde donanımsal çözümlerden FPGA incelenmiştir. Özellik çıkartma sisteminin düzensiz hafıza erişimi donanımsal hızlandırıcılar için problem oluşturduğundan parlaklık değerleri ile işlem yapılmıştır. Şekil 2.6 da eşik seviye uygulamasının bir örneği görülmektedir.



(a) Normal görüntü



(b) Eşikleme uygulaması sonrası

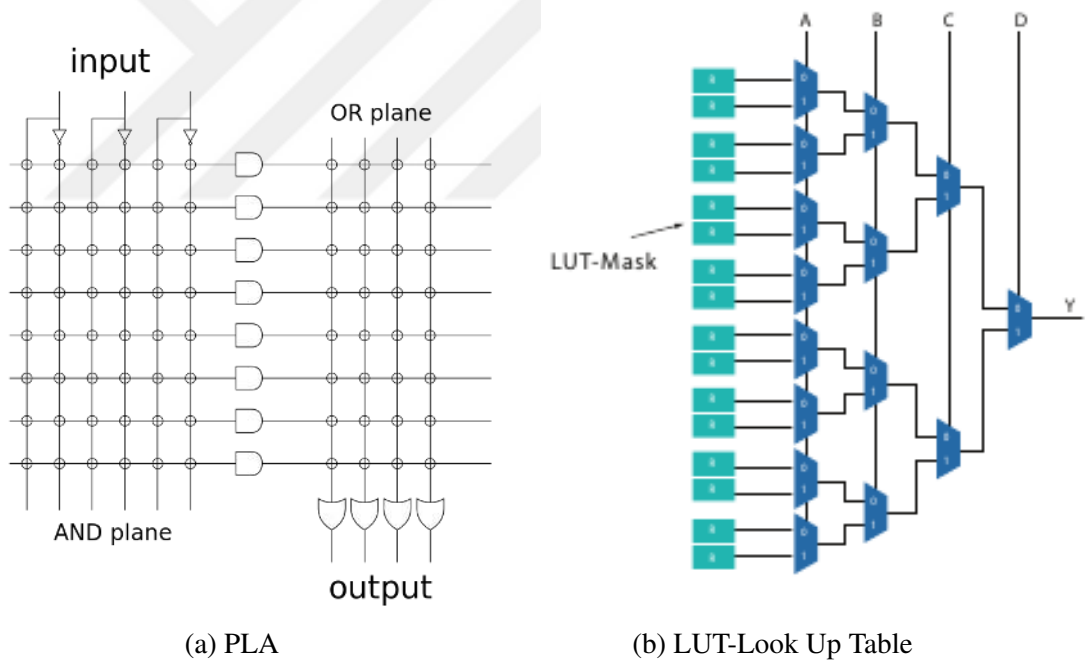
Şekil 2.6: Görüntüde eşik seviye uygulaması

3. FPGA’NIN NESNE TANIMADA KULLANIMI

3.1. FPGA

Yukarıda bahsi geçen işlem ünitelerinin işlemcileri sabit mimaride olup belli görevleri paralel olarak gerçekleştirilseler bile gerçekte sırayla yapmaktadırlar. İyileştirmelerin bazıları işlemci içindeki çekirdek sayılarını arttırmak ve bazıları da mevcut çekirdeklerin ana görevleri üzerinde borulama yaparak daha fazla işlem yapmalarını sağlamaktır.

Donanım yapısı işlemci gibi sabit olmayan kullanıcı tarafından belirlenen yongalar da mevcuttur. Bu yapıların en basiti PLD (Programmable Logic Device) temelde Şekil 3.1a daki yapısı ile belli müdahalelerle iç yapısında istenen kombinasyonel mantık devresinin oluşturulabildiği bir devredir [23]. Tek tek mantık kapılarını dizmek yerine mevcut şablon içinde ki bazı yolları silerek belli mantıksal ifadeler gerçekleştirilir.



(a) PLA

(b) LUT-Look Up Table

Şekil 3.1: Programlanabilen bazı mantık devreleri

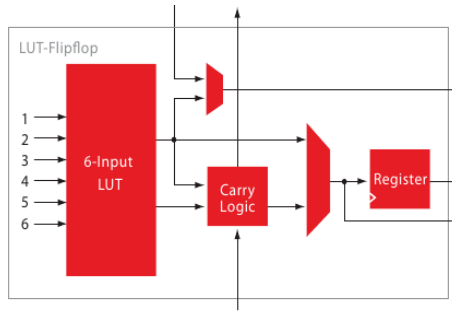
Şekil 3.1a’da bağlantılar donanımsal olarak işlenir, bu tek seferlik programlamadır. Ancak bu işlemin daha pratik yöntemi Şekil 3.1b deki gibi LUT kullanmaktır [24]. LUT her bir farklı giriş değeri için hafızasına işlenmiş veriyi çıkışa ilettiğinden hafızaya hangi işlemin çıktıları kaydedilmişse o işlevi yerine getirmektedir. Bu yapılar iç yapılarında görülen LUT-Mask kısmında kullanılan hafızaya göre çeşitlilik göstermektedir.

LUT yapısında ki hafıza kalıcı özellik taşıyorsa sisteme bir defa yazılır ve değiştirilinceye kadar sistemde kalır, buna örnek CPLDs (Complex Programmable Logic Devices) yapılarıdır; bu yongalarda hafızalar flash yapıdadır ve enerji kesilse bile bilgi saklı kalmaktadır. Ancak flash transistörlerinin çok fazla alan kaplaması sebebiyle her zaman kullanılmamaktadırlar. Bir diğer örnek ise FPGA yongalarıdır; bu yongalarda ki LUT hafızası temelde SRAM (Static RAM) hafızasıdır ve flash yapılarından daha az yer kaplamaktadır. Fakat bu yapılarda LUT hafızası geçicidir ve enerji kesintilerinde silinir, bu sebeple sistem her açıldığında bilginin tekrar yüklenmesi gerekmektedir. FPGA yongalarının bulunduğu devrelerde sistemin her başlatıldığında açılış sekansını yürütüp bilgiyi FPGA yongasına yazan bir seri programlayıcı yonga bulunmaktadır.

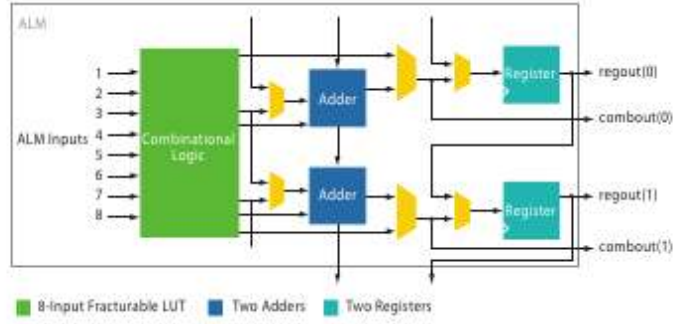
CPLD veya FPGA yongalarında fazlaca kullanılan bazı işlevler standart hale geldikçe üretici tarafından hazır yongalara dönüştürülebilmektedirler. Bu tür yapılara ASSPs (Application-Specific Standard Parts) denir ve bu yapılar belli görevler için özel işlevler barındıran uygulamaya özel entegre devrelerdir. Bir grup yonga ise ASICs (Application-Specific Integrated Circuits) olarak adlandırılır, bu yongalar üreticinin genele değil sadece belli bir ya da bir grup tüketici için ürettiği özel uygulama entegreleridir.

FPGA çipleri daha önce bahsedilen LUT yapılarıyla birlikte bazı flip-flop ve mantık devreleri barındıran özel mantıksal birimlerden oluşmaktadır. Bu birimleri birbirlerine bağlayan iç bağlantı yolları ve bunlara ait kontrol devreleri bulunmaktadır. Bu bağlantı yolları ayrıca mantık birimlerini çıkış birimlerine de yönlendirmektedir. Kısaca FPGA içerisinde mantık birimleri, bağlantı yolları ve giriş çıkış birimleri barındırmaktadır. Farklı gerilimdeki çevre ekipmanları ile bağlantının yapılabilmesi için bazı giriş-çıkış birimlerinin besleme hattı da ayrılmıştır.

Her mantıksal birim bir veya iki LUT ile belli sayıda mantıksal elemandan oluşmaktadır. Aşağıda Şekil 3.2 de bu yapılardan bazıları kabaca gösterilmiştir [24].



(a) 6 girişli LUT ile bir LE

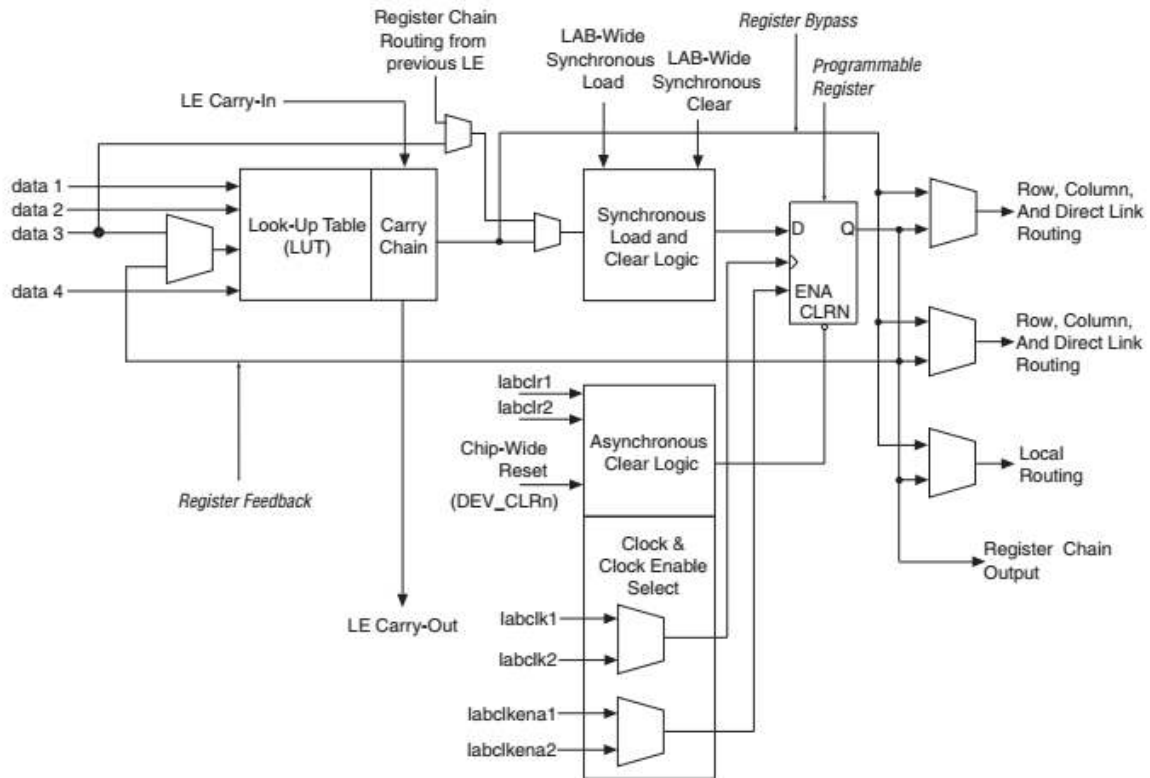


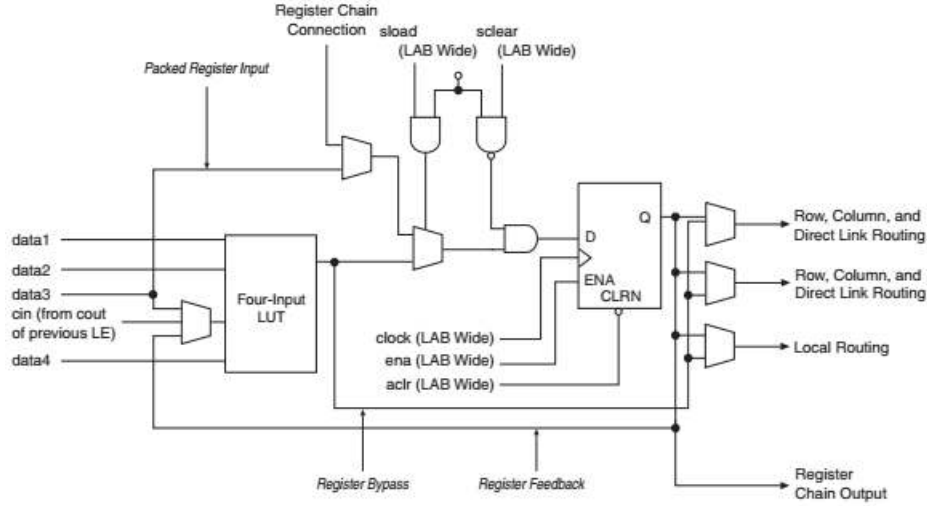
(b) 8 girişli LUT barındıran bir ALM

Şekil 3.2: Altera firmasının kullandığı çeşitli mantık birimleri (LE)

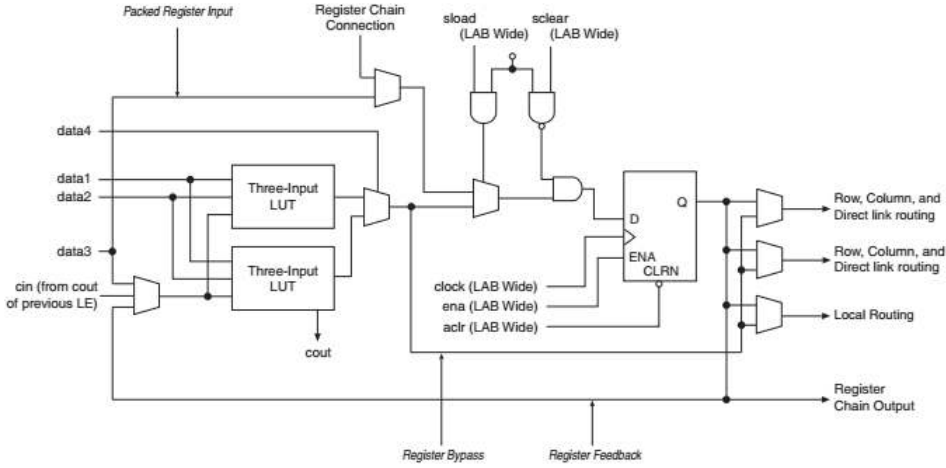
İşin temelini oluşturan bu mantık birimlerini firmalar kendilerine göre isimlendirmişlerdir. Örneğin Xilinx firması kendi mantık birimlerine “Logic Cell (LC)” olarak tanımlarken Altera firması “Logic Elements (LE)” olarak isimlendirmiştir. Bir FPGA yongasının en temel kapasite bilgisi barındırdığı bu mantık birimlerinin sayısıdır. Bunların dışında FPGA içerisinde bulunan dahili RAM (Random Access Memory) birimleri, hafıza yazmaçları, donanımsal çarpma birimleri ve PLL (Phase Locked Loop) gibi birimler ile birlikte bazı yongalarda donanımsal olarak yapılandırılmış ARM çekirdekleri de bulunmaktadır. Bu yongalar SOC (System On Chip) olarak bilinmektedir.

Daha detaylı bir lojik birim şeması Altera firması LE birimleri için aşağıda Şekil 3.3 de verilmiştir [25].





Şekil 3.4: Normal konfigürasyona sahip bir LE



Şekil 3.5: Matematiksel işlem için konfigüre edilmiş bir LE

Yukarda Şekil 3.4 ve 3.5 te aynı LE için farklı konfigürasyon modlarından örnekler gösterilmiştir [25]. Daha ilkeri kısımlarda konfigürasyon yapısının Quartus II programından alınan görüntüleri de mevcuttur.

3.2. FPGA ile Yapılmış Çeşitli Uygulamalar

FPGA ile donanımsal olarak hızlandırılmış çeşitli nesne algılama sistemleri ve geleneksel sistemlerle karşılaştırmaları literatürde görülebilmektedir.

FPGA ile hızlandırma sayesinde Intel i5 işlemcinin saniyede 2,5 kare işleyebildiği algoritmanın başarımını saniyede 60 kare değerine çıkararak gerçek zamanlı sistemler için uygun hale getiren örnek bir çalışmada trafikte “DUR” levhası gibi uyarı levhalarının gerçek zamanlı işlenebilmesi amaçlanmıştır [4]. Çalışmada Xilinx firmasına ait Kintex 7 serisi

üst seviye yüksek hızlı bir FPGA kullanılmış olup, 800x600 çözünürlük değeri için çalışma yapılmıştır.

Önerilen sistemde alınan sisteme aktarılan RGB görüntü sinyali *Integral Image Generator* adında bir modül vasıtasıyla önce gri seviyeye sonrada integral imaja dönüştürüldükten sonra bu imge parçasını hem hafıza kontrol birimine hemde imgede tanımlanmış bölgeleri bulmaya çalışan *Interest Point Detector* adıyla oluşturulmuş başka bir modüle göndermektedir.

Interest Point Detector modülü aldığı imge parçasında SURF (Speed Up Robust Features) dedektörü ile tanımlanmış bölge aramaktadır. Bu arama için görüntüde *Hessian* matrisi yardımıyla verilen noktanın etrafında x ve y eksenlerinde imge parçasının ikinci derece Gauss türevi ve konvolüsyon yardımıyla görüntüdeki anahtar imgeleri bulmaya çalışmaktadır. Bulduğu bölgelerin konum ve ölçeklerini kaydetmesi için dinamik hafıza modülüne aktarmaktadır.

Dinamik hafıza kontrolcüsü, modüllerin hafıza erişim isteklerini ve hafıza entegrelerinin kontrollerini yürütmektedir. Hafıza modülünden bilgileri okuyan *FREAK Descriptor* modülü anahtar noktalara ait hesaplamaları gerçekleştirmektedir. FREAK (Fast Retina Keypoint) modülü biyolojik sistemlerdeki retinada bulunan örnekleme ağından esinlenerek yapılmıştır. Üretilcek anahtar imge tanımlayıcısının gürültüye bağışık olmasını amaçlayan bir yapıdır. Daha sonra bu tanımlayıcı bilgi karşılaştırma için *Descriptor Matching* modülünde bilinen tanımlamalar ile eşleştirilmeye çalışmaktadır. Eşleştirme sonucu sistemin ana çıkışını oluşturmaktadır.

Sistemin normal bilgisayarlar ile gerçekleştirilen uygulamaları, işlemlerin kompleksliği ve hafıza başvurularının yoğunluğu sebebiyle çok uzun sürmektedir. Ancak önerilen sistemde yapılan deneyler çok yüksek hız artışları göstermiştir.

Başka bir uygulama örneğinde FPGA hızlandırma sayesinde geleneksel yöntemlere göre neredeyse 5 katına varan hız artışları %75 civarında enerji tasarrufu ile bile sağlanabilmiştir [5]. Üst seviye Xilinx Virtex-5 yongası ile yapılan araştırmada geleneksel paralel işlem ile yürütülen kayar pencere yaklaşımı ile görüntü bulma işleminden 320x240 örnek çözünürlük değerlerinde 4,9 kat daha hızlı sonuç vermiştir.

Üç temel modül olarak ECC (Edge Computation Core - Kenar İşleme Çekirdeği), WEU (Window Extraction Unit - Pencere Çıkarma Birimi) ve CE (Clasification Engine - Sınıflandırma Motoru) modüllerinin yanı sıra harici hafıza kontrol modülü ile hafıza erişimlerini ve

senkronizasyonu kontrol etmektedir. Sistem örüntü arama hızını arama uzayını kısıtlayarak hızlandırmış ve önceki örneklerinin üzerinde bir performansla çalışmıştır. Önceki örnekte Xilinx Virtex-5 yongası ile Intel 2.4GHz Core-2 Quad sisteminde SSE (Streaming SIMD (Single Instruction, Multiple Data) Extension) optimizasyonu yapılmış yazılıma göre 100 kat fazla hız artışını sağlayıp 80x60 çözünürlük için 625 FPS değerlerine ulaşılmıştır [26].

Ölçeklenebilir nesne algılama hızlandırıcılarının FPGA üzerinde uygulamasına örnek olarak verilebilecek bir başka çalışmada ise farklı FPGA yongalarına uyarlanabilecek şekilde bir otonom kodlayıcı üretilmiştir [6]. Bu kodlayıcı farklı FPGA ve çerçeve boyutları için uygun kodları üreten bir sistem olarak tasarlanmış olup, deneylerde 3.0 GHz Intel Pentium 4 işlemcili uygulamalarına göre %60 ila %250 gibi performans artışları sağlanmıştır.

Önerilmiş sistemde FPGA üzerinde Viola-Jones algoritması kullanılmaktadır [27]. İşlem 20x20 boyutunda ki arama penceresi kullanarak 320x240 boyutundaki çözünürlükte farklı ölçeklemelerde *Haar*-özelliklerini arayarak yapılmaktadır. Sistemde kullanılan donanım kaynakları sınıflandırıcı boyutu ile artarak büyümekte, ve daha büyük yapılar için farklı FPGA yongaları gerekmektedir.

Düşük seviye FPGA yongaları üzerinde yapılan uygulamalara örnek olarak trafik gözetleme sistemleri için FPGA temelli hareketli nesne yakalama algoritması gösterilebilir [7]. Önerilen sistem temelde arka plan çıkarımı, kenar bulma ve gölge bulma tekniklerini kullanarak çalışmaktadır. Xilinx Spartan6 üzerine VHDL (VHSIC (Very High Speed Integrated Circuit) Hardware Description Language) dilinde kodlanarak gerçekleştirilen uygulamada arka plan çıkarımı selektif ve selektif olmayan farklı iki blok ile yapılarak daha hassas ve doğru olması amaçlanmıştır. Kenar yakalama modülünün ardından konulan gölge yakalama modülü ile hatalı olarak yakalanan kenarlar giderilmeye çalışılmıştır.

Nesne bulma uygulamaları için FPGA ile yapılmış pixel işleme ünitesi örneği olarak yapılmış başka bir uygulamada ise DSP FPGA karşılaştırması yapılmaktadır [8]. Sistem dağıtık RAM mimarisi kullanarak nesne bulma sistemlerinin temel bileşenlerinden pixel işleme ünitesini ele almaktadır. Çalışma önceki bir çalışmada kullanılan programlanabilen TMS320C44 DSP işlemcisi ile yapılmış bir örnek ile birden fazla hatta RAM blokları barındıran bir Xilinx Virtex serisi FPGA yongasını karşılaştırmaktadır. Bağımsızca çalıştırılan RAM blokları sayesinde sistem daha yüksek bir paralel işlem kapasitesi elde etmiş olmasıyla diğer çalışmalardan ayrılmaktadır.

Mukayesesi yapılan DSP sistemi 30 MHz saat ile 320x400 çözünürlük ve 25~30 FPS

hızında çalışabilen bir adanmış görüntü işlemcisidir. Kullanılan FPGA sistemi ise bağımsız erişimli statik hafıza blokları ile 100 MHz hızında çalışan bir geliştirme kartıdır. FPGA için Handel-C dili kullanılmış olup bu dil C diline çok yakın bir donanım tanımlama dilidir. FPGA kartı PCIe (Peripheral Component Interconnect Express) yuvasından bilgisayara bağlanarak çalıştırılmıştır.

Çalışma sırasında, FPGA çipi devreye girmeden önce ana bilgisayar mevcut ve arka plan görüntüleri dahil olmak üzere 16 çerçeveyi harici statik ram bloklarından 0 ve 1 nolu bloklara yazmaktadır. Hafızanın 2 ve 3 nolu bloklarında ise işlemler sırasında kullanılan indirgenmiş ara çerçeveler ve bu çerçevelere ait bazı bilgiler saklanmaktadır. DSP sisteminin dinamik ram kullanma dezavantajı bulunmakla birlikte 100 MHz hızında çalışan FPGA kiti 2 ms altında bir süre ile işlemi bitirmiştir. TMS320C44 entegresi 30 MHz saat ile işlemi 77,4 ms sürede tamamlarken aynı DSP ailesinin o zamanki en iyi serisi TMS320C64x yongası 1.1 GHz saat sinyali ile işlemi ancak 3,15 ms süresinde bitirebilmektedir.

Bu tezde kullanılan FPGA geliştirme kartı ve kamerası ile yapılan bir çalışmada ise insansız hava araçları için FPGA temelli gerçek zamanlı hareketli nesne algılama sistemi önerilmiştir [9]. Sistemde ayrıca Nios II soft-processor 100 MHz saat sinyali ile kullanılmıştır. Görüntünün hava aracından yer istasyonuna gönderilmesinin sebep olacağı iletişim darboğazının giderilmesine yönelik olarak; esnek, hafif ve düşük güç gereksinimi olan bir seçenek olarak öne sürülen sistem saniyede 30 kare ve 640x480 çözünürlük ile çalıştırılmıştır.

Arka plan çıkartımı işlemi sabit kamera olmaması ve İHA'nın kendi hareketlerinden kaynaklanacak sorunlar sebebiyle parlaklık temelli yapılmıştır. Sistem temelde ardışık iki gri seviye çerçeve bilgisi üzerine çalışmakta, ve RANSAC (Random Sample Consensus) ile İHA'nın ihmal edilebilir düzeydeki öz hareketleri bulunup elimine edilmektedir. Ardından iki ardışık görüntünün farkının bir eşik değeriyle birlikte işlenmesi ile binary(ikili) görüntü elde edilmiştir. Binary görüntü üzerinde yapılan medyan filtreleme ve dilatasyon işlemlerinin ardından hareketli nesneler bulunmaya çalışılmıştır.

Nios II işlemcisi üzerinde çalıştırılan Linux işletim sistemi vasıtası ile RANSAC işleminin donanım olarak hızlandırılmamış kısmı başarılmıştır. Ancak Nios çekirdeğinin oldukça düşük olan hızı gerçek zamanlı sistem için sorun oluşturmuştur. Bu sebeple RANSAC işleminin donanım hızlandırmaya uygun kısımları Nios ile değil doğrudan donanımla başarılmıştır. RANSAC tarafından tespit edilen öz hareket miktarı daha sonra arka plan çıkartımı

işleminde kullanılmak üzere segmentasyon bölümüne mevcut çerçeve bilgisi ile birlikte aktarılmaktadır.

Sistemin ardışık çerçeveler gerektirmesi ve SDRAM belleğin video tamponu olarak kullanılması sebebiyle SRAM hafıza modülü ardışık çerçeveleri tutmak için kullanılmıştır. Sistemin gerçek zamanlı çalışmasını daha fazla kısıtlamamak için Nios çekirdeği yalnızca RANSAC işleminin bir kısmı için kullanılmış. Intel Core i5 sistemine kıyasla 2,29 kat, ARM işlemci barındıran örneklerle göreyse 53,57 kat hız artışı sağlanmıştır. FPGA yongasının kaynaklarının %20 kadarı kullanılmış olması kalan kısmın diğer uygulamalar için nedenli müsait olduğunu göstermektedir. Ancak belirtmek gerekir ki İHA sistemlerinde çok daha yüksek çözünürlüklü kameralar bulunur ve bunların görüntü işlemleri çok daha fazla kaynak harcayacaktır.

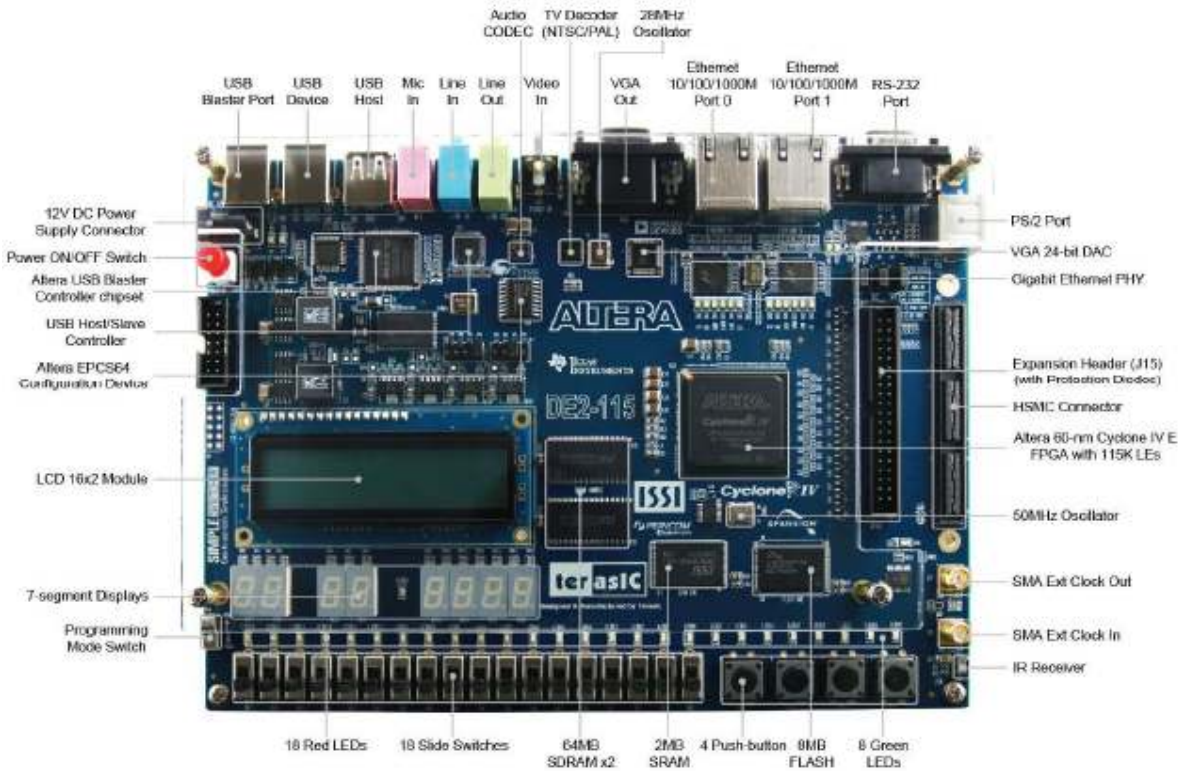
4. FPGA ile NESNE ALGILAMANIN GERÇEKLEŞTİRİLMESİ

4.1. Kullanılan Donanım Birimleri

Sistemde Terasic DE2-115 FPGA geliştirme kartı, TRDB-D5M CMOS kamera kiti, parlaklığı ayarlana bilen bir led aydınlatma lambası ve VGA bağlantısı bulunan bir LCD monitör kullanılmıştır. Şekil A.3 'te görülen sistemin programlanması USB (Universal Serial Bus) üzerinden yapılmıştır.

4.1.1. Terasic DE2-115 FPGA geliştirme ve eğitim kartı

Terasic firmasını ürettiği DE2-115 FPGA geliştirme ve eğitim kartı, üstünde bütünleşik çok sayıda çevre birim ile donatılmış FPGA temelli DE (Development and Education) serisi kartlardan biridir. Şekil 4.1 den görülebileceği gibi kartın geniş çevre birim yelpazesi onu çeşitli uygulamalar için elverişli kılmaktadır [28].



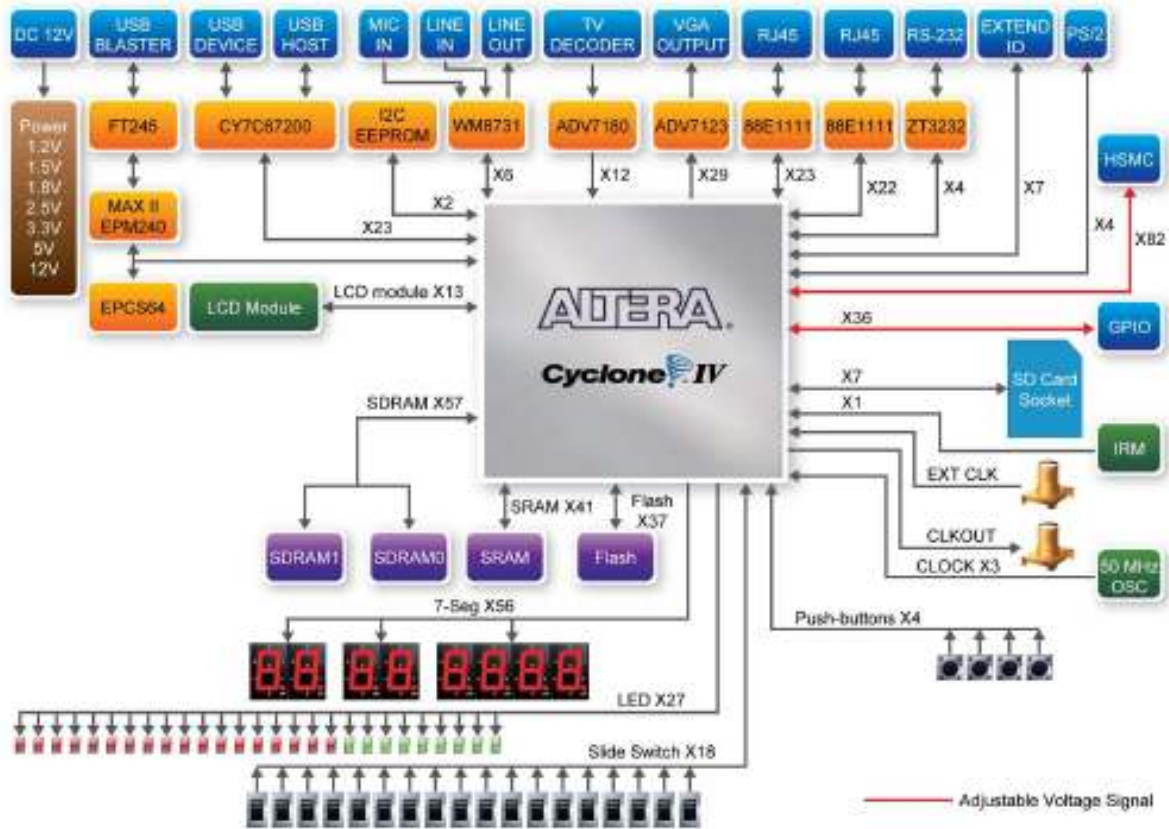
Şekil 4.1: Terasic DE2-115 eğitim kartı ön görünüşü

Kartın temel bileşeni Altera Cyclone® IV EP4CE115 FPGA yongasıdır. Bu yonganın tam kodu EP4CE115F29C7N olup sırasıyla temel özellikleri kodlamadan okunabilmektedir;

EP4C : Cyclone IV
 E : Enhanced logic/memory (Geliştirilmiş mantıksal yapı)
 115 : 114.480 LEs (Mantıksal birim sayısı)
 F : FBGA (FineLine Ball Grid Array)
 29 : 780 pin
 C : Commercial temperature ($T_j = 0^{\circ}\text{C}$ to 85°C) (Çalışma sıcaklığı)
 7 : Speed Grade 7 (gecikme katsayısı)
 N : Lead-free Pack

Yonganın temel özellikleri dışında ek olarak 3.888 Kbit dahili RAM hafızası, 266 adet 18x18 dahili donanımsal çarpma ünitesi, 4 adet PLL, 20 adet genel saat sinyali ağı, 528 adet I/O teminali bulundurmaktadır.

FPGA yongalarının programları kalıcı olmadığından kart üzerinde bir adet EPCS64 seri konfigüratör entegresi bulunmaktadır. Ancak deneyler sırasında tüm konfigürasyon dosyaları SW19 anahtarı RUN moduna alınarak doğrudan FPGA belleğine yazılmıştır.

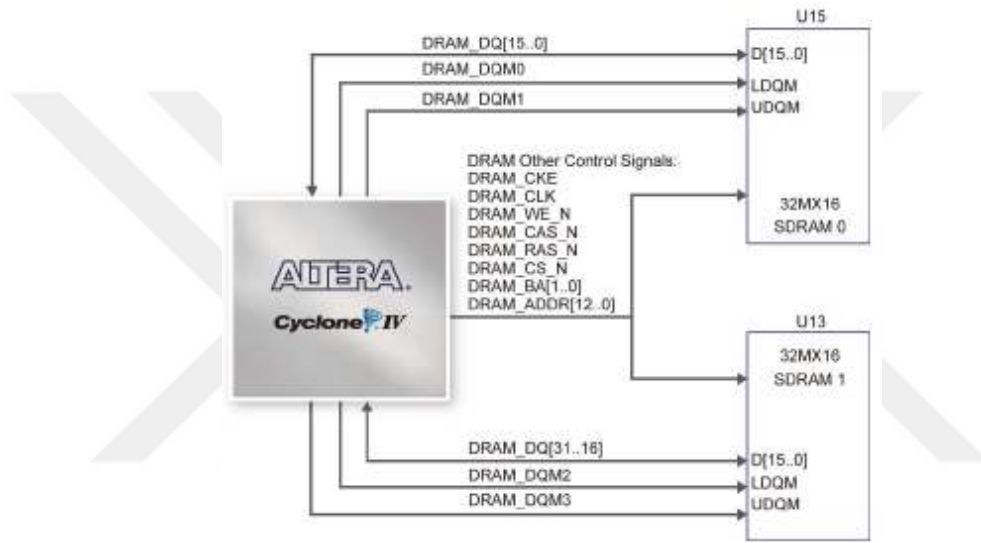


Şekil 4.2: Terasic DE2-115 eğitim kartı blok diyagramı

Kart üzerindeki çevresel donanımlar FPGA yongasına Şekil 4.2 de görüldüğü gibi bağlanmıştır [28]. Şekilde her donanıma bağlantı için kullanılan hat sayısı ayrıca üzerlerinde belirtilmiştir. Bilgisayardan bağlantı ve programlama için Altera firması USB-Blaster adıyla

kendine ait bir teknoloji kullanmaktadır, bu kartta bu dahili olarak bulunmaktadır. Ayrıca JTAG bağlantı girişi de kart üzerinde mevcuttur.

Kart üzerinde FPGA yongasına bağlı en önemli yapılar genişleme yuvaları ve hafızalardır. Kart üzerinde birkaç çeşit hafıza entegresi bulunmaktadır. Bunlar sırasıyla 2 adet 32Mx16 bit genişliğinde SDRAM entegreleri ile toplamda 128 MB'lık dinamik hafıza, 1 adet 1Mx16 bit genişliğinde SRAM ile 2 MB'lık statik hafıza, 4Mx16 bit genişliğinde Flash hafıza ile 8 MB'lık flash hafıza ve 32 kbit'lik EEPROM hafıza bulunmaktadır. Dinamik hafıza entegrelerinin FPGA bağlantı şemaları Şekil 4.3 te görülmektedir [28].



Şekil 4.3: Terasic DE2-115 eğitim kartı SDRAM bağlantısı

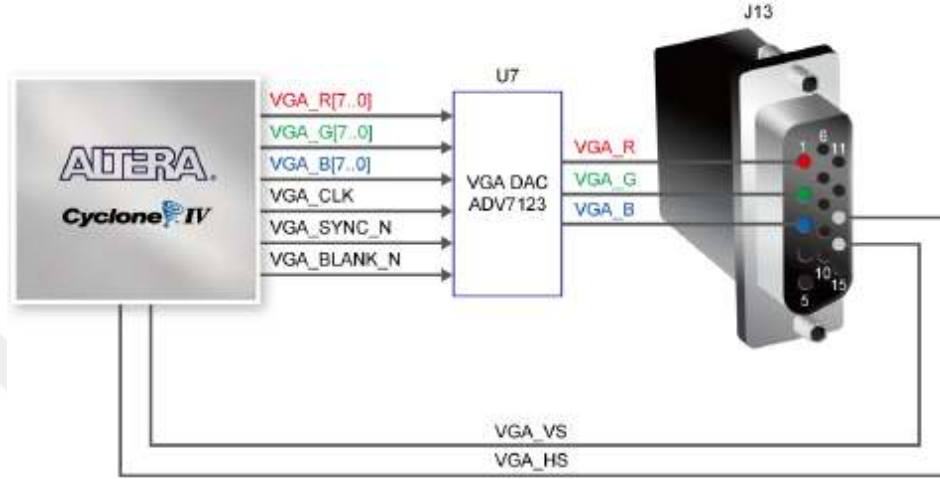
Gösterge ve anahtar takımı olarak 18 adet seçme anahtarı, 4 adet basmalı buton (güçlütleri donanımsal olarak bastırılmış), 18 adet kırmızı LED (Light Emission Diode), 9 adet yeşil LED ve 8 adet 7-segment display bulunmaktadır. Bunların dışında bir adet 16x2 LCD (Liquid Crystal Display) modül bulunmaktadır.

Kart üzerinde saat sinyali için kullanılan 50 MHz lik bir kristal ve harici saat sinyali alıp verebilmek için birer adet SMA (SubMiniature version A) bağlantı noktası bulunmaktadır.

Kartın güç girişi 12 VDC olup entegreler için gerekli gerilimler konvertörler yardımıyla dönüştürülerek üretilmektedir. Şayet bağlanan çevresel birimler farklı lojik besleme gerilimleri ile çalışıyorlarsa HSMC (High Speed Mezzanine Card) için JP7 ve GPIO (General Purpose I/O) için JP6 kullanılarak uygun gerilimler seçilebilmektedir. Seçilebilecek gerilimler; 3.3 V, 2.5 V, 1.8 V ve 1.3 V olarak belirtilmiştir.

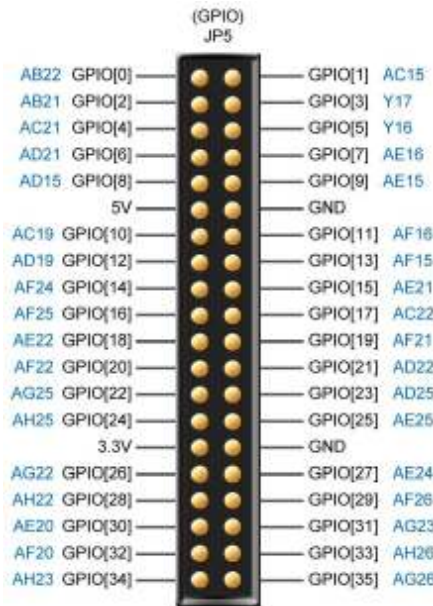
VGA (Video Graphics Array) bağlantısı yapmak için kart üzerinde bulunan 15 pinlik

D-SUB terminali ADV7123 VGA DAC entegresi vasıtasıyla FPGA yongasına bağlanmıştır. Her ne kadar VGA DAC çipi 10 bitlik olsa da çevresel bağlantıların harcadığı onca terminal sebebi ile FPGA çipine 8 bitlik olarak bağlanmıştır. Şekil 4.4 te VGA bağlantı sistemi görülmektedir [28]. VGA DAC entegresinin alt 2 LSB biti bağlı değildir.



Şekil 4.4: Terasic DE2-115 eğitim kartı VGA bağlantısı

Altera firmasının kullandığı genişleme yuvalarından önemli ikisi bu kartta birer adet olmak üzere mevcuttur. Kart üzerinde 1 adet GPIO ve 1 adet de HSMC genişleme yuvası bulunmaktadır.



Şekil 4.5: Terasic DE2-115 eğitim kartı GPIO bağlantısı

Sistemde kullanılan GPIO genişleme yuvası diyot korumalı olup 40 adet pine sahiptir.

Şekil 4.5 ile görüldüğü üzere bu pinlerden ikisi GND biri +5V ve bir tanesi de +3.3V olarak bağlanmıştır [28]. Bu 4 adet güç pininden sonra geriye kalan 36 adet sinyal ucu hem diyot korumalıdır, hemde bağlanacağı cihazın gerilim seviyesine uygunlaştırma için gerilim seviyesi JP6 ile ayarlanabilmektedir. Varsayılan olarak GPIO sinyal uçları 3.3V gerilime ayarlanmıştır. Kullanılan kamera modülü de aynı gerilimi kullandığından deneylerde değişiklik yapma ihtiyacı doğmamıştır.

4.1.2. Terasic D5M CMOS kamera kiti

Mevcut Terasic DE2-115 geliştirme kartı ile GPIO yuvasından bağlantı kurabilecek şekilde tasarlanmış multimedya ek kartlarından TRDB-D5M kamera kiti kendinden önceki kamera modüllerinden daha yüksek çözünürlükte olduğundan ve bir önceki kart çeşidi üretimden kalktığından bu çalışma için seçilmiştir.



Şekil 4.6: Terasic TRDB-D5M 5 megapiksel kamera kartı

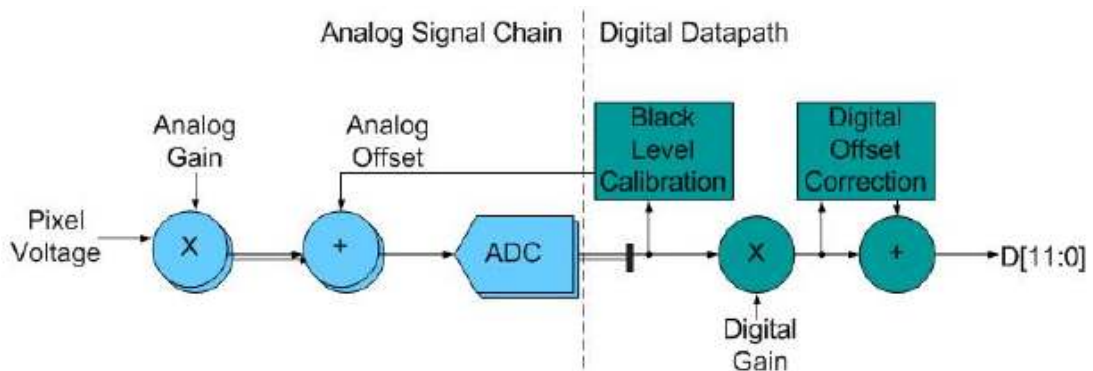
Yukarıda Şekil 4.6 da Terasic firmasının ürettiği TRDB-D5M 5 megapiksel dijital kamera geliştirme ve eğitim kartının ön görünüşü görülmektedir [29]. Bu kit, üzerinde görüntü yakalamak için bir CMOS algılayıcı barındırmaktadır. Kit, DE2-115 kartına GPIO üzerine doğrudan ya da beraberinde gelen paralel uzatma kablosu aracılığı ile bağlanabilmektedir. Kartın seçilmesindeki etmenlerden biri görüntünün sensörden çıktığı gibi çıkışa verilmesi ve herhangi bir formata dönüştürülmemiş olmasıdır.

D5M kamera kiti otomatik karanlık seviyesi kalibrasyonu yapabilir, kameranın kazanç, çerçeve ebadı, ve hızı ile pozlama süresi ayarlanabilmektedir. Düşük aydınlatmalı ortamlarda iyi performans göstermesi, düşük karanlık akımı, dahili PLL, I2C arayüzü ile programlanabilmesi ve nispeten hızlı bir sensör barındırması diğer olumlu özellikleri olmuştur. Ayrıca kamera görüntüyü yatay ya da dikey olarak aynalayabilmekte, satır sütun atlayabilmekte ya da bindirebilmekte ve kullandığı global shutter ile bazı görüntü bozulmalarını engelleyebilmektedir. Diğer başlıca özellikleri Tablo 4.1 ile gösterilmiştir [29].

Parameter		Value
Active pixels		2,592H x 1,944V
Pixel size		2.2 μ m x 2.2 μ m
Color filter array		RGB Bayer pattern
Shutter type		Global reset release (GRR),
Maximum data rate/master		96 Mp/s at 96 MHz
Frame rate	Full resolution	Programmable up to 15 fps
	VGA (640 x 480)	Programmable up to 70 fps
ADC resolution		12-bit
Responsivity		1.4 V/lux-sec (550nm)
Pixel dynamic range		70.1dB
SNRMAX		38.1dB
Supply Voltage	Power	3.3V
	I/O	1.7V~3.1V

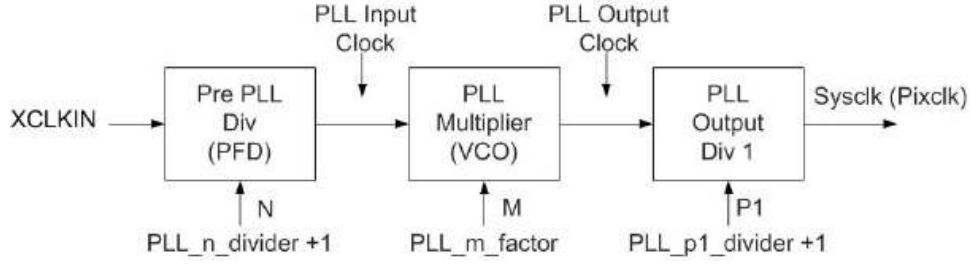
Tablo 4.1: Terasic TRDB-D5M 5 megapiksel kamera temel özellikleri

TRDB-D5M dahili olarak analog ve dijital olarak kazanç kontrolü yapabilmektedir. Bununla ilgili ilkesel şema Şekil 4.7 de verilmiş olan sinyal hattından anlaşılabilmektedir [30].



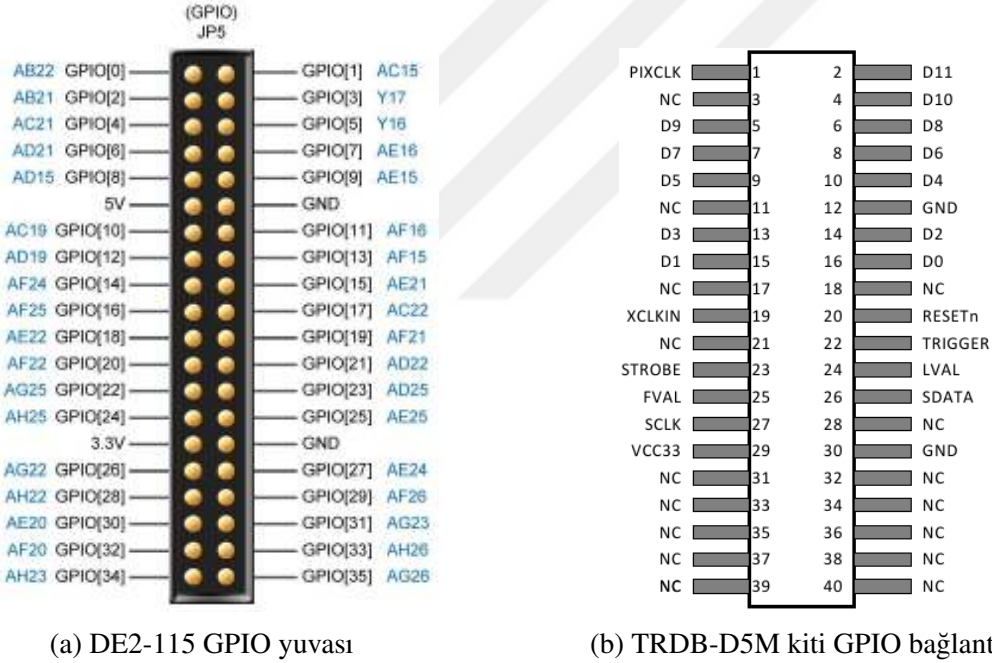
Şekil 4.7: Terasic TRDB-D5M 5 megapiksel kamera sinyal hattı

TRDB-D5M dahili PLL sayesinde saat sinyalini ayarlayabilmektedir. D5M dahili PLL yapısı Şekil 4.8 de görülebilmektedir [30].



Şekil 4.8: Terasic TRDB-D5M 5 megapiksel kamera kiti PLL sistemi

TRDB-D5M kamera kiti DE2-115 geliştirme kartına GPIO genişleme yuvası üzerinden bağlanmaktadır. Aşağıda Şekil 4.9 ile iki kartın ilgili terminallerinin pinleri karşılıklı olarak görülecek şekilde yan yana gösterilmiştir [28, 29]. Genişleme yuvasından sağlanan +5VDC gerilim kart tarafından kullanılmamaktadır.

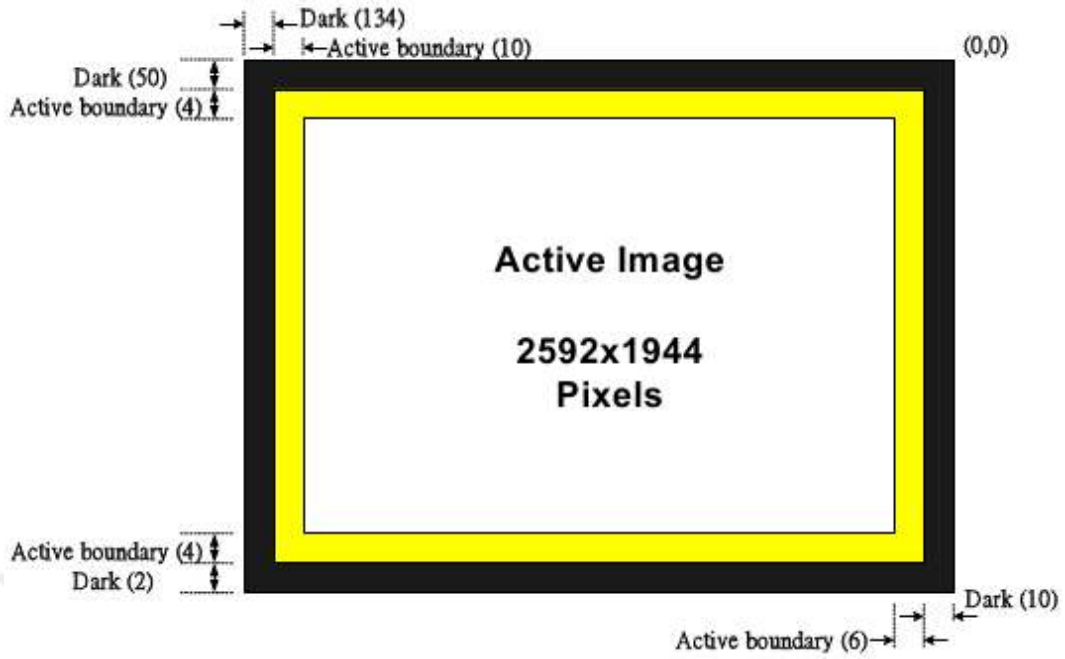


(a) DE2-115 GPIO yuvası

(b) TRDB-D5M kiti GPIO bağlantısı

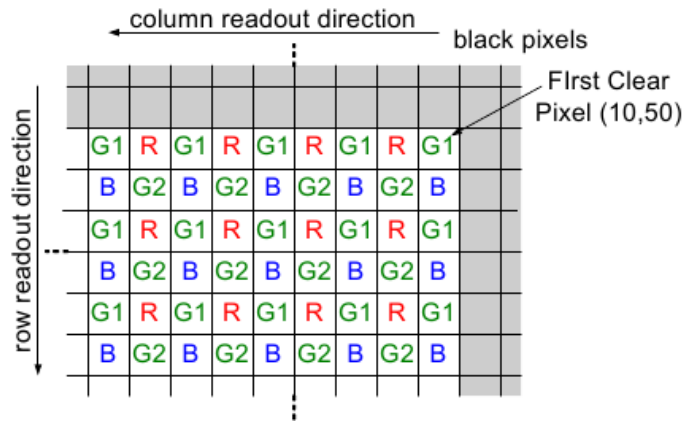
Şekil 4.9: DE2-155 ve TRDB-D5M karşılıklı GPIO bağlantıları

Kamera sensörü aslında 2751x2003 'lük bir piksel ağı barındırmaktadır, ancak Şekil 4.10 da görüleceği üzere bu alanın dış kısımlarında aktif resim bölgesine dahil olmayan karanlık pikseller ve aktif pikseller mevcuttur [30]. Bu pikseller hariç tutulduğunda görüntü için aktif piksel boyutu 2592x1944 boyutuna düşmektedir.



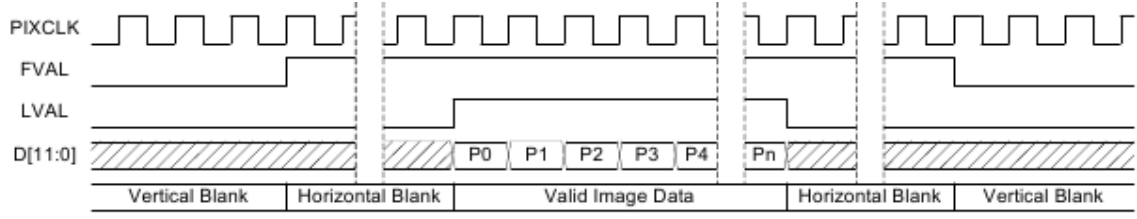
Şekil 4.10: Terasic TRDB-D5M 5 megapiksel kamera sensör detayı

Kamera kitinin sensörü Bayer mozaik filtresi bulundurmaktadır. Şekil 4.11 de görülen sağ üstteki ilk piksel 1. yeşil olup bu satırda sadece Yeşil 1 ve Kırmızı bulunmaktadır [30]. Alt satırda ise Mavi ve Yeşil 2 bulunmaktadır. Görüldüğü gibi sensörün piksellerinin yarısı yeşil diğer yarısı ise eşit olarak mavi ve kırmızı pikseller içermektedir.



Şekil 4.11: Terasic TRDB-D5M 5 megapiksel kamera sensörü mozaik filtresi

Kamera görüntüyü aktarırken renkleri ayırmadan satır satır sırasıyla ham olarak göndermektedir. Kamera görüntü bilgisinin geçerli olduğunu satır ve çerçeve geçerli sinyallerinin belli kombinasyonları ile alıcı sisteme aktarmaktadır. Şekil 4.12 de görüldüğü üzere ham olarak alınan bilgilerin işlenerek RGB formunda görüntü oluşturulması gerekmektedir [30].



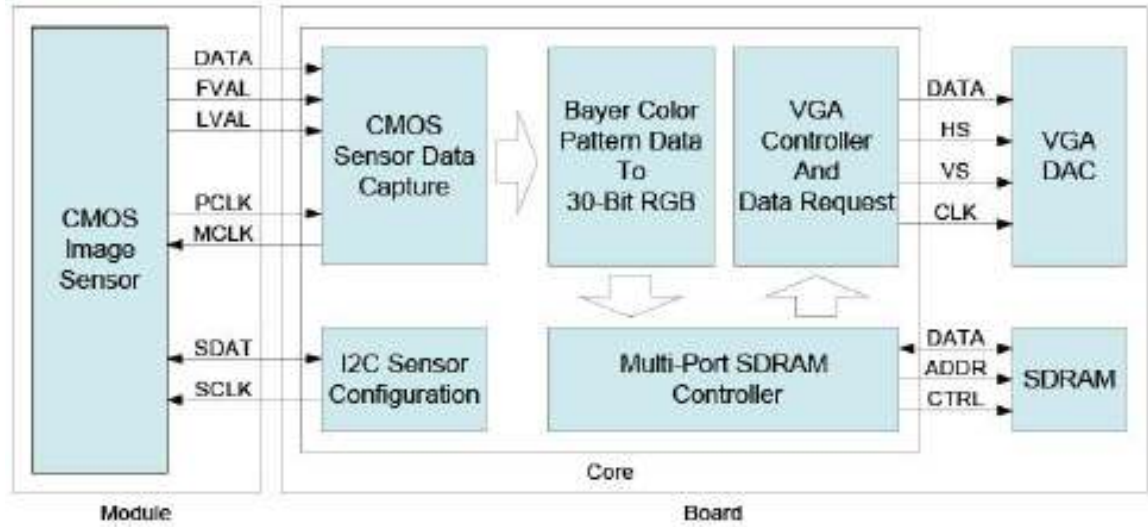
Şekil 4.12: Terasic TRDB-D5M 5 megapiksel kamera ham görüntü sinyali

Şekil 4.12 ile de görüleceği üzere görüntü sinyalinin geçerli olduğunu kabul etmek için hem FVAL (Frame Valid) hemde LVAL (Line Valid) sinyallerinin aktif olması gerekmektedir. Bu şartlar oluşmadan veri hattı üzerindeki sinyalin bir anlamı bulunmamaktadır.

4.2. Kullanılan Referans Görüntüleme Sistemi

4.2.1. Referans sistemin genel yapısı

Mevcut sistem temelde görüntüyü kameradan çekerek temel işlemlerden sonra VGA çıkışında görüntüleme birimini sürmektedir.



Şekil 4.13: Referans Görüntüleme sistemi blok diyagramı

Şekil 4.13 ile görüleceği üzere sistem belirli bloklar halinde kurulmuştur [30]. Core olarak gösterilen FPGA yongası içinde görülen bölümler ise donanım olarak FPGA içinde sentezlenen yapılardır. Daha detaylı sistem bilgisi için Şekil A.4 incelenebilir. Burada sistemin çalışması için ilgili modüllere çeşitli hizmetler sunan bazı birimler gösterilmemiştir. Şekil A.1 de sistemin FPGA yongası üzerinde ki fiziksel yerleşimi görülebilmektedir.

4.2.2. Referans sistemin genel çalışma prensibi

Sistem öncelikle başlangıç anında kameranın açılış sekansını idare ederek onu konfigüre eder, kamera gerekli ayarlamalardan sonra istenen konfigürasyonda çalışmaya ve sinyal göndermeye başlamaktadır. Kameranın konfigürasyonunu FPGA içinde sentezlenen I2C modülü yapmaktadır. Kameradan gelen görüntüleri yakalayan birim bu yakaladığı *geçerli* verileri RGB şekline dönüştürecek olan üniteye iletmektedir. RAW2RGB modülü ham veriyi alarak RGB sinyaline dönüştürmektedir. Buraya kadar ki sistem sadece kamerayı çalıştırıp ham verisini RGB sinyaline çevirmektedir. Bir başka sistem de görüntüyü VGA kontrolcüsü üzerinden VGA DAC entegresine göndererek izlenebilmesini sağlamaktadır. Ne varki bu iki temel bölüm farklı frekanslarda çalıştıklarından birbirleri ile eş zamanlı çalışmamaktadırlar. Bu problemin üstesinden gelebilmek için farklı iki frekanstaki sistemlerin toplamı bir hızda çalışan dinamik bellek sistemi kullanarak iki sistemi birbirine video tamponlama yöntemi ile bağlanmıştır.

Referans sistemde varsayılan olarak kontrol girişleri için belli bazı butonlar ve anahtarlar kullanılmıştır, VGA çıkışı dışında 7 segmentli göstergelerde taranan çerçeve sayısı kümülatif olarak gösterilmektedir. Kumandalarda 0. buton reset, 1. buton 0 numaralı seçme anahtarı vasıtasıyla pozlama süresi, 2. buton fotoğraf çekimi ve 3. buton fotoğraf modundan çıkıp serbest moda geri dönmeyi sağlamaktadır.

FPGA yongasında gerçekleştirilen sistemin daha açık bir şeması Şekil A.4 den de görüleceği gibi biraz daha farklıdır. Şekil A.4 incelendiğinde görülen farklı yapılardan sdram_pll modülü diğer modüllere belirlenmiş katsayılarla ayarlanmış olan saat sinyali vermektedir. Örneğin bu sistemde kamera kitine 25 MHz olarak saat sinyali gönderilmektedir. Kamera modülünde bulunan dahili PLL devresinin konfigürasyonu 25 MHz 'lik saat sinyalini alıp kendi PLL devresinde ön bölücü ile önce 6 ya böler, sonra 24 ile çarpıp ardından sinyali 2 ye böler, bu şekilde ürettiği 50 MHz saat sinyalini ana saat sinyali olarak kullanmaktadır. Ürettiği sinyali yakalayacak modül olan CCD_Capture modülü artık dışarıdan kameradan gelen bu saat sinyali ile çalışmaktadır.

CCD_Capture modülünün 50MHz frekansta çalışarak yakladığı ham görüntü sinyali FVAL ve LVAL sinyalleri vasıtasıyla geçerli olduğu kesinleştiği zaman registerlar ile man-dallanarak RAW2RGB modülüne verilir RAW2RGB modülü I2C_CCD_Config modülünün D5M kamerasını konfigüre ettiği haline göre ayarlanmış ve gelen ham veriyi zamanlamaya

bakarak RGB sinyaline dönüştürüp ardından Sdram_Conrtol modülüne iletmektedir.

Sdram_pll modülü VGA_Controller modülünü 800x600 çözünürlükte 60 kare hızda çalıştırabilmek için için $800 \times 600 \times 60 = 28.800.000$ yani yaklaşık 30MHz hızında çalıştırması gerekse de gerçekte Tablo 4.2 yatay ve dikey karartma ve senkronizasyon işaretlerinin de bu süreye dahil edilmesi sonucunda sistemi $60 \times 1056 \times 628 = 39.790.080 \approx 40\text{MHz}$ saat sinyali ile beslemesi gerekmektedir [30]. VGA_Controller modülü uygun zaman gelince çerçeve başını belirterek Ram kontrol modülünden bilgiyi sıralı olarak çeker ve VGA DAC entegresine göndermektedir. Entegre gelen dijital sinyali analog sinyale çevirip VGA çıkışını sürer böylece görüntü monitörden izlenebilmektedir.

Parametre	Değer	Parametre	Değer
V_SYNC_CYC	= 4	H_SYNC_CYC	= 128
V_SYNC_BACK	= 23	H_SYNC_BACK	= 88
V_SYNC_ACK	= 600	H_SYNC_ACK	= 800
V_SYNC_FRONT	= 1	H_SYNC_FRONT	= 40
V_SYNC_TOTAL	= 628	H_SYNC_TOTAL	= 1056

Tablo 4.2: VGA yatay ve dikey senkronizasyon değerleri

Görüldüğü üzere burada biri 50MHz ile diğeri ise 40MHz çalışan iki ayrı sistem bulunmaktadır. Bu iki sistemin beraber çalışabilmesi için arada bir video tamponlama yapılması gerekmektedir. Bu işlevi sistem dinamik hafıza kontrolcüsü yardımıyla yerine getirmektedir. Sdram_pll modülü harici dinamik hafıza entegrelerini ve onları kontrol eden modülü ayrı ayrı 100MHz saat frekansı ile beslemektedir. Dinamik belleklerin özelliklerinden kaynaklanan bir sebeple bu iki saat sinyali arasında belli bir faz farkı bulunmaktadır.

RAW2RGB modülü çıkışında ki her biri 12 bitlik 3 renk çıkışı ile toplam 36 bite ulaşan veri hattı ile 2 X 16 bitlik hafıza sistemi birbirlerine bağlanırken renk bilgilerinin düşük değerlikli alt iki bitleri ihmal edilmektedir. Böylece 10 bitlik 3 renk bilgisi sadece 30 bitlik bir genişlik ihtiyacı duymaktadır. Kontrolör her iki 16bitlik hatların MSB bitleri 0 olmak koşulu ile renk verilerini Tablo 4.3 ile gösterildiği gibi doldurmaktadır.

Veri Hattı	MSB	5-bit	10-bit
WR1DATA	= 0	G[11:7]	B[11:2]
WR2DATA	= 0	G[7:2]	R[11:2]

Tablo 4.3: Renk bilgisinin hafıza hatlarına dağılımı

Yukarıda sayılanların dışında sistemde çerçeve sayısını sayıp kümülatif olarak göstermek için sayıcı ve 7-segment göstergeleri sürmek için decoder gibi çalışan modüller de bulunmaktadır. Ayrıca sistemin sağlıklı bir başlangıç sekansı yürütebilmesi için tasarlanmış özel bir resetleme sistemi; Reset_Delay modülü bulunmaktadır.

Sistemde u8 bileşeni olarak görülen I2C_CCD_Config modülü sistem başlangıç sekansı sırasında kameraya gerekli konfigürasyon bilgilerini gönderir. Kamera, çalışması için gerekli konfigürasyon bilgilerini kendi içinde bulunan yazmaçlarda tutar, Özel amaçlı kaydediciler gibi davranan bu yazmaçların sistemin açılış sekansı sırasında gerekli bilgilerle ayarlanması gerekmektedir.

Sistemde u8 bileşeni olarak görülen I2C_CCD_Config modülünün açılış sekansında yaptığı ayarlamalar;

- Satır ve satır aynalama özelliğini aktifleştirir.
- Pozlama süresi ilk değerini atar.
- Yatay ve dikey karartma sürelerini ayarlarını yapar.
- Veri çıkışını PIXCLK (piksel saati) sinyalinin yükselen kenarına ayarlar.
- Sırasıyla yeşil1, mavi, kırmızı ve yeşil2 kazançlarını ayarlarını yapar.
- PLL çarpan ve bölücüleri ayarlayıp başlatır.
- Başlangıç satır ve sütun değerlerini atar.
- Çerçevenin satır ve sütun boyutlarını atar.
- Satır ve sütunlar için bindirme özelliğini aktifleştirir.
- Siyah seviye ayarı için değer atar.

Deneyler sırasında I2C_CCD_Config modülünün satır ve sütun başlangıç noktalarını sensörün köşe (0,0) noktasına ayarlandığı görülmüştür.

```
assign sensor_start_row    = iZOOM_MODE_SW ? 24'h010036 :  
24'h010000 ;  
assign sensor_start_column = iZOOM_MODE_SW ? 24'h020010 :  
24'h020000 ;
```

Kamera tam karşılığı değil sağ alt tarafa yönelmiş olan FOV (Field of View) tam ortalananak gelecek şekilde yeniden ayarlanmıştır.

```

assign sensor_start_row    = iZOOM_MODE_SW ? 24'h010036 :
24'h010174;
assign sensor_start_column = iZOOM_MODE_SW ? 24'h020010 :
24'h0201F0;

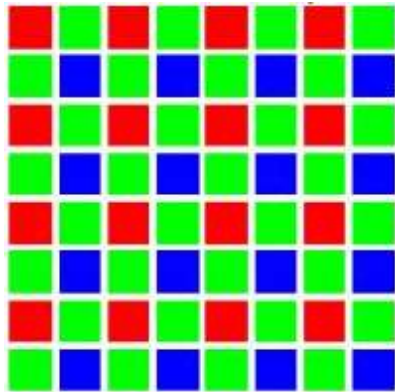
```

Ayrıca varsayılan pozlama süresi de değiştirilmiştir.

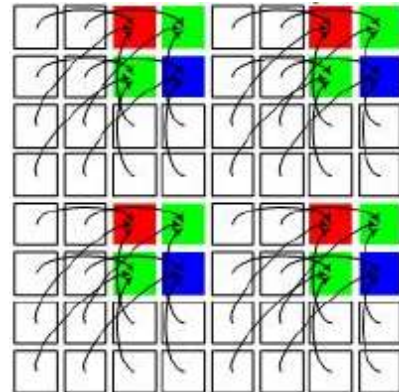
Kamera kitinin en yüksek çalışma frekansı 96 MHz olduğu halde DE2_115 kiti ile çalışmasını sağlayan sistem 50 MHz kullanmaktadır. Bu bazı FPS değerlerinin de buna bağlı olarak değişmesine yol açmaktadır. Örneğin kamera 800x600 VGA formatında satır ve sütun bindirme özelliği ile kullanılmaktadır. Bu konfigürasyonda varsayılan 96 MHz saat ile 85.2 FPS değerine ulaşabilecek iken 50 MHz saat sinyali ile en yüksek 44 FPS değerine ulaşabilmektedir. Ayrıca pozlama süresinin getirdiği bazı kısıtlamalar sebebiyle deneylerde en çok 32 FPS değerleri gözlemlenebilmiştir.

Kameradan gelen görüntü sinyalleri RAW (ham) formatta iken her satır için sırayla pixel renk bilgilerini barındırmaktadır. Ancak bayer filtresi sebebiyle her satırda ya sadece mavi ve yeşil ya da sadece yeşil ve kırmızı bulunmaktadır. RGB bilgisini düzgün olarak elde etmek için en az ardışık iki satır bilgisinin sütun değerleri karşılıklı olacak şekilde bulunulması gerekmektedir. Birden çok RGB kod çözme algoritması olmakla birlikte sistemde iki çeşit çözücü bulunmaktadır.

Kamera konfigürasyonu ile 1600x1200 çerçeve Şekil 4.14 ile etkisi görselleştirilmiş bindirme özelliği ile 800x600 ebadına dönüşmektedir [30]. Kamera hem satır hem sütun bindirmesini desteklemektedir, bu özellik sayesinde FOV değiştirilmeden çözünürlük düşürülebilmektedir. Bindirme işlemi deklere edilmiş bindirme miktarı kadar satır ve sütun pilxellerinin değerlerinin *ortalamasını* tek değer olarak çıkışa aktarmaktadır.



(a) Bindirmesiz pixel dizimi



(b) Yatay ve dikey bindirme ile

Şekil 4.14: Yatay ve dikey bindirme ile pixel okuma

Her gelen satır bilgisi satır ebadı kadar genişliği olan FIFO (First In First Out) bellekte tamponlanarak saklanmaktadır. Yeni satırın pixelleri geldikçe FIFO tampondan çıkan komşu pixel renkleri ile değerlendirilerek RGB renk bilgisi oluşturulmaktadır.

4.3. Gerçekleştirilen Sistem ve Önemli Bazı Yapı Blokları

İlk önemli yapı, görüntü işleme penceresini veri ile besleyecek olan *Line-Buffer* ya da satır tamponu da diyebileceğimiz yapıdır.

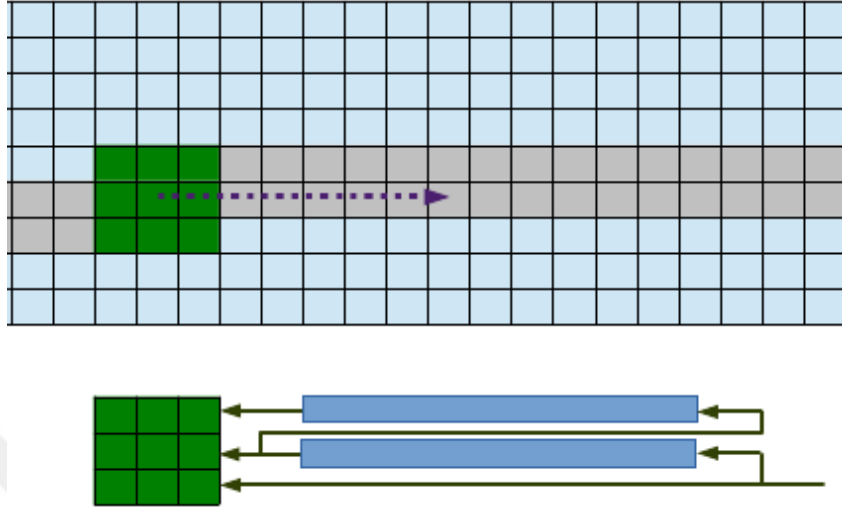
4.3.1. Satır tamponlama

Bilgisayar sistemlerinde en önemli unsurlardan birisi hafıza ve onun kullanımıdır. Bir FPGA çipinin içinde birden fazla çeşit ve sayıda hafıza bulunmaktadır. Bunlardan en önemlisi register olarak adlandırılan yazmaçlardır. Bunlar temel birimlerdir ve program yazımı sırasında normal tanımlama diliyle rahatlıkla kullanılabilirler. Ancak bu hafızalar gerçekte lojik birimin içinde bulunan hafızadır ve program yazıldıkça zaten kullanılmaktalar dolayısıyla kod yazdıkça tükenmektedirler. Özel bir hafıza türü ise içerideki RAM tabanlı hafızadır. Bunları kullanmak için editörde sihirbaz benzeri yardımcılar kullanmak gerekmektedir. Bu yardımcı programlayıcı araçlar sayesinde FPGA içinde dahili olarak bulunan RAM hafıza birimleri istenen özelliklerde yapılandırılarak modül olarak kullanıma sunulmaktadır.

DE2-115 kartında bulunan EP4CE115F29C7N yongasında 114.480 LE mevcut olduğundan aynı sayıda da register bulunur, ancak içindeki dahili ram hafıza boyutu bundan daha yüksek olup 3.981.312 adettir, kısaca yaklaşık 3.8Mb RAM hafıza bulundurmaktadır. Bunlar FPGA içerisinde M9K blokları olarak bulunmaktadır. Her M9K bloğunda 8.192 adet (eşlik bitleri dahil 9.216 adet) hücrecik bulunmaktadır. Bu bloklar tekli ya da ikili port modunda ayarlanabilmektedirler. Ayrıca her blok 8.192 hafıza birimini farklı modlarda çalıştırabilmektedirler (örnek; 8.192x1, 4096x2, 256x32 gibi). Blokların yerleşimini görmek için Şekil A.1 de açık yeşil çizgili alanlar incelenebilmektedir. Koyu yeşil olan bölgeler yapılandırılarak kullanılan RAM hafıza bloklarını işaret etmektedir. Bloklardan bir tanesinin yapılandırılmış hali Şekil A.2 de görülebilmektedir.

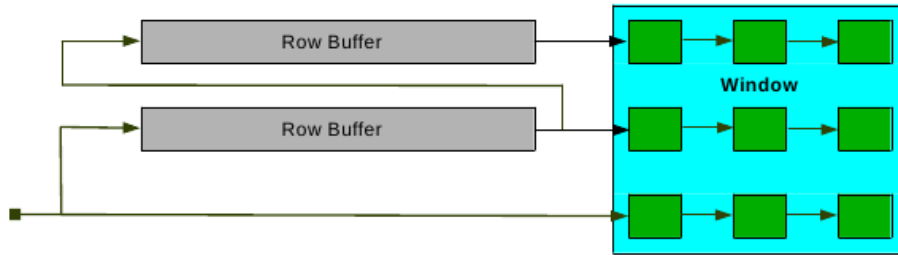
Kameramızın çözünürlüğü 800x600 boyutuna ayarladığımızdan ve 12 bitlik ADC bulundurduğundan tüm çerçeveyi barındırmak için yaklaşık 5.5 Mbitlik hafıza ihtiyacı duyulmaktadır. Ancak bu boyutta hafıza hem mevcut değil, hemde RGB dönüşümü sonunda bu sayı 3 katına çıkıp 16 Mbiti aşacağından genelde FPGA çiplerinde her defasında bir ya da

birkaç satır tamponlanabilmektedir. Sadece üzerinde işlem yapacağımız temel pencereyi sağlayacak kadarlık bir bandı tamponlayarak problem giderilmeye çalışılmaktadır. Genelde de bu tamponlama için dahili RAM hafıza birimleri kullanılmaktadır.



Şekil 4.15: Görüntü işleme pencereciği ve satır tamponlaması

Şekil 4.15 ile gösterilen sistem ile görsel olarak sistemin anlaşılması daha kolay olsa da genelde lojik diyagramlar çizilirken sinyallerin soldan geldiği düşünüldüğünden Şekil 4.16 ile gösterim devre tasarımında alışkanlıklarımıza daha çok hitap etmektedir.



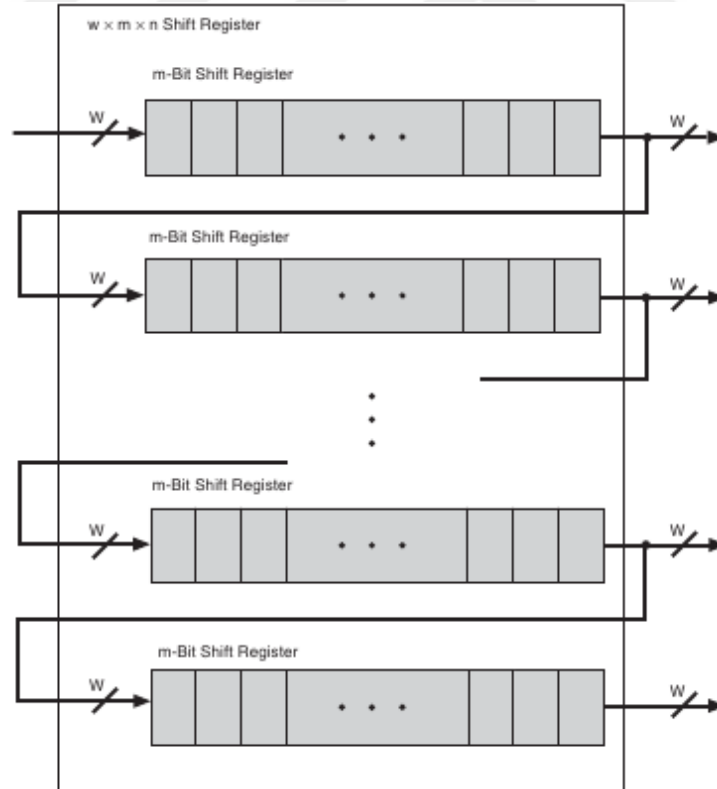
Şekil 4.16: Bir pencere işlemi için uygulanan paralel satır tamponlama

Şekil 4.16 ile görülen sistemde görüntü işleme sistemlerinin temel unsuru olarak çokça kullanılan 3 x 3 boyutunda ki pencere ve onu veri ile besleyen sol kısımda ki hat tamponları görülebilmektedir. Hat tamponları burada temel olarak kaydırmalı kaydediciler gibi kullanılmaktadır. Bilgi alt kısımdan geldikçe hem pencerenin alt sol kısmından işlenmek üzere pencereye giriş yapıyor hemde ilk hat tamponuna sol taraftan dolmaktadır. Tampon boyutu bir satır genişliği kadar seçilerek tampondan çıkıp pencereye akan pixel ile pencereye alttan yeni giriş yapan pixel gerçekte altlı üstlü olarak komşu olabilmektedirler.

Satır ya da hat tamponlama olarak adlandırdığımız sistemde pencere işlemi temelde merkezde bulunan pixel için yapıldığından gerçekte bir hat süresi boyunca sinyal geciktirilmişdir.

Hat tamponlama için bir ya da birkaç kaydırmalı kaydedici birden kullanılır, bu sebeple üreticiler konfigürasyon sihirbazlarını buna uygun olarak ayarlamışlardır. Örnek olarak Şekil 4.17 ile görülen IP (Intellectual Property - Fikri Mülkiyet) bloğu uygun şekilde oluşturulduğunda doğrudan paralel bir tampon olarak kullanılabilir [25]. Örnek olarak sistemimizde $w=8$, $m=799$ ve $n=2$ olarak bir uygulanmıştır.

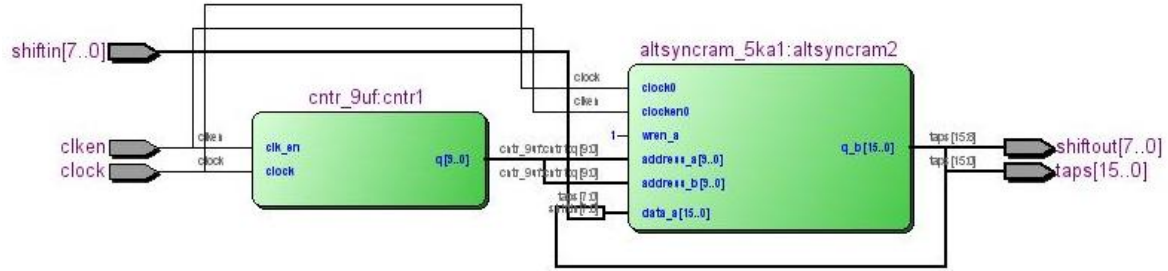
Altera şirketi Quartus II ile sunduğu IP bloklarıyla bu hizmeti vermekte ve blok yapısı oluşturulurken bir sihirbaz gerekli ayarlamalar için değer girmemizi istemektedir. Yapı oluşturulduğunda gerçekte Altera şirketine ait bir donanım tanımlama dili olan AHDL ile blok oluşturulmakta ancak kendi sistemimizde modül olarak kullanmak üzere Verilog ya da VHDL uyumlu prototip kısımları oluşturulmaktadır. Bu başlık kısımları alınarak farklı modüller içinde alt modül olarak kullanabilmektedir.



Şekil 4.17: Altera alt_shift_taps bloğu

Otomatik olarak oluşturulan modüllerden birisinin RTL blok şeması Şekil 4.18 de görülebilmektedir. Burada yapı iki ayrı kısımdan oluşmaktadır. Sağda ki modül aslında RAM

bloğunun kendisi, bir tarafta sadece giriş diğer tarafta sadece çıkış bulunmaktadır. Gerçekte giriş ve çıkış $w \times n$ boyutunda olup girişin ilk w genişliği asıl giriş olup kalan giriş terminaleri çıkışa kaydırmalı olarak bağlanmıştır. Diğer modül ise yanındaki RAM modülüne adres sağlamakla görevli bir sayıcı olarak çalışmaktadır.



Şekil 4.18: Altera alt_shift_taps bloğu RTL şeması

Sistemin sol tarafında kaydırmalı giriş ile birlikte saat girişi ve izin sinyali bulunmaktadır. Sağ tarafta ise çıkış olarak hem ana çıkış hemde paralel musluklar görülebilmektedir. Ayrıca sağ tarafta bulunan modülün iç yapısı için Şekil B.8 de gösterilen RTL blok yapısı incelenebilmektedir.

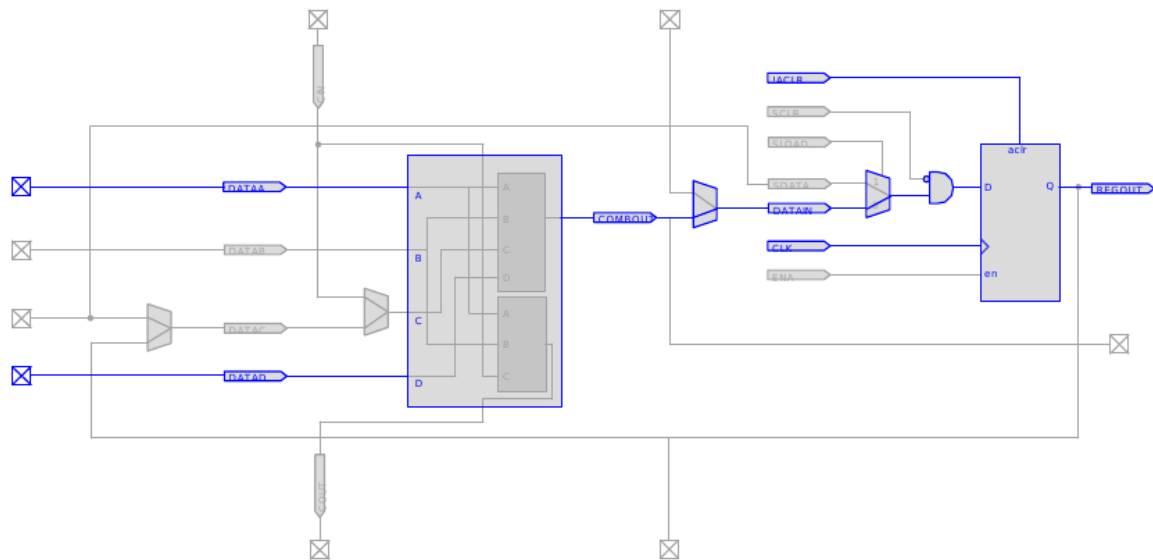
4.3.2. FPGA sisteminde gecikmeler ve bunların kontrolü

FPGA yongalarında zamanlama en can alıcı ve karmaşık konulardan birisi olarak karşımıza çıkmaktadır. Temelde kullanılan bloğun çeşidi, yeri, besleme voltajı v.b. gibi çok sayıda parametre gecikmeleri değiştirmektedir. Altera Quartus II yazılımı zamanlama ve gecikmelerle ilgili analizler yapabilir ancak bunun için kısıtların girilmiş olması gerekmektedir.

Temel bilgi olarak bir FPGA yongasının kodunda bulunan “speed grade” numarası en önemli parametredir. Hız kademesi olarak anlayabileceğimiz bu değer temelde cihaz içinde bulunan bir macrocell biriminin nano saniye cinsinden gecikme süresidir [31]. FPGA yongasında ise ana birim lojik birimdir (LE). Örnek olarak eğitim setimizdeki FPGA yongasının numarası EP4CE115F29C7N olup soldan ikinci parametre olan 7 rakamı cihazın hız kademesini, dolayısıyla gecikme süresini göstermektedir. Bu değer küçük olduğu zaman FPGA daha hızlı çalışabilmektedir. Bizim örneğimizde görülen hız kademesi değeri 7 ns olduğuna göre $f = 1/7ns \approx 142MHz$ yaklaşık değeri FPGA için çıkılabilecek en yüksek saat hızı olmaktadır.

FPGA için temel birim lojik birim olduğundan hem mantıksal hem de matematiksel

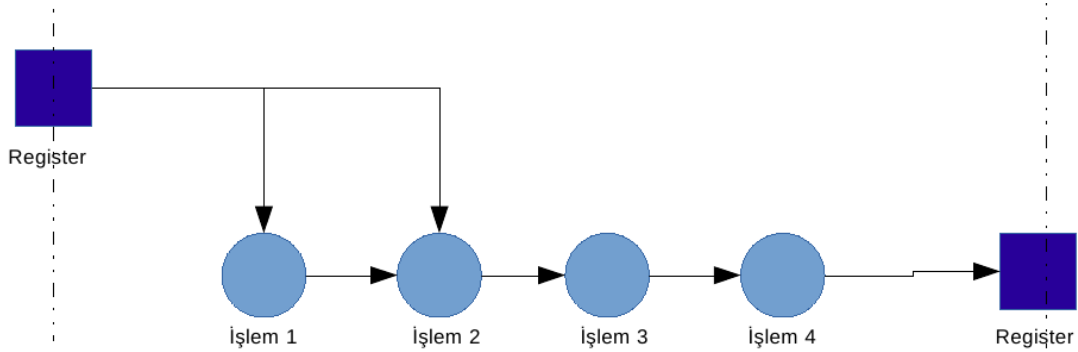
Şekil 4.19: Matematiksel işlem için konfigüre edilmiş bir LE



Şekil 4.20: Mantıksal işlem için konfigüre edilmiş bir LE

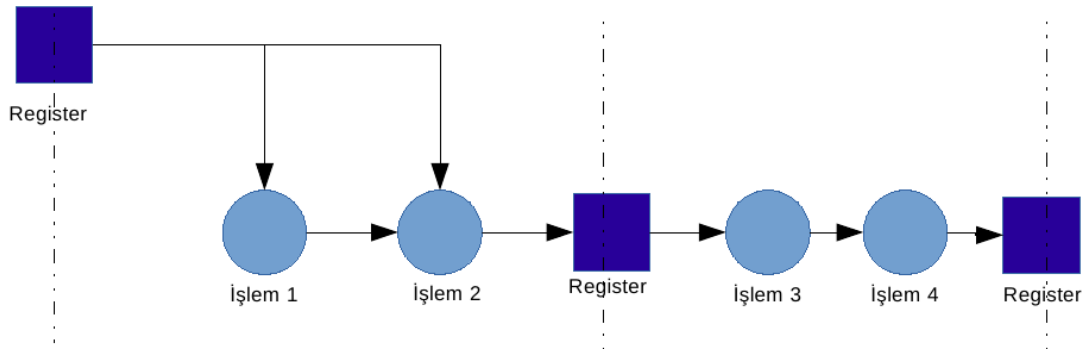
Gecikmelerden dolayı FPGA için kod yazarken istediğimiz kadar özgür olamayız ve özellikle yüksek gecikmelere sahip yongalarda uzun sürecek işlemler için peşi sıra register kullanarak işlemin sağlıklı yürütüldüğünden emin olmak durumunda kalınmaktadır. Örneğin

gecikme yeterince düşük ve saat hızımız da uygun ise Şekil 4.21 de gösterildiği gibi birçok işlem iki kaydedici arasına sıkıştırılabilmektedir.



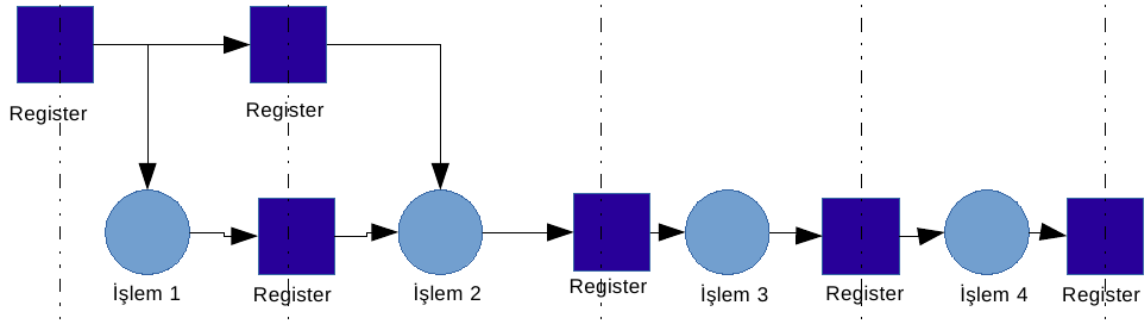
Şekil 4.21: Düşük gecikmeli bir sistemde işlem örneği

Her bir işlem için kullanılan lojik birim sayısı farklı olmakla birlikte düşük gecikmeli sistemler düşük saat hızlarında kodlamada bize büyük esneklik sunmaktadırlar. Ancak gecikme ya da hız arttıkça kodlamada daha dikkatli olmak ve gerekirse sinyali yazmaçlar aracılığı ile kararlı bir şekilde ikinci aşamaya ulaştırmak gerekebilmektedir. Örnek olarak Şekil 4.21 ile aynı işlemi uygun bir konuma kaydedici yerleştirerek Şekil 4.22 de olduğu gibi gecikme değeri yüksek bir sistemde de gerçekleştirilebilmektedir. Yazmaçların sağladığı mandallama olmadan sistemin kararlılığı teyit edilememektedir.



Şekil 4.22: Ortalama değerde gecikmeye sahip bir sistemde işlem örneği

Ancak saat hızımız ya da gecikme katsayımız çok yüksek ise neredeyse her işlemten sonra mandallama yapmamız gerekmektedir. Hatta Şekil 4.23 te gösterildiği gibi sadece işlemlerden çıkan sonuçları değil, bir işlem için gerekli girişi bile eğer gecikecek diğer bir girişle birlikte işlenecekse geciktirerek işleme sokmak durumunda kalınmaktadır.



Şekil 4.23: Yüksek gecikme değerine sahip bir sistemde işlem örneği

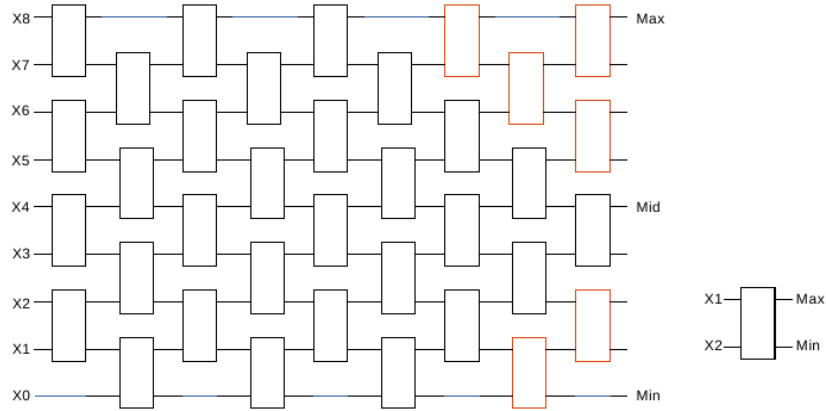
Bazı durumlarda geciktirerek işlem yapmak bile sistemi çalıştırmaya yetmeyebilmektedir. Örneğin bir sinyalin ardışık değerlerinin beraberce işlenmesi ve işlenmiş ardışık değerlerinin de fonksiyonda kullanılması gerekiyorsa sinyalin kaynağını yavaşlatmak ya da işlemleri gerçek zamanlı işleyebilmek için daha hızlı bir donanım değiştirmek gerekebilmektedir.

4.3.3. Medyan filtre

Median (Medyan) filtre görüntü işlemede sıklıkla kullanılan non-lineer filtrelerden birisidir. Temelde amaç görüntüde hatalı şekilde oluşan yüksek parlaklık değerlerinin düzeltilmesidir. Bunun için işlenecek piksel pencere merkezinde olacak şekilde görüntü üzerinde gezdirilen belli boyuttaki bir pencere içinde kalan tüm piksel değerlerinin medyan değeri bulunup merkezde ki hücreye yeni değer olarak yazılmaktadır.

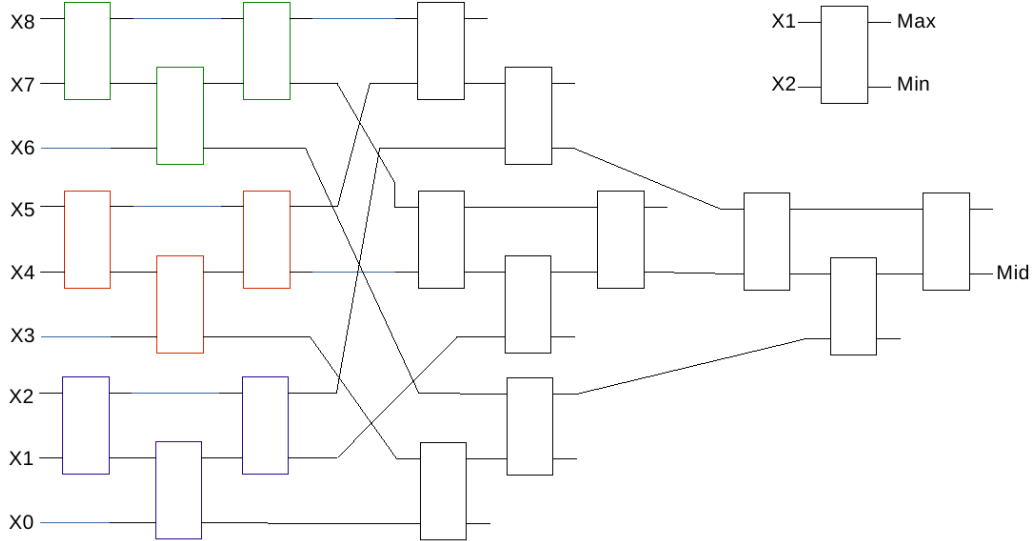
Medyan filtrenin optimizasyonu

Medyan işlemi aslında ortanca değeri bulmak olduğundan yoğunlukla karşılaştırma ve mux işlemi gerektirmektedir. Şekil 4.24 'te gösterilen düzenlemede normal medyan işlemi için gereken karşılaştırma ve seçme işlemi sayısı görülmektedir. İşlem penceresinin 3 x 3 olması 9 adet değerin içinden medyan değerinin bulunmasını gerektirmektedir. Burada 36 elemanın tümü kullanıldığında sıralama yapıyor. Ancak buna gerek olmadığından sadece medyan değerini bulmak için siyah olarak işaretli birimler yeterlidir. Bu da toplam 30 adet karşılaştırma gerekmektedir. Burada görülen her bir işlem bir adet karşılaştırma iki adet MUX ünitesi barındırmaktadır.



Şekil 4.24: 9 değer içinden medyan değerini bulmak için gereken işlem yığını

Medyan işleminin hızlı yapılabilmesi için bazı optimizasyonlar gerekmektedir. Şekil 4.25 te gösterilen bir yöntem ile önce 3 lü gruplara ayrılan değerler ilk karşılaştırma sisteminin sonuna en büyük 3 değer, en küçük 3 değer ve orta 3 değer olarak ayrıştırılmaktadır. Buraya kadar 3 x 3 adet karşılaştırma yapılmaktadır. Daha sonra ortancaların medyan değeri, büyüklerin en küçüğü ve küçüklerin en büyüğü bulunuyor. Son olarak bunların medyan değeri kullanılıyor. Bu sistemde toplam 19 adet işlem gerekli olmaktadır.



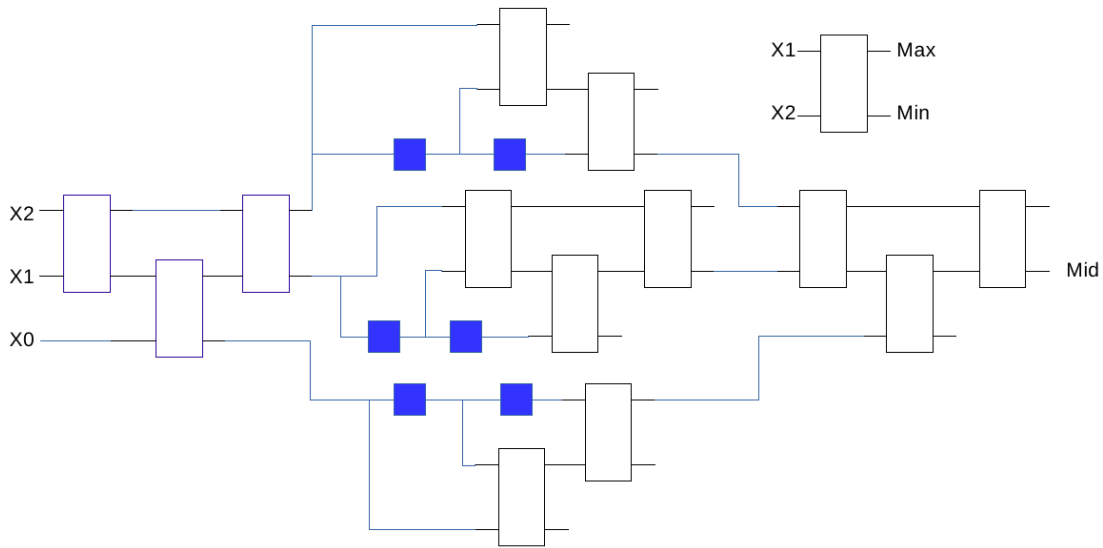
Şekil 4.25: Optimize edilmiş bir medyan bulma sistemi

Son optimizasyon işleminde, hafıza kullanılarak bazı karşılaştırma işlemlerine olan gereksinim kaldırılmıştır. Şekil 4.25 te gösterilen sistemde kırmızı mavi ve yeşil 3'lü karşılaştırma gruplarından ikisinin işlevi kaydediciler vasıtasıyla yapılmaktadır.

Şekil 4.16 tekrar incelenirse işlem için kullanılan 3x3 pencere aslında bir yönde ilerlemekte dolayısıyla pencerenin sol tarafında kalan hücrecikler gerçekte her defasında bir sağda

ki hücrenin değerini almaktadır. Bu özellikten faydalanılarak dikey ilk 3 hücrenin üzerinde işlem yapılarak büyük, küçük ve orta değerler bulunmaktadır. Dikkat edilirse her sinyal uçlarından her defasında ilgili 3'lü sütunun büyük, küçük ve ortanca değerleri verilmektedir. Bu değerleri kaydırmalı kaydedici vasıtasıyla sisteme verdiğimizde toplamda 6 adet işlem-den daha tasarruf edilmiştir.

Şekil 4.26 da görüleceği üzere 6 adet kaydedici(mavi kutucuklar) kullanılarak sistem 6 adet karşılaştırma işleminden kurtarılmıştır. Gerçekte her LE içinde en az bir tane kaydedici olduğu halde şekildeki iki giriş ve iki çıkışlı birimlerin her biri için birden fazla LE kullanılması gereklidir.



Şekil 4.26: Donanım için optimize edilmiş bir medyan bulma sistemi

4.4. Sistemin Modüllerinin Birleştirilmesi ve Çalıştırılması

Sistem kameradan aldığı görüntü bilgisini yakalayıp RGB verisine dönüştürdükten sonra dinamik belleğe göndermekte, burda eklenmiş modül RAW2RGB ile Sdram_Controller modülleri arasına yerleştirilmiştir. Şekil B.3 te gösterilen ve devreye *temp* adıyla eklenen modüle ayrıca verileri displayden okumak için 7-segment göstergeler, yeşil ve kırmızı ledler ile komut verebilmek için butonlar ile seçme anahtarları bağlanmıştır.

Gelen renk bilgileri önce istenirse sabit değerle çarpma vb gibi işlemleri mümkün kılmak için *normalizer* modülüne aktarılmakta, bu modülde ayrıca VGA kontrolcüsünce kullanılmayacak alt bitleri işlemlerde kaynak sarfı fazla olmasın diye ihmal edilerek sadece kullanılacak bitler çıkışa aktarmaktadır. Gri seviye bilgisi de burada elde edilmektedir. Gri bilgisi kırmızı, mavi ve iki kat olacak şekilde yeşil bilgisi toplandıktan sonra alt iki bit ihmal

edilerek dörde bölünmüş olur ve bu haliyle kullanılmıştır.

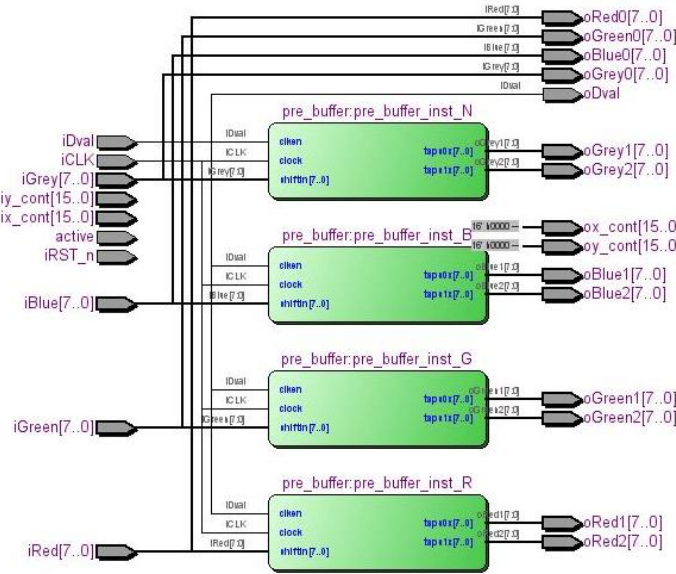
Bundan sonra sinyaller 3x3 boyutunda pencere ve maskelerle işlenmek üzere *mask* modülüne aktarılmaktadır. Bu modülden çıkan işlenmiş sinyallere ekranda yardımcı çizgiler çizmek ve işlenmiş sinyalleri incelemek amacıyla *fcross* adında bir başka modüle aktarılmakta burdan sonra ise RAM kontrol modülüne aktarılmak üzere çıkışa sürülmektedir.

Sistemde temp modülü içinde u3 olarak bulunan ve iç yapısı Şekil B.5 te gösterilmiş olan *display* modülü temp içindeki farklı noktalar ve sinyallerden veri toplayıp barındırdığı sayıcılar vasıtasıyla belli bazı bilgileri buton ve seçme anahtarlarından gelen komutlara bağlı olarak çıkışa yani 7-segment displayler ve ledlere aktarılmaktadır. Bu seçimleri oluşturabilmek adına 2 ve 3 numaralı butonlar sırasıyla yukarı ve aşağı tuşları olarak bağlanmıştır. Seçme anahtarlarından SW1 iç sayıcılar için yön seçiminde, SW2 medyan filtresinin devreye sokulup çıkarılmasını kontrol etmekte, SW3 ve SW4 anahtarları ise 1. ve 2. genel seçme anahtarları olarak ilişkilendirilmiştir.

Şekil B.4 te gösterilen Fcross modülü aşağı ve yukarı butonları ile yön seçme anahtarına bağlı olarak çalışan sayıcıların kontrol ettiği akümülatörlerle oluşturduğu eşik değerleri segmentasyon işlemi için kullanılmaktadır. İki farklı eşik değer oluşturulmaktadır. Bunlardan biri her kırmızı, mavi ve yeşil bilgisi için eşik olmakta, diğeri ise gri bilgisi için eşik olarak kullanılmaktadır.

Eşik değerler başlangıçta sıfır olduklarından sistem açıldıktan sonra seçme anahtarı ve butonlar yardımıyla ayarlanması gerekmektedir. Görüntü bilgisinde değerlerin eşik değerleri aştığı noktalar işaretlenerek kaba bir şekilde nesne seçimi yapılmaktadır. Tek nesne için x ve y yönlerinde ağırlık merkezleri binary resim bilgisine göre oluşturulmaktadır. Oluşturulan bu $\bar{x}$ ve $\bar{y}$ noktaları ve çerçeve sonunda belli yazmaçlara atanarak gösterim için display modülüne aktarılmaktadır.

Şekil B.7 de gösterilen Mask modülü görüntü işleme ve tamponlama modüllerine ev sahipliği yapmak üzere yazılmış bir ana modüldür. İki temel modülden ilki görüntü işleme penceresine veri sağlamak amacıyla çalıştırılan satır tamponlarını barındıran *mask_pre_buffer* modülüdür. Bu modül içerisinde gri dahil olmak üzere her renk için ayrı ayrı tamponlar barındırmaktadır.



Şekil 4.27: Mask ön tamponu u0 RTL blok şeması

Tamponlar 3x3 lük pencere beslemek üzere tasarlanmıştır. Toplamda 4 adet tamponun her birinde 8 bitlik 2x799 'luk hat tamponları bulunmaktadır. Pencere işlemlerini yapmak üzere görüntü işleme modüllerini barındıracak olan *mask_process* isimindeki modül ise ikinci ana modülümüz olmaktadır.

Şekil B.9 da gösterilen Mask_process modülü içerisinde her bir renk için bir tane olacak şekilde *process_colour* modülü barındırmaktadır. Process_colour modüllerinin her biri Şekil B.10 ile gösterildiği gibi medyan filtre modülünü ve bu modülün devreye alınıp çıkarılmasından oluşacak görüntü kaymalarını engelleyecek geciktirme sistemi ile filtrenin uygulanacağı iç çerçevenin başlangıç ve bitiş değerlerini kontrol eden yapıyı barındırmaktadır. Medyan filtresinin 3x3 lük penceresinin tam olarak aktif görüntü penceresinin içinde bulunduğunu gördükten sonra sistem filtreyi devreye almaktadır.

Medyan modülü daha önceki başlıkta anlatılmış en son yapı olan donanım için optimize edilmiş medyan filtresini barındırmaktadır. Bu filtre modülünün bilgileri tezin ekler bölümüne bırakılmıştır. Ek C kısmında sayfa 68 den başlayan verilog kodları medyan modülüne ait olup sentezlenen devrenin RTL blok şeması Şekil B.11 de incelenebilmektedir.

Gerçekleştirilen sistemin genel yapısına ait RTL blok diyagramı için Şekil B.2 incelenebilmektedir. Gerçekleştirilen yapının FPGA yüzey yerleşimi için Şekil B.1 incelenebilir. Sistemin nesne algılaması sırasında alınan örnek fotoğraflar için Şekil B.12 ve B.13 incelenebilir. Ayrıca tek nesne için sistemin hesapladığı nesne merkezi koordinatlarının göstergede 16'lık sayı sistemiyle okunması Şekil B.14 'de görülebilmektedir.

5. SONUÇLAR ve ÖNERİLER

5.1. Sonuçlar

Denemelerde tek çerçeve geçişi ile yapılabilen işlerin daha kolay yapılabildiği görülmüştür. Sistemde oluşan asıl gecikmeler temelde kullanılan tamponlamalar sebebiyle oluşmaktadır. Bu tamponlamalardan RAW2RGB modülünde uygulanmış kısmının oluşturduğu aslında bir gecikme değil önceki verinin hafızada tutulması olarak görülebilir, ancak görüntü işleme birimlerinde kullandığımız 3x3 boyutunda ki temel birimi veri ile beslemek için kullandığımız hat tamponları bir satır süresi boyunca gecikmeye sebep olmaktadır.

Ayrıca doğası gereği çerçeve birimini beklemek zorunda olan işlemler yukarıda bahsedilen gecikmeye hem maruz kalmakta hemde kendi gecikmelerini bu sürece dahil etmektedirler, ancak bu ihmal edilebilmektedir. Temelde FPGA sistemlerinde oluşan gecikmeler, görüntüyü alan birimin satır zamanlamasının katları kadar olmaktadır. Denemelerimizde neredeyse canlı görüntü üzerinde işlemler yapılabilmektedir.

Birden fazla filtrenin peşi sıra kullanımı için birden fazla sayıda hat tamponu kullanılması durumunda, kullanılan tüm tamponların gecikmelerinin birbiri üzerine ekleneceği unutulmamalıdır.

Sahnenin aydınlık seviyesinin önemi büyük olmakla birlikte çok şiddetli bir ışık kullanmamıza gerek kalmamıştır. Kameranin dahili analog ve dijital kazanç devrelerinin başarısı bizlerin *normalizer* modülünde renklerin aydınlık değerleri üzerinde ayrıca iyileştirme işlemleri yapmamızı gereksiz kılmıştır. Yapmayı denediğimiz bazı ışık seviyesi işlemlerinde renk verisi doyuma girmiş ve görüntü bozulmalar meydana gelmiştir.

Deneyler sırasında nispeten düşük aydınlık seviyelerinde çalıştığımızdan eşik seviye değerlerimizde bu sebeple küçük olmuştur. Sekiz bitlik genişlik ile maksimum 255 değeri için düşündüğümüzde eşik değerimiz bazı aydınlık seviyeleri için 50 değerinin civarında olmuştur.

FPS hızını arttırmak için pozlama süresini kısalttığımızda sensör daha düşük değerler üretmekte bu sebeple basit bir LED lamba kullanmak durumunda kalınmaktadır. Çok yüksek aydınlık seviyeleri için parlama sorunu özellikle parlak renkli cisimlerin görüntüleri ile çalışılırken sorun olmuştur.

İlk çalışmalar sırasında denenen bazı renk filtrelerinin doğru çalıştığını görebilmek için

sahneyi LCD ekranda oluşturup kameradan yakalamaya çalışılmıştır. Ancak LCD monitör görüntüleri ile gerçek sahne görüntüleri arasında parlaklık ve renk algısında bazı farklılıklar olduğu görülmüştür. LCD ile yakalanan başarının tekrarlanabilmesi için aydınlık seviyesini yükseltmek gerekmiştir. Parlak cisimlerdeki yansımalar daha ciddi problemler haline gelmiştir.

Aydınlatmanın uygulanma açısı sebebiyle oluşan gölgeler maalesef sistem tarafından cismin kısımları olarak algılanmıştır. Bu sebeple aydınlatma için kullanılan lamba, kameranın çok yakınına taşınmak durumunda kalmıştır.

Tek bir nesne için kullanılan ve nesnenin görüntüdeki şeklinin ağırlık merkezini bulan uygulama özellikle nesne, görüntü alanının sınırlarına temas ettikçe hatalı değerler üretse de genel olarak hızlı ve başarılı sonuçlar vermiştir.

Medyan filtresi görüntüde belli ölçüde detay kaybına ve flu görüntüye sebep olmuştur. Ancak obje merkezi koordinatları gibi bazı değerlerin nispeten daha sabit değerler olmasını da sağlamıştır. Donanımsal bazı kısıtlamalar sebebiyle görüntü bilgisinin alt 4 bitinin ihmal edilmesi de benzeri bir kararlılık etkisi göstermiştir. Hızlı değişen LSB bitlerinin işleme alınmamıştır. Ancak belirtmek gerekir ki kullanılan kamerada piksellerde bindirme işleminin uygulanmış olması belkide medyan filtresi ile çözülebilecek muhtemel bozulmaların daha baştan büyük oranda giderilmiş olması muhtemeldir.

Sentezlediğimiz donanım kısımları, FPGA yongamız geniş olduğundan kaynak harcamamız büyük oranlara çıkmamıştır. İlk sistem FPGA kaynaklarının %3'ünü kullanırken değiştirilen hali ile kaynak kullanımı %5~6 civarına çıkmıştır. Bu bile eklenebilecek yapı miktarıyla ilgili bize fikir vermektedir. Örneğin ikinci bir kamera kiti ile stereo görme sistemleri çalışması mümkündür, ancak bunun için mutlaka hafıza dar boğazının aşılması gerekmektedir.

Yazılımsal sistemlerin geniş esnekliğinden yoksun olmasına rağmen erken görme işlevlerinde CPU (Central Processing Unit) üzerindeki iş yükünü hafifletmekte oldukça başarılı bir sistem olarak FPGA sistemleri gelişmeye gayet açık sistemlerdir. Kullanımda karşılaşılan problemlerin büyük kısmı aslında kodlamasının bir *veri akış dili* olmasından ve normal yazılan programlardan farklı yapıda olmasından kaynaklansa da halen problemin büyük kısımlarından biri hafızanın kullanımındadır. Sistemimizde ki dinamik hafıza farklı frekanslarda çalışan sistemlerin birbirine bağlanmasında tampon olarak kullanılmıştır. Bu onun farklı işlemler için kullanılmasını engellemiştir.

Kullanılan FPGA düşük özelliklere sahip bir seriye ait olduğundan özellikle gecikmelerin ayarlanması kısmında sorunlarla karşılaşmıştır. Register kullanarak mandallama yapmak belli ölçüde çözüm olsa da FPGA yongasının lojik birim yayılma gecikmesini bu seri için uzun olması problem oluşturmuştur.

Her ne kadar FPGA iç yapısında yazılımsal işlemci oluşturulabiliyorsa da bu işlemcilerin doğal çalışma frekansı FPGA çalışma frekansı ile aynı olacağından sağlayacağı imkânlarda kısıtlı olacaktır. Bu sebeple FPGA yongalarının normal görüntü işleme sistemlerinde ek donanımlar olarak bulundurulması daha avantajlı olacaktır.

5.2. Öneriler

Kullanılan kart bir geliştirme kartı olduğundan donanımsal bağlantıların genele yönelik yapılmış olması ve kullanılan FPGA tipinin özellikleri bazı açılardan kısıtlamalar getirmiştir. FPGA kodlanırken iç yapısında devre oluşturulduğu gibi kartı tasarlarırken de müdahale şansı olması durumunda seçenekler artacaktır. Örneğin mevcut yongaya tek bir hat üzerinden dinamik bellek modülü bağlanmıştır, halbuki bu seri yongalar kuzey, güney, doğu ve batı yönlerinden bellek bağlanabilecek yapıda üretilmişlerdir. Bu kullanmak için birden fazla hafıza biriminin bağlanabilmesini sağlamaktadır. Ancak sistemde sadece bir tane dinamik hafıza hattı bulunmakta ve o da senkronizasyon problemini gidermek için kullanılmaktadır.

Yayılma gecikmesi konusunda Cyclone serisi FPGA yongaları yerine firmanın Stratix serisi yongalarının sağladığı 2 ns gibi gecikme değerleri bir saat darbesi süresinde yapılabilir işlemleri arttırılabilmekte, ya da saat frekansını doğrudan arttırılmasına izin vermektedir. Ancak bu yongaların programlarının derlenebilmesi için Quartus II geliştirme ortamının lisanslı olarak kullanılması gerektirmektedir. Halbuki Cyclone serisinde aynı programın Lite sürümü derleme yapmakta yeterli olmaktadır.

Dikkat edilmesi gereken noktalardan bir diğeri de maliyettir. Kullanılan setin maliyeti neredeyse üzerinde bulunan FPGA yongasının maliyeti ile benzer olmaktadır. Zaten setlerin fiyatlarını neredeyse FPGA yonga fiyatı belirlemekte ve genelde de pahalıya mal olmaktadır. Bu kendi kartımızı tasarlarırken maliyetin daha yüksek olacağını göstermektedir.

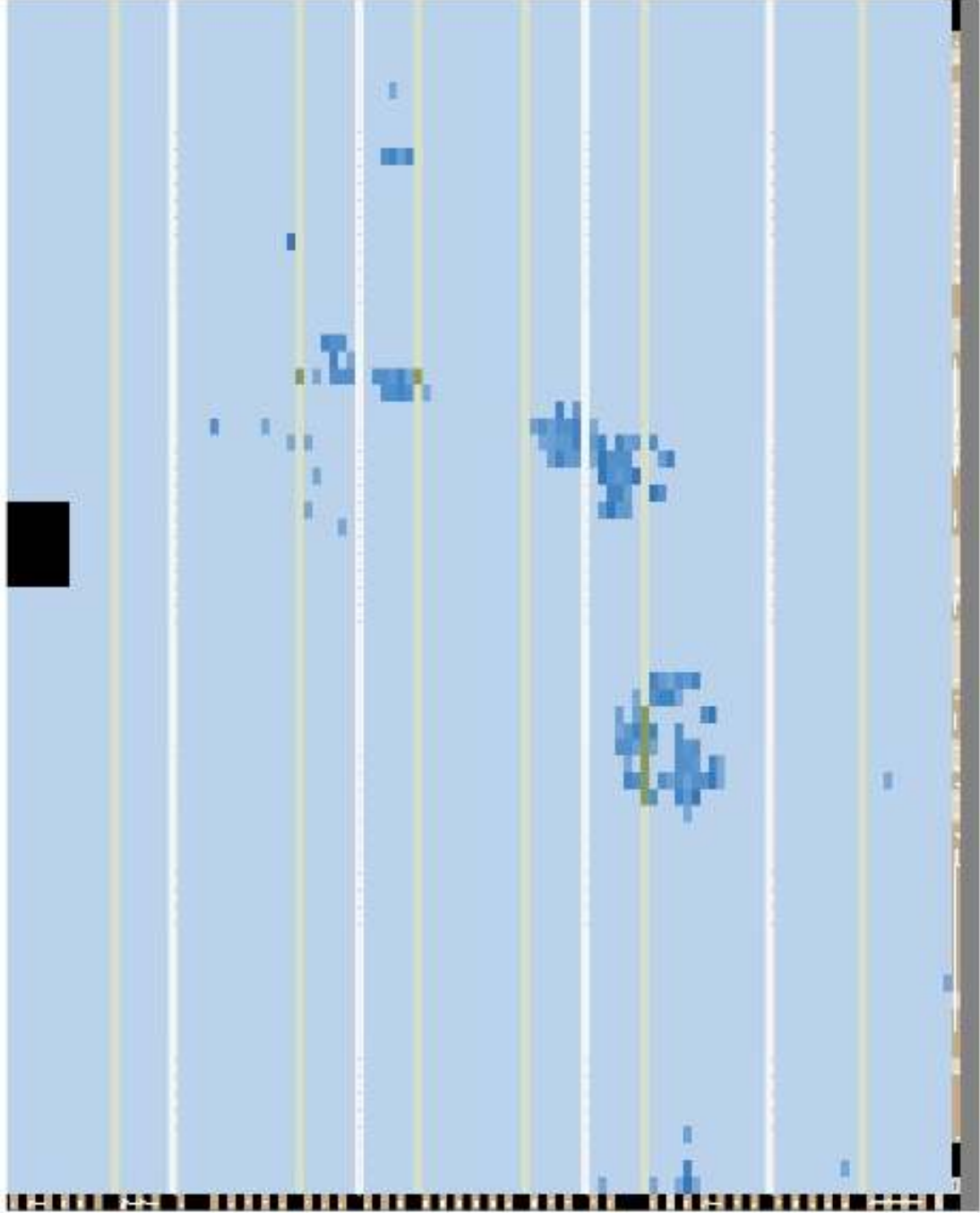
KAYNAKLAR

- [1] **Bailey, D.G.**, 2011 Design for Embedded Image Processing on FPGAs John Wiley & Sons (Asiz) Pte Ltd, 1 Fusionopolis Walk, #07-01 Solaris South Tower.
- [2] Brain Facts and Figures <https://faculty.washington.edu/chudler/facts.html>.
- [3] **de Silva, C.W.**, 2016 Sensors and Actuators: Engineering System Instrumentation 2 Auflage CRC Press LLC.
- [4] **Zhao, J., Huang, X. and Massoud, Y.**, 2014 An efficient real-time FPGA implementation for object detection in New Circuits and Systems Conference (NEWCAS), 2014 IEEE 12th International IEEE 313–316.
- [5] **Kyrkou, C., Ttofis, C. and Theocharides, T.**, 2011 FPGA-Accelerated Object Detection Using Edge Information in 2011 21st International Conference on Field Programmable Logic and Applications IEEE 167–170.
- [6] **Huang, C. and Vahid, F.**, 2011 Scalable object detection accelerators on FPGAs using custom design space exploration in SASP IEEE Computer Society 115–121.
- [7] **Salve, M., Rajput, M. and Shingate, S.**, 2013 Fpga Based Moving Object Detection Algorithm Implementation for Traffic Surveillance *Mayur Salve et al Int. Journal of Engineering Research and Applications* **3(5)** 1829–1832.
- [8] **McCurry, P., Curry, P.M., Morgan, F. and Kilmartin, L.**, 2001 Xilinx FPGA Implementation of a Pixel Processor for Object Detection Applications in In Proceeding of the Irish Signals and Systems Conference, UCD 404–411.
- [9] **Tang, J.W., Shaikh-Husin, N., Sheikh, U.U. and Marsono, M.N.**, 2016 FPGA-Based Real-Time Moving Target Detection System for Unmanned Aerial Vehicle Application *International Journal of Reconfigurable Computing* **2016** 167–170.
- [10] **Eric, A.**, 2015 FPGA Implementation of Median Filter using an Improved Algorithm for Image Processing *International Journal for Innovative Research in Science & Technology* **1(12)**.
- [11] **Fahmy, S.A., Cheung, P.Y.K. and Luk, W.**, 2005 Novel FPGA-based implementation of median and weighted median filters for image processing in International Conference on Field Programmable Logic and Applications, 2005. 142–147.
- [12] **Sanny, A. and Prasanna, V.K.**, 2013 Energy-Efficient Median Filter on FPGA in 2013 International Conference on Reconfigurable Computing and FPGAs 1–8.
- [13] **S.S.Tavse, P.M.Jadhav and M.R.Ingle**, 2012 Optimized Median Filter Implementation on FPGA Including Soft Processor *International Journal of Emerging Technology and Advanced Engineering* **2(8)** URL <http://www.ijetae.com/files/Volume2Issue8/IJETAE081238.pdf>.
- [14] https://en.wikipedia.org/wiki/Video_camera_tube video camera tube 21.06.2016.
- [15] **B., W. and S., G.**, 1974 Buried channel charge coupled devices US Patent 3,792,322.

- [16] **Tompsett, M.F.**, 1978 Charge transfer imaging devices US Patent 4,085,456.
- [17] Charge-coupled device https://en.wikipedia.org/wiki/Charge-coupled_device.
- [18] http://www.anafocus.com/products-and-services/standard-products/afls4k_1S4k_Line-Scan_Image_Sensor. 2012.
- [19] Active pixel sensor https://en.wikipedia.org/wiki/Active_pixel_sensor.
- [20] YCbCr <https://en.wikipedia.org/wiki/YCbCr> 21.06.2016.
- [21] Bayer filter https://en.wikipedia.org/wiki/Bayer_filter.
- [22] **Bayer, B.E.**, 1976 Color imaging array US Patent 3,971,065 A.
- [23] **Kr., I.**, 2012 schematic drawing example of programmable logic array URL [https://commons.wikimedia.org/wiki/File:Programmable_logic_array\(schematic_drawing_example\).svg](https://commons.wikimedia.org/wiki/File:Programmable_logic_array(schematic_drawing_example).svg).
- [24] FPGA Architecture URL https://www.altera.com/en_US/pdfs/literature/wp/wp-01003.pdf https://www.altera.com/en_US/pdfs/literature/wp/wp-01003.pdf.
- [25] Cyclone IV Device Handbook https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-iv/cyclone4-handbook.pdf 2010.
- [26] **He, C., Papakonstantinou, A. and Chen, D.**, 2009 A novel SoC architecture on FPGA for ultra fast face detection in Computer Design, 2009. ICCD 2009. IEEE International Conference on IEEE 412–418.
- [27] **Viola, P. and Jones, M.**, 2001 Robust Real-time Object Detection *International Journal of Computer Vision* **57(298)** 137–154.
- [28] Terasic DE2-115 User Manual URL http://www.terasic.com.tw/cgi-bin/page/archive_download.pl?Language=English&No=502&FID=cd9c7c1feaa2467c58c9aa4cc02131af 2013.
- [29] TRDB-D5M 5 Mega Pixel Digital Camera Development Kit User Manual URL <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=68&No=281&PartNo=1> 2014.
- [30] Terasic TRDB-D5M Hardware Spacification 2009.
- [31] Altera Device Part Number Format <https://www.altera.com/products/general/devices/dev-format.html#speed>.

A. Mevcut Sistem

A.1. Mevcut sistemin FPGA yüzey yerleşimi



Şekil A.1: Mevcut D5M sistemi FPGA yüzey yerleşimi

A.2. FPGA dahili M9K RAM bloklarından bir örneği

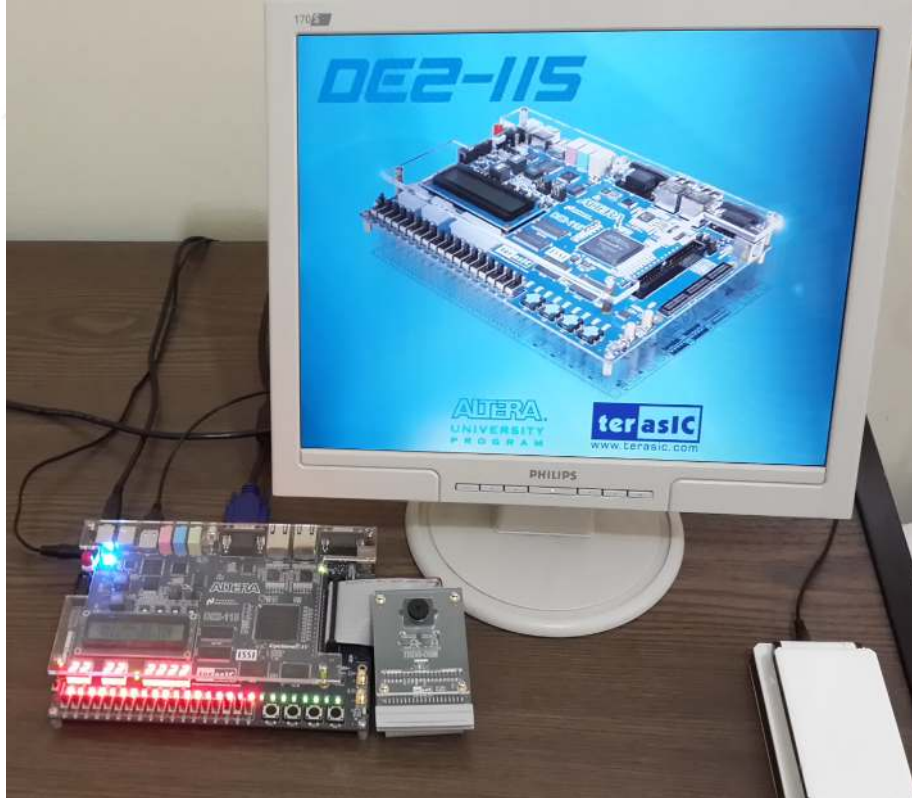


Şekil A.2: M9K hafıza bloğu konfigürasyon örneği

A.3. Mevcut görüntüleme sistemi



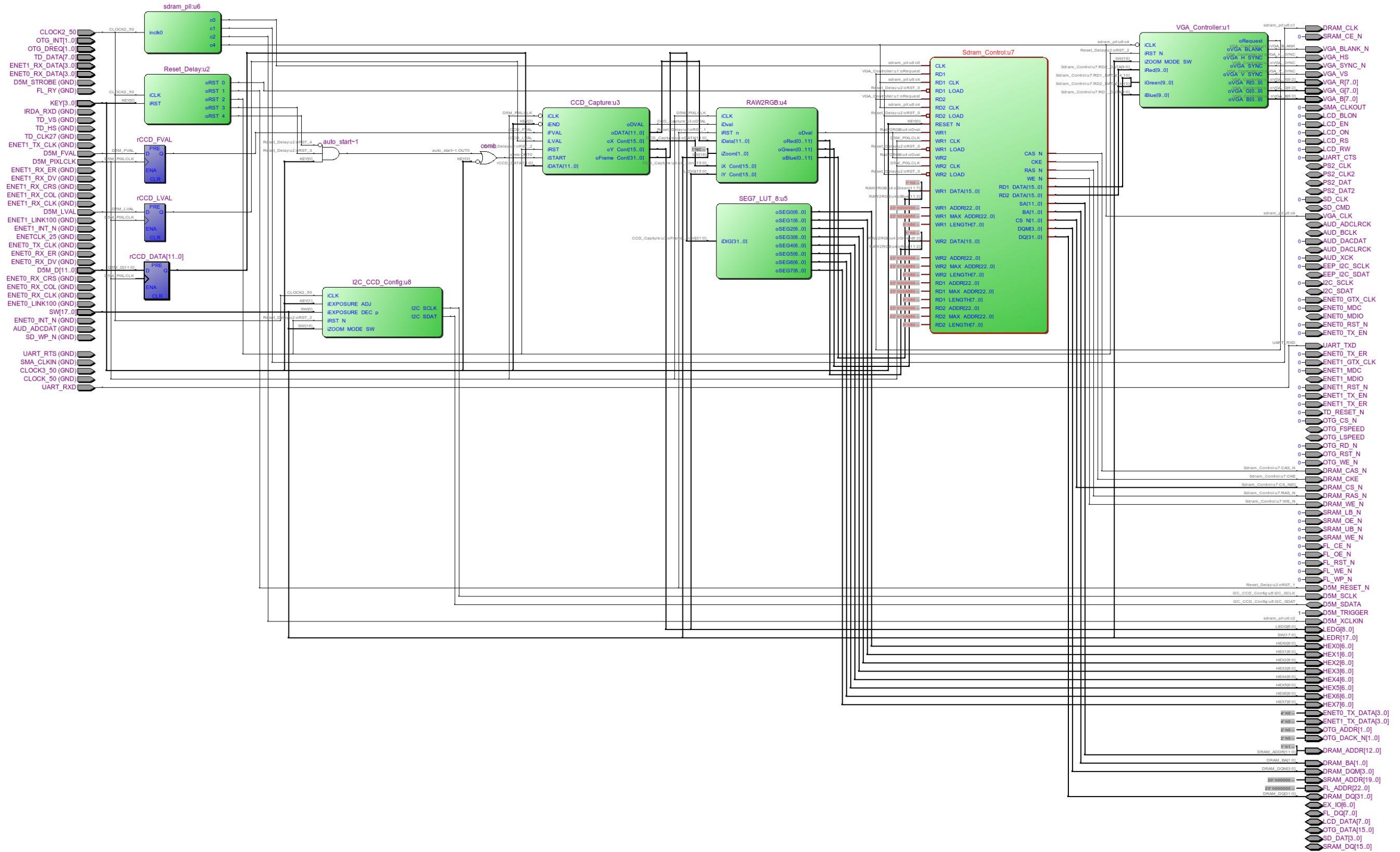
(a) FPGA kartı ve D5M kamera kiti



(b) Referans olarak kullanılan sistem

Şekil A.3: Referans olarak kullanılan sistem

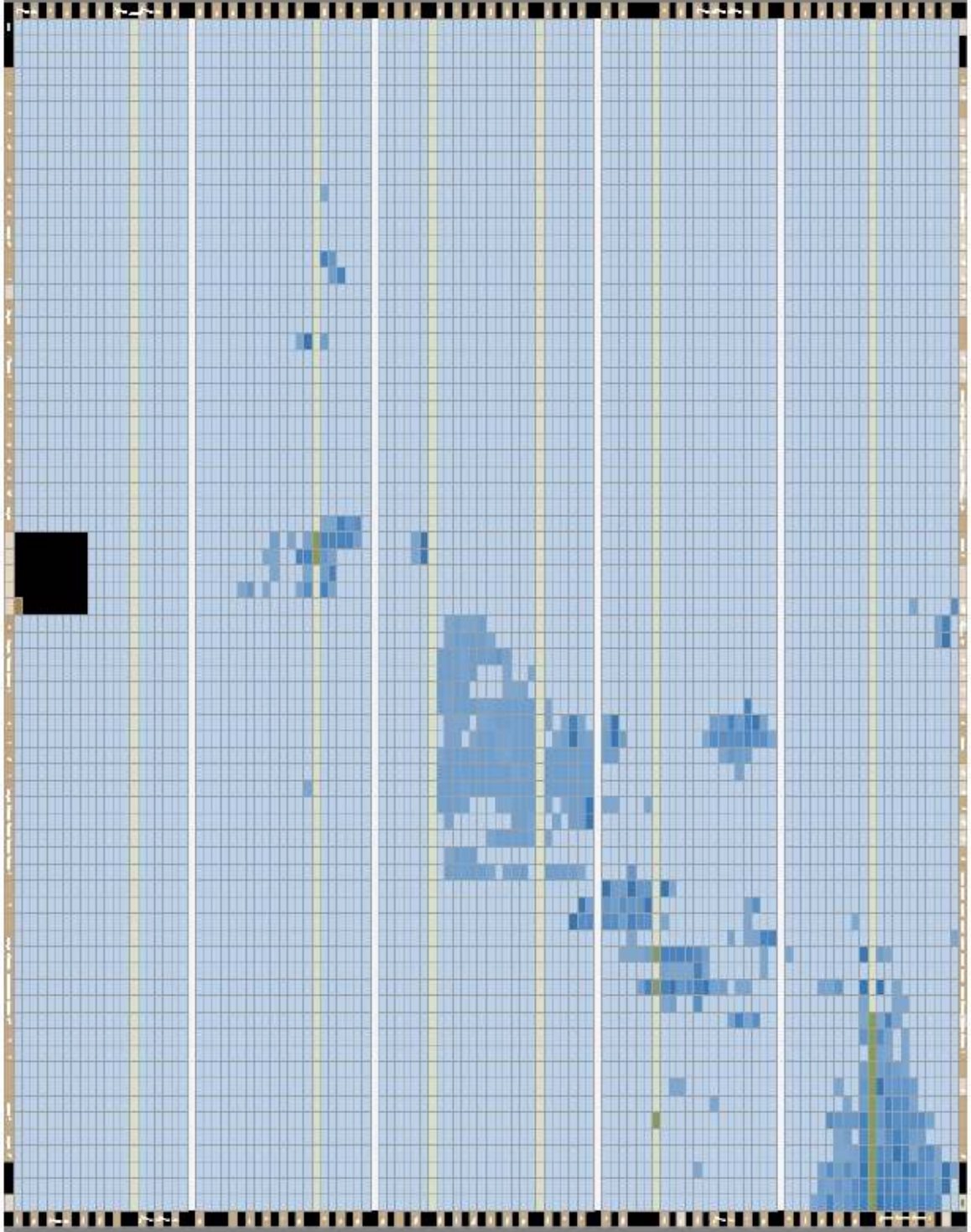
A.4. Mevcut sistemin RTL blok şeması



Şekil A.4: Mevcut D5M sistemi RTL blok şeması

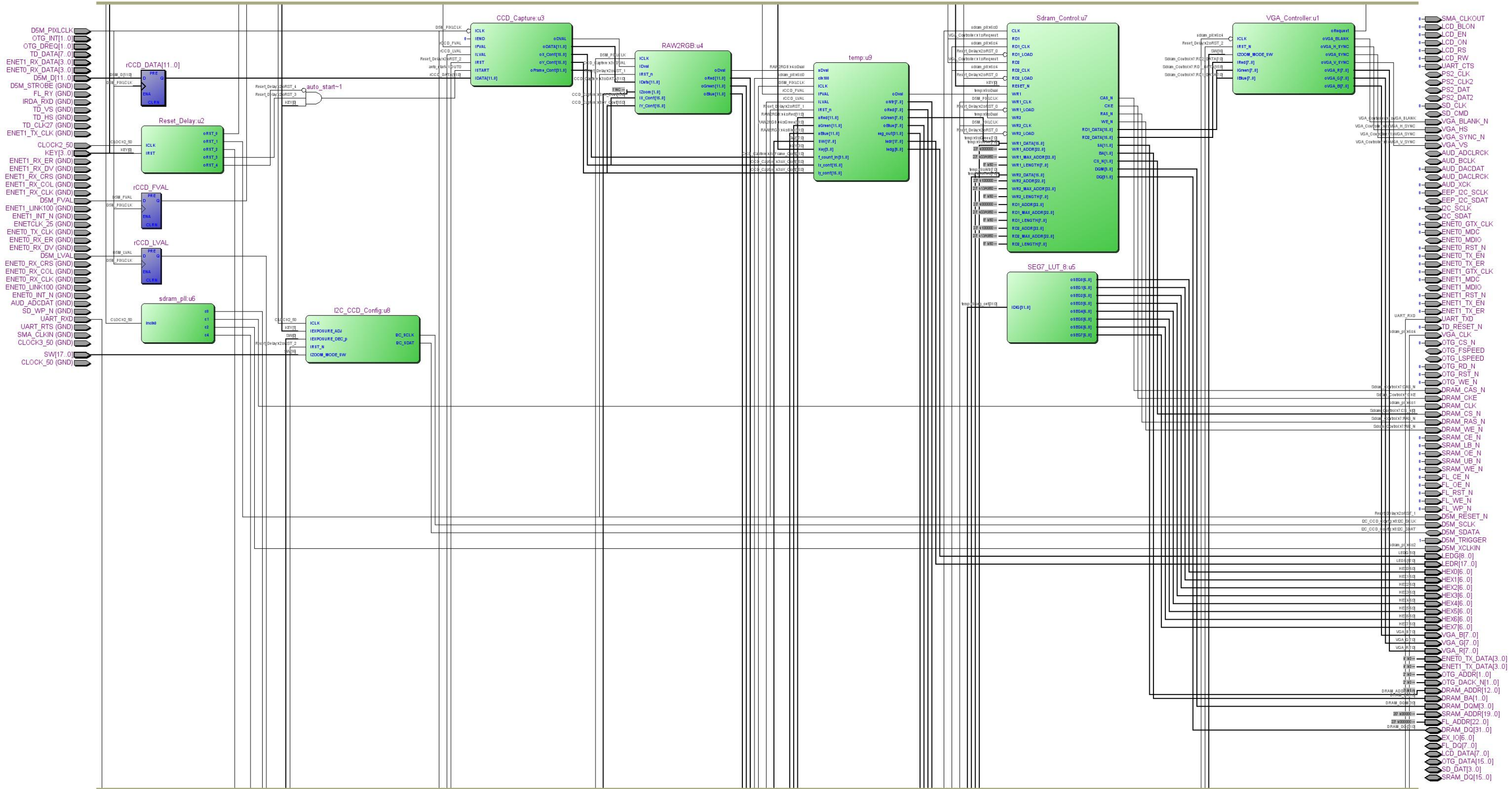
B. Gerçekleştirilen Sistem

B.1. Gerçekleştirilen sistemin FPGA yüzey yerleşimi



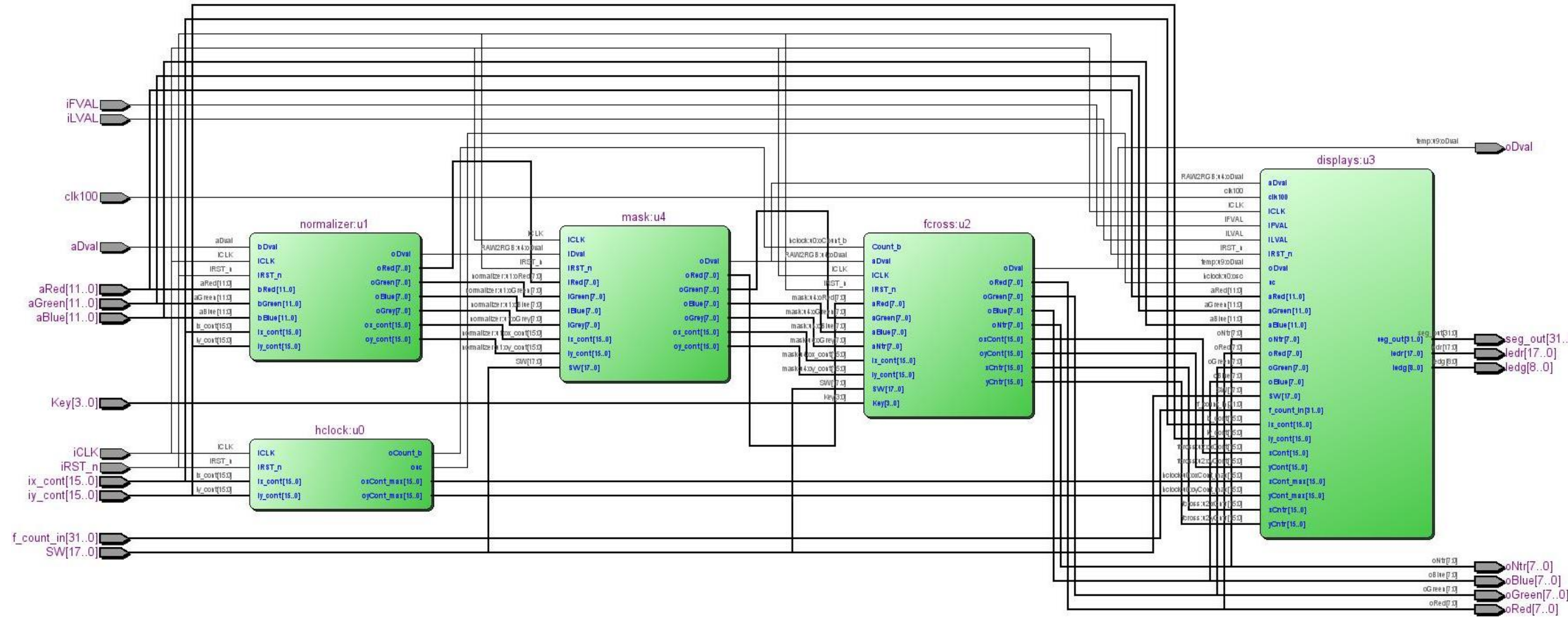
Şekil B.1: Gerçekleştirilen D5M sistemi FPGA yüzey yerleşimi

B.2. Gerçekleştirilen sistemin RTL blok şeması



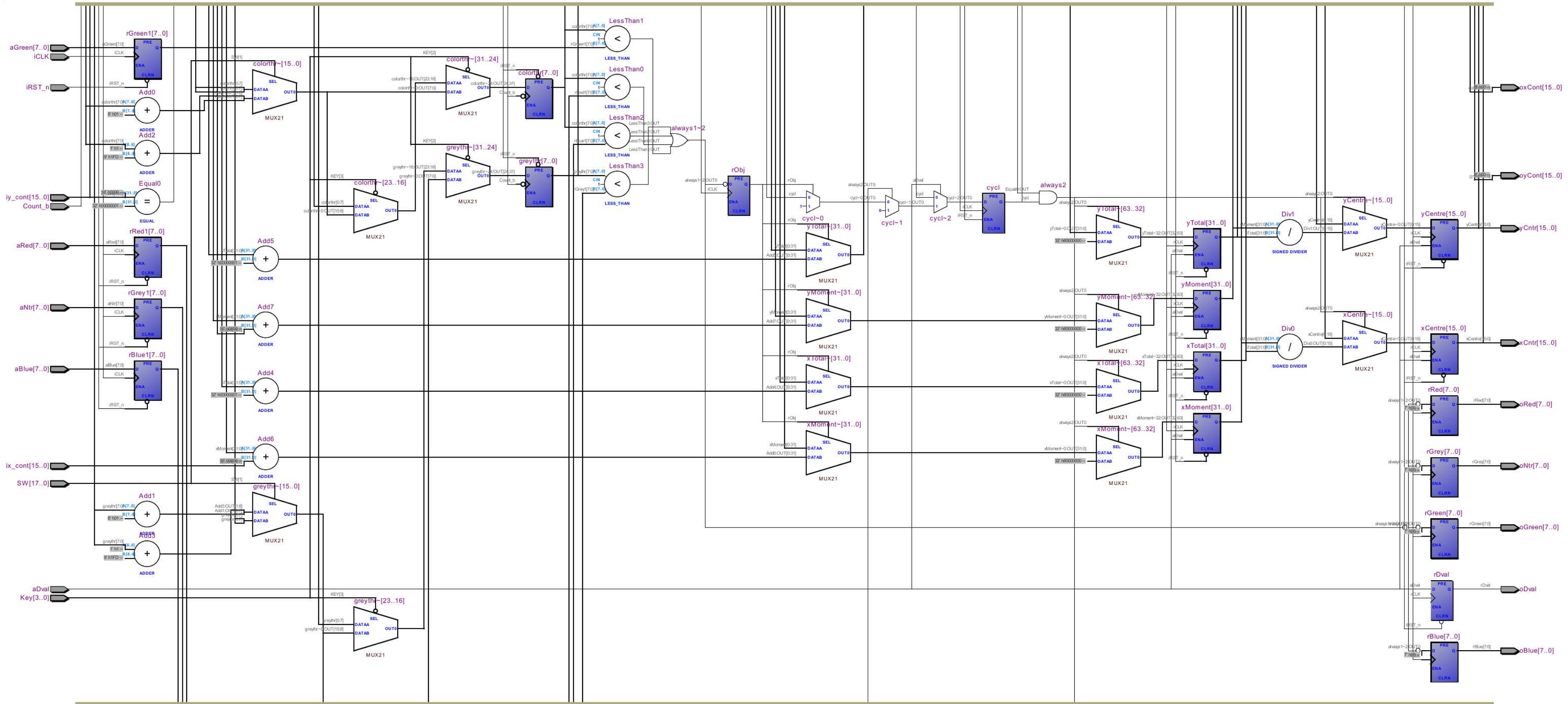
Şekil B.2: Değiştirilmiş sistemin RTL blok şeması

B.3. Eklenen modülün RTL blok şeması



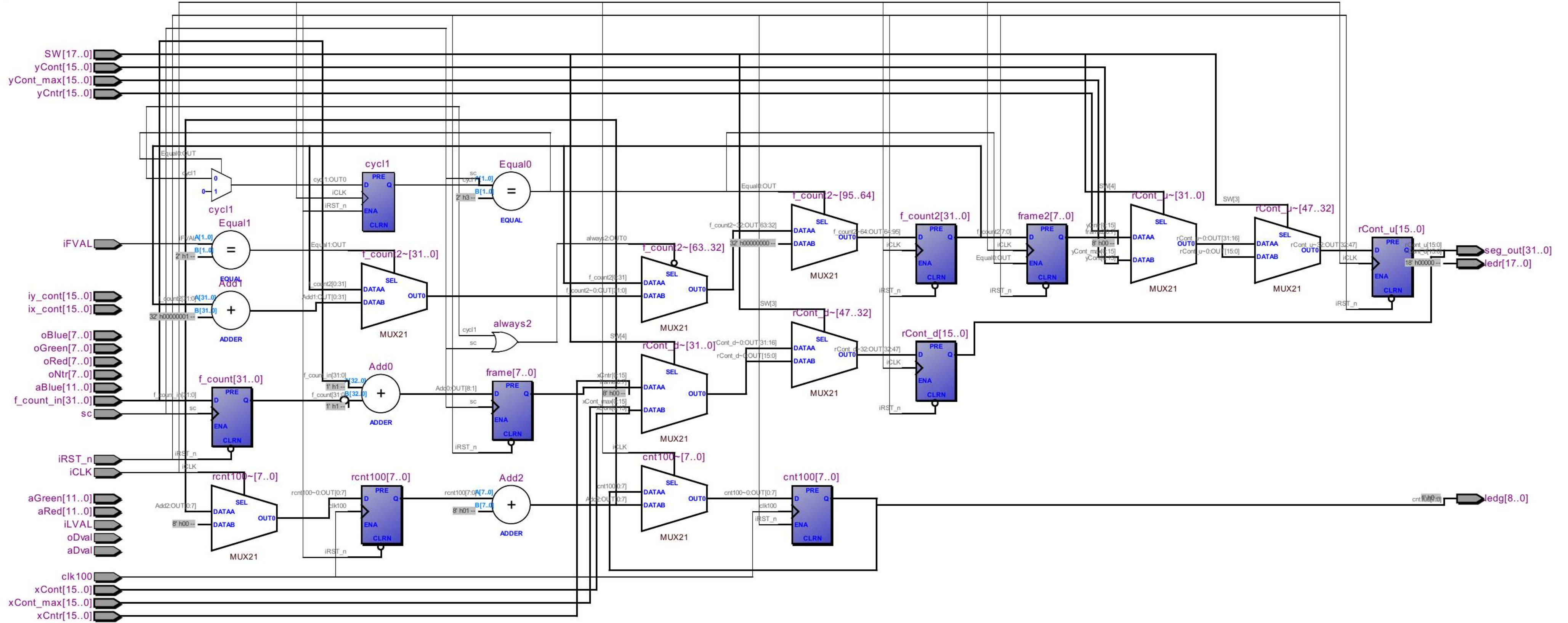
Şekil B.3: Eklenen modülün RTL blok şeması

B.4. fcross modülün RTL blok şeması



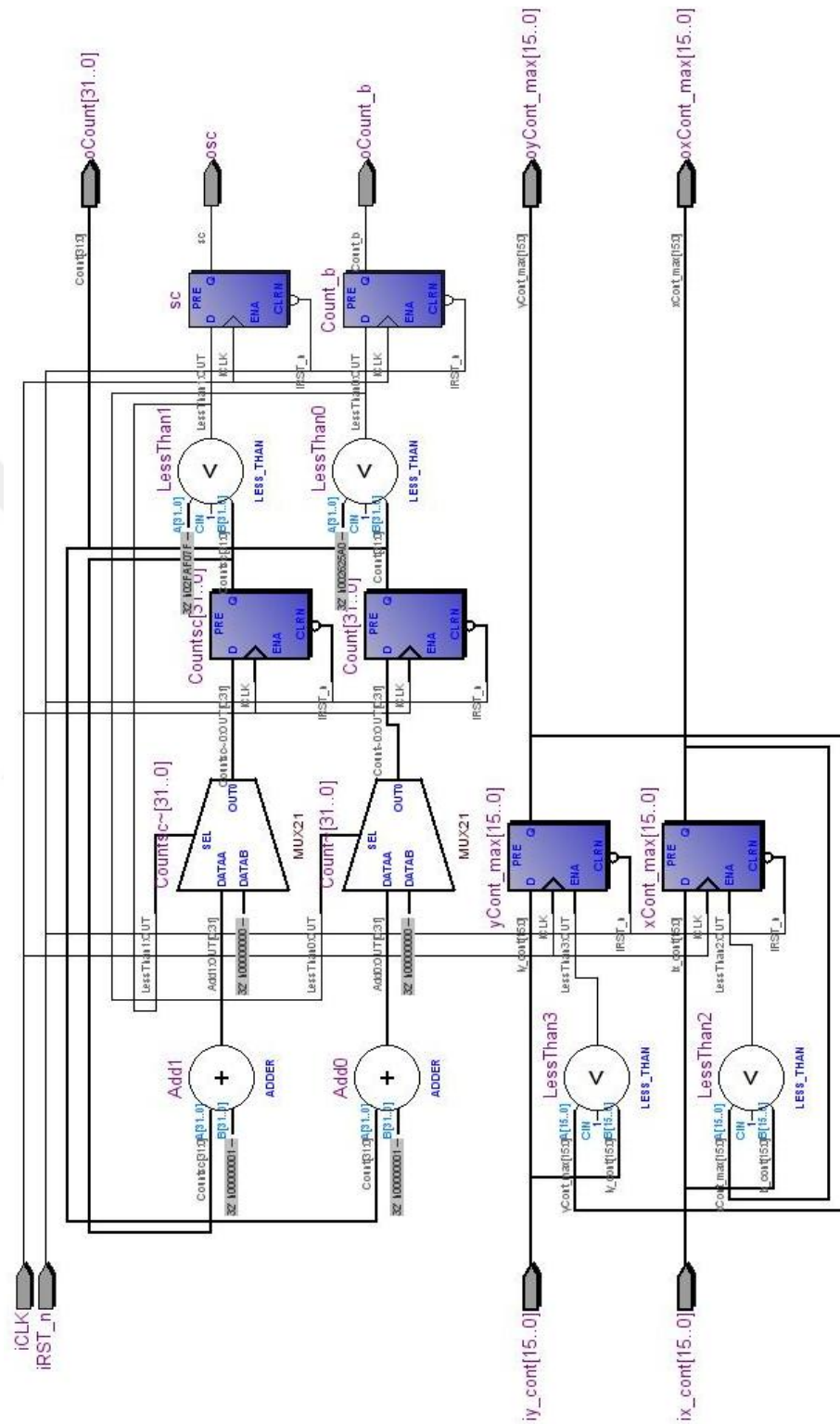
Şekil B.4: fcross modülün RTL blok şeması

B.5. Displays U3 RTL blok şeması



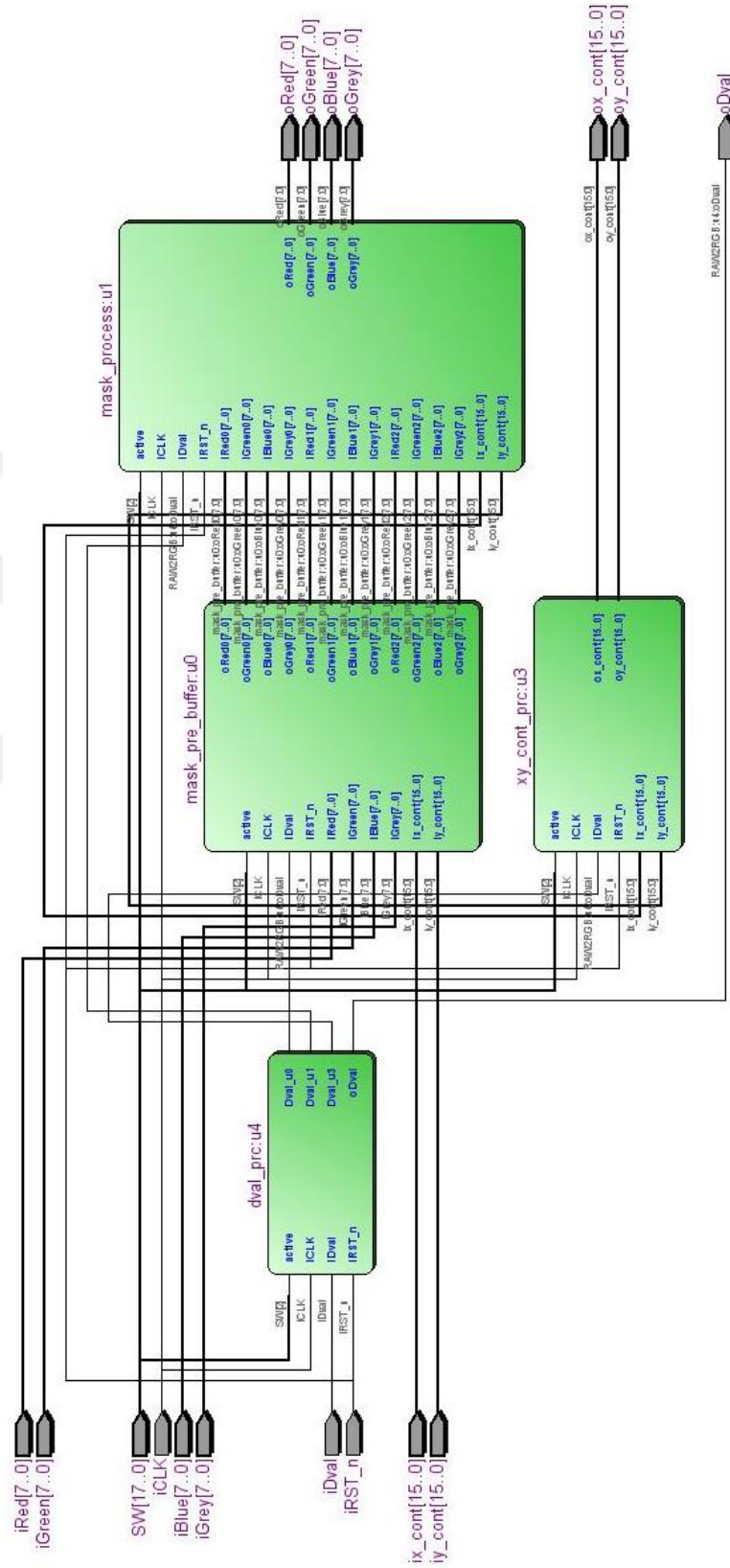
Şekil B.5: Displays U3 RTL blok şeması

B.6. hclock modülü RTL şeması



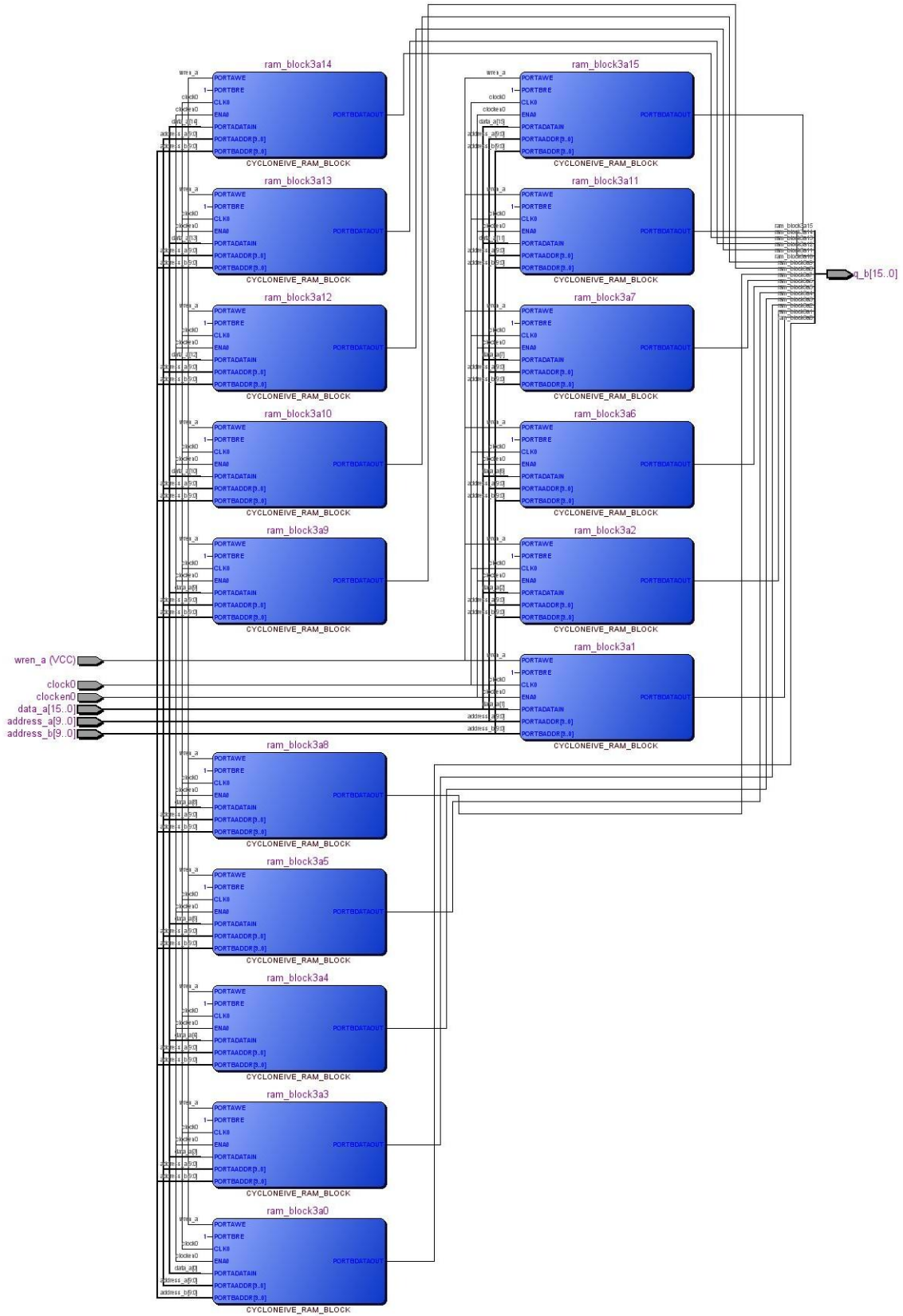
Şekil B.6: hclock modülü RTL şeması

B.7. Mask U4 RTL blok şeması



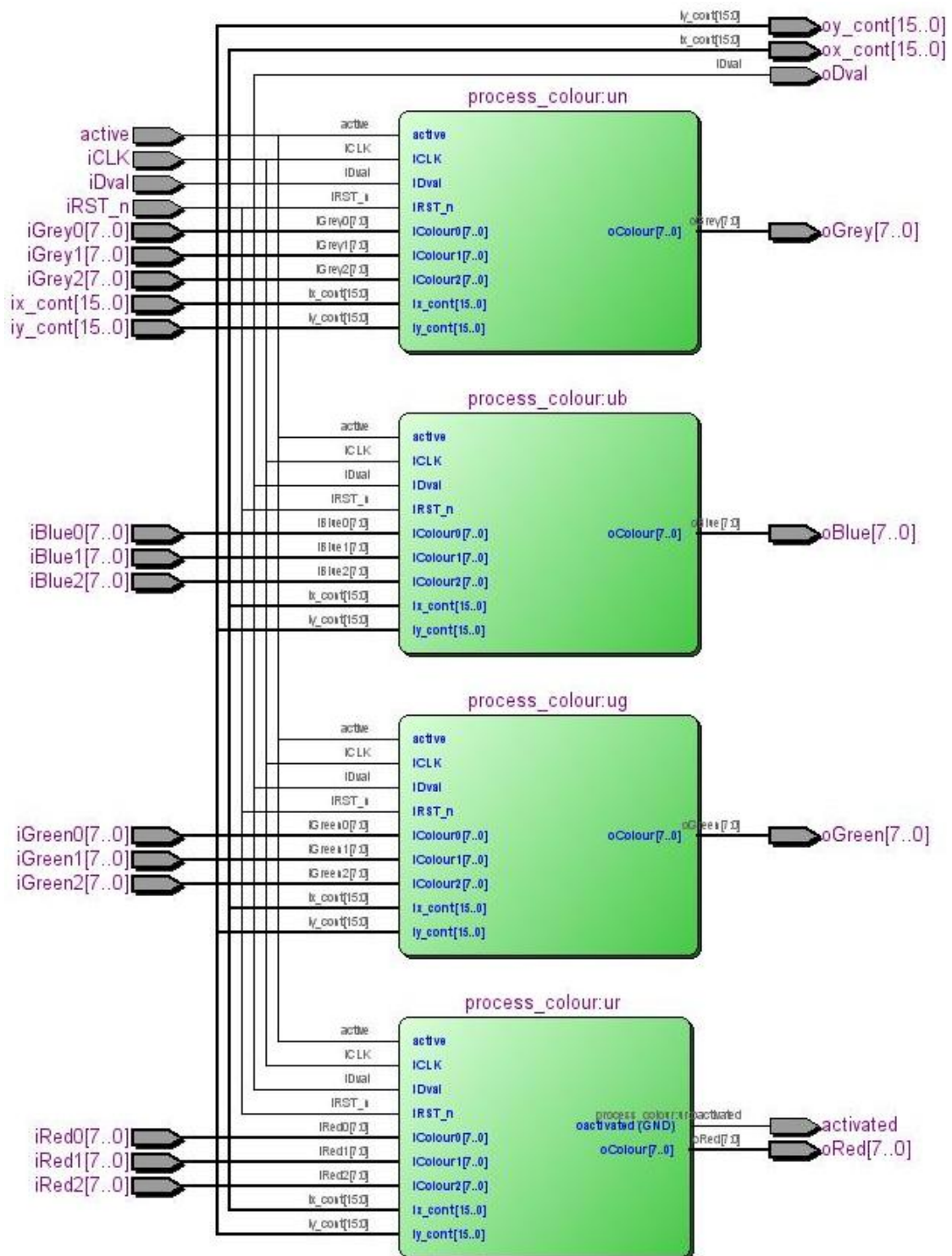
Şekil B.7: Mask U4 RTL blok şeması

B.8. Tamponda kullanılan RAM hafıza RTL blok şeması



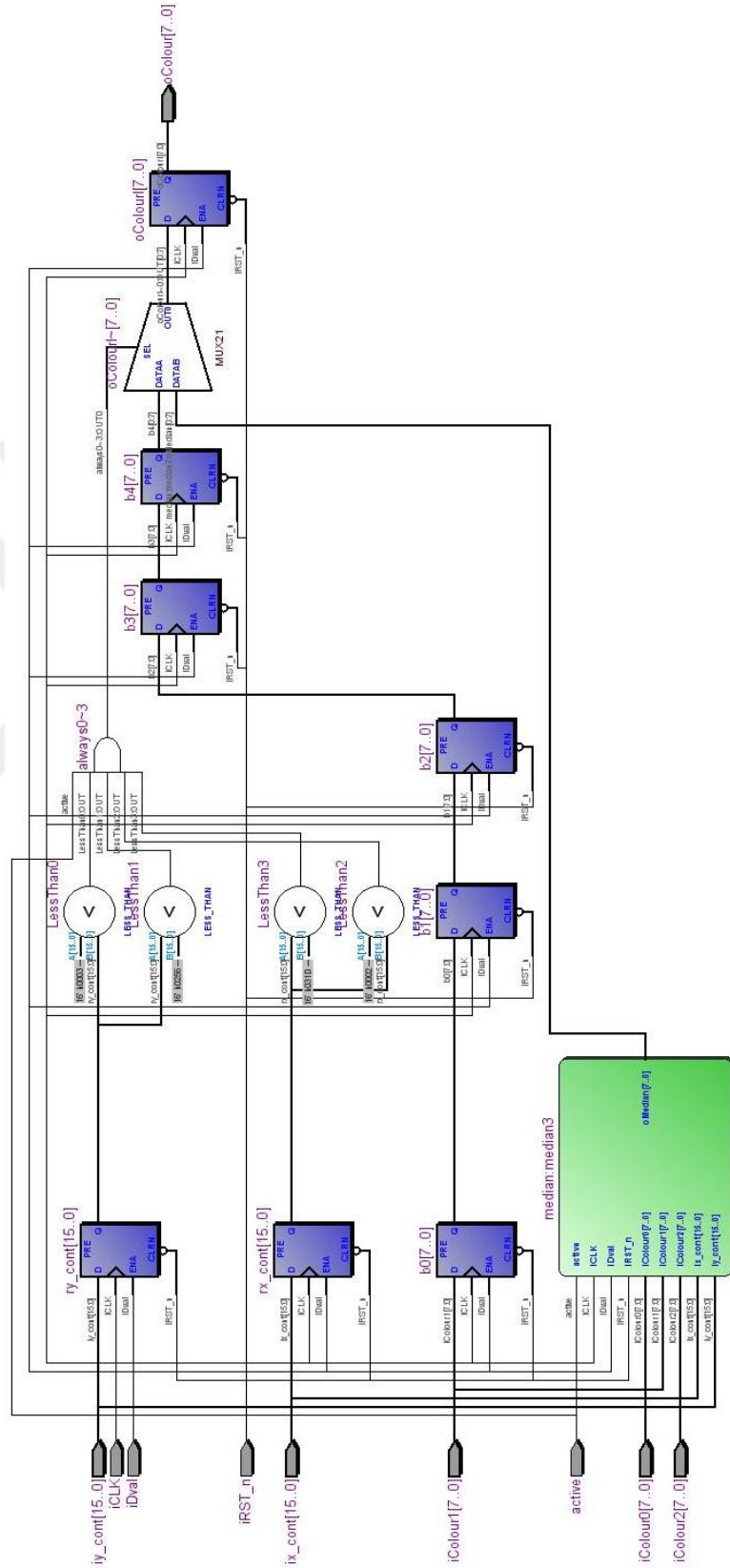
Şekil B.8: Tamponda kullanılan RAM hafıza RTL blok şeması

B.9. Mask process modülü RTL blok şeması



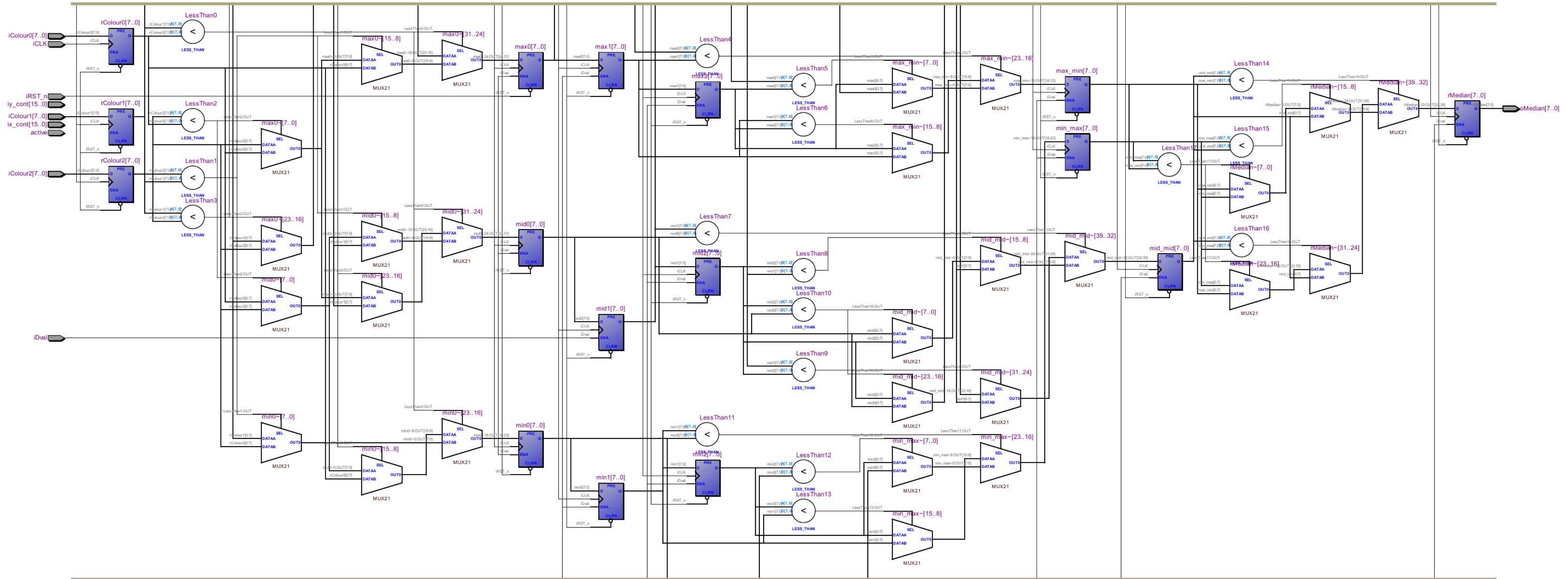
Şekil B.9: Mask process modülü RTL blok şeması

B.10. Colour modülü RTL blok şeması



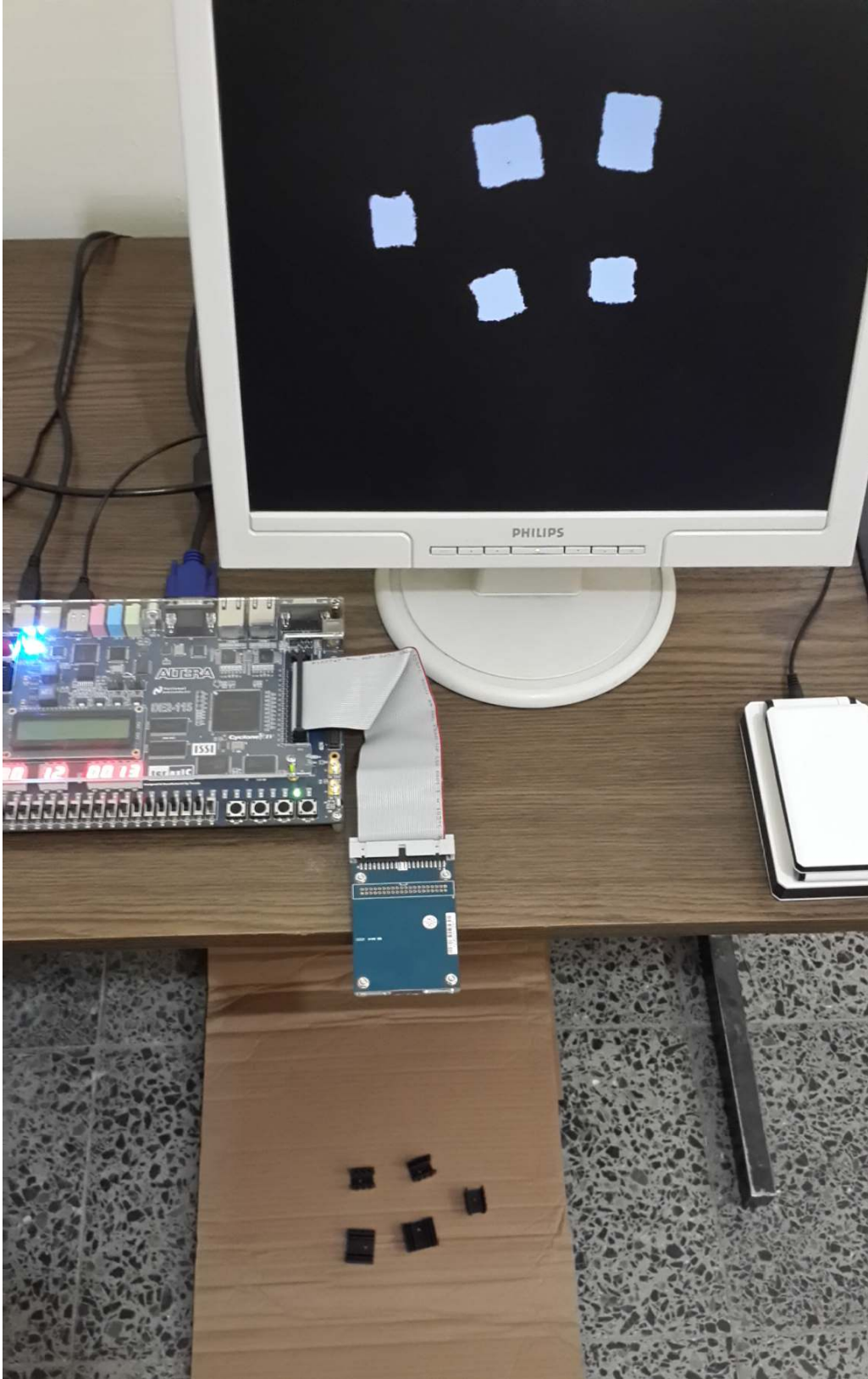
Şekil B.10: Colour modülü RTL blok şeması

B.11. Median modülü RTL blok şeması



Şekil B.11: Median modülü RTL blok şeması

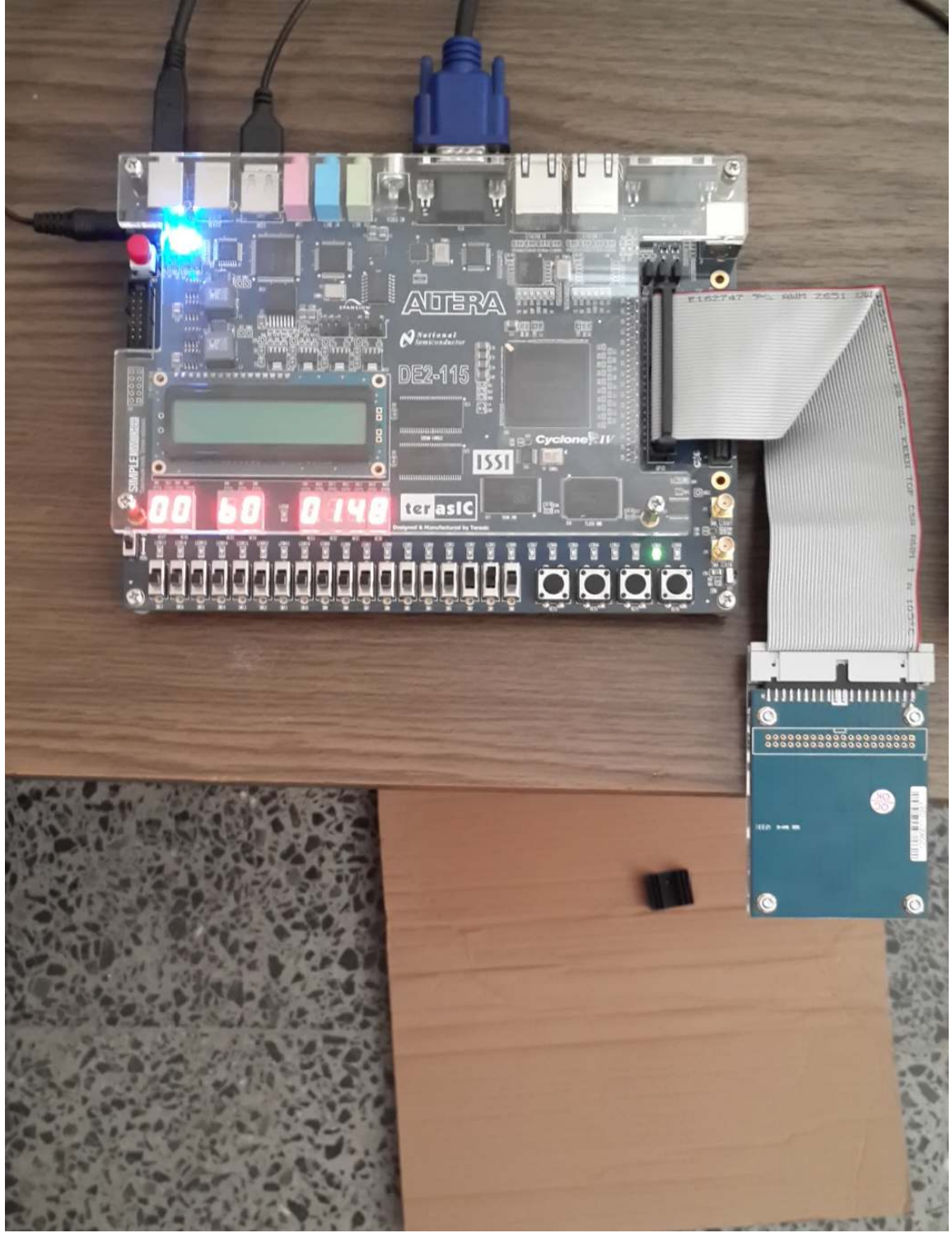
B.12. Çalışma esnasında sistemin görüntüleri



Şekil B.12: Nesne algılama (çoklu nesne)



Şekil B.13: Nesne algılama (tek nesne)



Şekil B.14: Nesne merkezinin göstergeden okunması (tek nesne için)

C. Median Modülünün Verilog Kodları

```
module median( iCLK,
               iRST_n,
               iDval,
               iColour0,
               iColour1,
               iColour2,
               oMedian,
               oDval,
               ix_cont,
               iy_cont,
               active
             );
input          iCLK, iRST_n;
output [7:0]   oMedian;
output        oDval;
input  [7:0]   iColour0;
input  [7:0]   iColour1;
input  [7:0]   iColour2;
input          iDval, active;
input  [15:0]  ix_cont;
input  [15:0]  iy_cont;
reg  [7:0]     max0, max1, max2, mid0, mid1, mid2, min0, min1,
           min2, max_min, mid_mid, min_max;
reg  [7:0]     rColour0, rColour1, rColour2, rMedian;
wire  [7:0]    oColour1;
assign oMedian = rMedian;
always@ ( posedge iCLK or negedge iRST_n ) // input
begin
if( !iRST_n )
begin
rColour0 <= 0;
rColour1 <= 0;
rColour2 <= 0;
end
else
begin
rColour0 <= iColour0;
rColour1 <= iColour1;
rColour2 <= iColour2;
end
end
always@ ( posedge iCLK or negedge iRST_n ) // first stage
begin
if( !iRST_n )
begin
```

```

max0 <= 0;
max1 <= 0;
max2 <= 0;
mid0 <= 0;
mid1 <= 0;
mid2 <= 0;
min0 <= 0;
min1 <= 0;
min2 <= 0;
end
else
  if( iDval)
  begin
    if( rColour0>rColour1 )
    begin
      if( rColour1>rColour2 )
      begin
        max0 <= rColour0; mid0 <= rColour1; min0 <= rColour2;
      end
    else
    begin
      if( rColour0>rColour2 )
      begin
        max0 <= rColour0; mid0 <= rColour2; min0 <=
rColour1;
      end
    else
    begin
      max0 <= rColour2; mid0 <= rColour0; min0 <=
rColour1;
    end
  end
end
else
begin
  if( rColour1<rColour2 )
  begin
    max0 <= rColour2; mid0 <= rColour1; min0 <= rColour0;
  end
  else
  begin
    if( rColour0>rColour2 )
    begin
      max0 <= rColour1; mid0 <= rColour0; min0 <=
rColour2;
    end
  else

```

```

        begin
            max0 <= rColour1; mid0 <= rColour2; min0 <=
rColour0;
        end
    end
end
max1 <= max0;
max2 <= max1;
mid1 <= mid0;
mid2 <= mid1;
min1 <= min0;
min2 <= min1;
end
end
always@ ( posedge iCLK or negedge iRST_n) // mid stage
begin
    if( !iRST_n )
    begin
        max_min <= 0;
        mid_mid <= 0;
        min_max <= 0;
    end
    else
    if(iDval)
    begin
        if( max0 < max1 ) if( max0 < max2 ) max_min <= max0;
                        else max_min <= max2;
        else if( max1 < max2 ) max_min <= max1;
                        else max_min <= max2;
        if( mid0>mid1 )
        begin
            if( mid1>mid2 ) mid_mid <= mid1;
            else
            begin
                if( mid0>mid2 ) mid_mid <= mid2;
                else mid_mid <= mid0;
            end
        end
    end
    else
    begin
        if( mid1<mid2 ) mid_mid <= mid1;
        else
        begin
            if( mid0>mid2 ) mid_mid <= mid0;
            else mid_mid <= mid2;
        end
    end
end
end

```

```

        if( min0 > min1 ) if( min0 > min2 ) min_max <= min0;
                           else           min_max <= min2;
    else           if( min1 > min2 ) min_max <= min1;
                           else           min_max <= min2;
    end
end
always@ ( posedge iCLK or negedge iRST_n) // last stage
begin
    if( !iRST_n ) rMedian <= 0;
    else
        if(iDval)
            begin
                if( max_min>mid_mid )
                    begin
                        if( mid_mid>min_max )    rMedian <= mid_mid;
                        else
                            begin
                                if( max_min>min_max ) rMedian <= min_max;
                                else
                                    rMedian <= max_min;
                            end
                        end
                    end
                else
                    begin
                        if( mid_mid<min_max )    rMedian <= mid_mid;
                        else
                            begin
                                if( max_min>min_max ) rMedian <= max_min;
                                else
                                    rMedian <= min_max;
                            end
                        end
                    end
                end
            end
        end
    end
end
end
endmodule

```

ÖZGEÇMİŞ

Adı Soyadı : Mehmet Rıza SARAÇ
Doğum Yeri : Muş
Doğum Tarihi : 04.04.1980
Medeni Hali : Bekâr
Yabancı Dili : İngilizce

Lise : Muş Anadolu Lisesi 1995 - 1998
Lisans : İnönü Üniversitesi Mühendislik Fakültesi
Elektrik Elektronik Mühendisliği Bölümü 1998 - 2002
Yüksek Lisans : Fırat Üniversitesi Fen Bilimleri Enstitüsü
Mekatronik Mühendisliği Ana Bilim Dalı 2012 -

Deneyimler : 2004 - 2005 Mühendis, Prizma Elektrik Elektronik Ltd. Şti.
2. Organize Sanayi Sitesi Malatya
: 2006 Mühendis, Muş Hasköy KHGB
: 2007 Mühendis, Muş Merkez KHGB
: 2008 - 2010 Mühendis, Muş İl Özel İdaresi
: İmar ve Kentsel İyileştirme Müdürlüğü
: 2011 - Araştırma Görevlisi, Fırat Üniv. Teknoloji Fakültesi
Mekatronik Mühendisliği Bölümü

İletişim : Fırat Üniversitesi Teknoloji Fakültesi
Mekatronik Mühendisliği Bölümü, 23119 - Elazığ
Tel : +90 424 237 00 00 - 4212
eposta: mrsarac@firat.edu.tr, mrza49@gmail.com