

**GENEL AMAÇLI I/O KARTI İLE LED EKRAN MODÜLÜNÜN
KONTROLÜ**

Hasan Hüseyin BAŞ

**YÜKSEK LİSANS TEZİ
ELEKTRONİK BİLGİSAYAR EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**EYLÜL 2006
ANKARA**

Hasan Hüseyin BAŞ tarafından hazırlanan GENEL AMAÇLI I/O KARTI İLE LED EKİRAN MODÜLÜNÜN KONTROLÜ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç.Dr.Abdullah ÇAVUŞOĞLU
Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği ile Elektronik Bilgisayar Eğitimi Anabilim Dalında Yüksek lisans tezi olarak kabul edilmiştir.

Başkan: : Prof. Dr. Ömer Faruk BAY

Üye : Doç.Dr. Abdullah ÇAVUŞOĞLU

Üye : Yrd. Doç. Dr. Hasan Şakir BİLGE

Üye : _____

Üye : _____

Tarih : 28/08/2006

Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

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

Hasan Hüseyin BAŞ

GENEL AMAÇLI I/O KARTI İLE LED EKRAN MODÜLÜNÜN KONTROLÜ

(Yüksek Lisans Tezi)

Hasan Hüseyin BAŞ

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Eylül 2006

ÖZET

Bu çalışmada, Bilgisayarlarda yaygın olarak bulunan PCI veri yolunu kullanarak giriş çıkış işlemlerinde kullanılması için PCI kartı gerçekleştirilmiştir. Kartın varolan kartlardan farkı ucuz, basit yapıda olmasıdır. Kart üzerinde yapılacak küçük değişikliklerle farklı amaçlarda da kullanılabilir. PCI kartının kontrol edeceği günümüzde dış ortamlarda spor salonları, statlar ve şehir için kullanılan renkli LED ekranlarının küçük bir modülü gerçekleştirilmiştir. PCI kartını kullanarak ekran modülü kontrol edilmiştir. Ekran modülünde farklı renk tonları ve hareketli görüntüler başarı ile elde edilmiştir.

Bilim Kolu : 702.1.013
**Anahtar Kelimeler : Bilgisayar giriş çıkış donanımı,
Bilgisayar port programlama**
Sayfa Adedi : 88
Tez Yöneticisi : Doç. Dr. Abdullah ÇAVUŞOĞLU

LED DISPLAY MODULE CONTROLLED BY GENERAL PUPOSE I/O CARD**(M. Sc. Thesis)****Hasan Hüseyin BAŞ****GAZİ UNIVERSITY****INSTITUTE OF SCIENCE AND TECHNOLOGY****September 2006****ABSTRACT**

In this study, by using the PCI (Peripheral Component Interconnect) data bus, which is commonly found in computers, PCI (Peripheral Component Interconnect) card has been achieved in order to be used in the input output processes. The differences of the card from the present cards are that it is cheap and basic. By small making small alterations on the card, it can also be used for a variance of purposes. Today, a small module of the color LED (Light Emitting Diode) screens had been achieved, which can be controlled by PC (Personel Computer) card and which is used for the outdoor places such as sports saloons, stadiums. Screen module was controlled by using PCI (Peripheral Component Interconnect) card. Different color tones and sprite graphic was achieved from these screen modules.

Science Code	: 702.1.013
Key Words	: Computer input output hardware, Computer port programing
Page Number	: 88
Adviser	: Doç. Dr. Abdullah ÇAVUŞOĞLU

TEŞEKKÜR

Yüksek lisans eğitiminin süresince emeği geçen bütün hocalarıma ve özellikle, yüksek lisans tezimin seçiminde ve tamamlamasında esirgemediği bilgi ve tecrübesi için değerli hocam Doç. Dr. Abdullah ÇAVUŞOĞLU ‘na teşekkürlerimi sunarım.

Ayrıca, kaynak toplamamda, yazımında ve tasarlanmasında bana yardımcı olan Servet BOZPOLAT, Burak BAYRAK, Engin KUZU, Candaş ÖRSÇELİK, Ufuk TORUNOĞLU, Osman KUZU, Abdullah EREN, Mustafa YILDIZ, Mahmut AYDINLI, Kerim ÖZDEMİR ’e teşekkürleri bir borç bilirim.

Bu arada, çalışmaları sırasında desteklerini benden eksik etmeyen aileme ve dostlarıma teşekkür ederim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	xi
ŞEKİLLERİN LİSTESİ	xii
SİMGELER VE KISALTMALAR	xvi
1. GİRİŞ	1
2. VERİYOLU STANDARTLARI.....	3
2.1. AGP Veri Yolu.....	3
2.1.1. AGP veri yolunun özellikleri ve çalışması.....	3
2.2. USB Veri Yolu.....	5
2.3. ISA Veri Yolu.....	8
2.4. VESA Veri Yolu.....	9
2.5. PCI Veri Yolu.....	10
2.5.1. PCI yerel yolunun özellikleri	11
2.5.2. PCI yerel yolunun sinyalleri	13
2.5.3. Basit yazma işlemi	15
2.5.4. Basit okuma işlemi.....	17
2.5.5. Transfer modları.....	18

Sayfa

3. PCI KARTINDA KULLANILAN DEVRE ELEMANLARI	20
3.1. Temel Giriş/Çıkış Portu.....	20
3.2. Temel I/O (Giriş/Çıkış) Komutları Ve Kontrol Sinyalleri	20
3.3. Temel Giriş Portu.....	21
3.4. Temel Çıkış Portu.....	22
3.5. 8255 Giriş Çıkış Entegresi.....	24
3.5.1. 8255 yapım çeşitleri.....	25
3.5.2. 8255 çalışma hızı.....	25
3.5.3. 8255 çalışma şekli.....	26
3.5.4. Kontrol kayıtcısı.....	27
3.5.5. Mod ayarları.....	28
3.5.6. Giriş ve çıkış kombinasyonları.....	30
3.5.7. Mod 1.....	30
3.5.8. Mod 2.....	33
3.5.9. Bit kontrolü.....	34
3.6. Programlanabilir Kombinezonsal Devreler(GAL).....	35
3.7. Işıklı Diyot Kullanımı	38
3.7.1. Işıklı diyotların yapısı ve çalışma şekli	38
3.7.2. Işıklı diyotlarda renk.....	40
3.7.3. Işıklı diyotlarda R direnci hesabı.....	42
3.8. Karşılaştırmacılar	44
3.9. Çözücüler	46
3.10. Basit Bir Hafıza Hücresi (Mandal)	48

Sayfa

4. IŞIK VE RENK.....	51
4.1. Renk ve Işık Ayrı Şeydir.....	52
4.2. Beyaz Işık.....	52
4.3. Renk İşaretleri	53
4.4. Renkler ve Bunların Kademelendirilmesi	54
4.5. Renk Doyması.....	55
4.6. Parlaklık, Parlaklık Yoğunluğu.....	55
5. PROJENİN GERÇEKLEŞTİRİLMESİ.....	56
5.1. GAL'ın programlanması	56
5.1.1. Durum diyagramı	56
5.1.2. Gal'in abel dili ile programlanması.....	57
5.2. PCI Kartının Yapımı	61
5.2.1. Adres çözümleme işlemi	61
5.2.2. GAL'in çalışması	62
5.2.3. Verilerin alınması.....	63
5.2.4. 8255 entegrelerin programlanması.....	64
5.3. Kartın Çıkış Portu	67
5.4. Display Kartının Gerçekleştirilmesi.....	69
5.4.1. Konnektör.....	69
5.4.2. Satır ve sütun kontrolü	70
5.4.3. LED hücrelerinin sürülmesi	71
5.5. Devrenin Programlanması.....	72
5.5.1. LED'lerle ana ve ara renklerin elde edilmesi.....	72

Sayfa

5.5.2. 8255 programlanması.....	75
6. SONUÇ VE DEĞERLENDİRME.....	80
KAYNAKLAR	81
EKLER	82
EK-1 PCI Veri Yolu ve PCI Kartı Devresi	83
EK-2 LED Ekran Modülü Devresi.....	84
EK-3 PCI Kart üzerinden LED Ekranı Kontrol eden C programı	85
ÖZGEÇMİŞ	88

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. AGP veri yolunun PCI veri yoluna göre hız farkı görülür.....	5
Çizelge 3.1. 8255 adres ve kontrol yolu ile port secimi.....	27
Çizelge 3.2. 8255 entegresinin mod seçimleri	34
Çizelge 3.3. Yarı iletken malzemeye göre renkler	40
Çizelge 3.4. Hafıza hücresi çıkış değerleri	49
Çizelge 5.1. AD[31:24] bitlerinin kullanımı ve oluşan port numaraları	65
Çizelge 5.2. Renklerin hex kodları.....	78

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. AGP veri yolunun genel görüntüsü	3
Şekil 2.2. USB veri yolunun genel görüntüsü.....	5
Şekil 2.3. ISA veri yolunun genel görüntüsü	8
Şekil 2.4. VESA veri yolunun genel görüntüsü	9
Şekil 2.5. 5 volt PCI slot	10
Şekil 2.6. 3.3 volt PCI slot	11
Şekil 2.7. 5 volt 64 bit PCI slot.....	11
Şekil 2.8. 3.3 volt 64 bit PCI slot.....	11
Şekil 2.9. 5 volt PCI slot ve kartı	12
Şekil 2.10. PCI sinyalleri	13
Şekil 2.11. C/BE# komut yapısı.....	14
Şekil 2.12. C/BE# byte yetki yapısı	14
Şekil 2.13. Yazma işlemlerinde sinyallerin yönü.....	15
Şekil 2.14. Yazma işlemlerindeki sinyallerin zamanlaması.....	15
Şekil 2.15. Okuma işlemlerindeki sinyallerin yönü	17
Şekil 2.16. Okuma işlemlerindeki sinyallerin zamanlaması	17
Şekil 2.17. Normal mod'daki adres ve veri akış şekli.....	18
Şekil 2.18. Burst mod'daki adres ve veri akış şekli	19
Şekil 3.1. RD, WR'dan üretilen IOR ve IOW devresi.....	20
Şekil 3.2. 12H I/O portundan TTL data girişinin yapıldığı basit bir giriş portu	22
Şekil 3.3. Giriş/çıkış port numarası 2FH'ta bilgiyi tutan basit bir çıkış portu	23

Şekil	Sayfa
Şekil 3.4. 8255 entegresinin ayak yapısı	24
Şekil 3.5. 8552A entegresinin blok şeması	24
Şekil 3.6. CPU, 8255 bağlantısı	26
Şekil 3.7. 8255 kontrol kayıtcısı	28
Şekil 3.8. 8255 mod seçimi	28
Şekil 3.9. 8255 giriş modu bağlantı sinyalleri	31
Şekil 3.10. 8255 çıkış modu bağlantı sinyalleri	32
Şekil 3.11. 8255 bit kontrol seçimi	34
Şekil 3.12. GAL'ın pin yapısı	36
Şekil 3.13. GAL'ın iç yapısı	36
Şekil 3.14. GAL'ın programlanabilen hücreleri	37
Şekil 3.15. GAL'ın programlanmış hücre durumu	37
Şekil 3.16. GAL'ın programlanabilen çıkış birimi	38
Şekil 3.17. Yarı iletken malzeme ışık yayılımı	39
Şekil 3.18. LED kesiti	39
Şekil 3.19. Renklerin dalga boyu	40
Şekil 3.20. Işık yoğunluğu ve ideal akım(20mA)	41
Şekil 3.21. İnsan gözünün algıladığı renk dalga boyları	42
Şekil 3.22. LED devresi	42
Şekil 3.23. Akım ve gerilim bağlantısı	43
Şekil 3.24. LED'in mandal entegresi ile sürülme devresi	44
Şekil 3.25. Basit karşılaştırma işlemi	45
Şekil 3.26. 2 bitlik binary karşılaştırma işlemi	45

Şekil	Sayfa
Şekil 3.27. 4 Bitlik binary karşılaştırma işlemi.....	46
Şekil 3.28. 4 Bitlik karşılaştırma yapan entegrenin genel görüntüsü.....	46
Şekil 3.29 Devrenin lojik denklemi gösterilmektedir	47
Şekil 3.30. 4-16'lık decoder giriş ve çıkış şekli	47
Şekil 3.31. 741574 4 Hattan 16 hata decoder.....	48
Şekil 3.32. Basit bir hafıza hücresi prensibi.....	49
Şekil 3.33. Hafıza hücresinin S tipi flip-flop'la çalışma.....	49
Şekil 3.34. Hafıza hücresinin RS flip-flop tipi	50
Şekil 3.35. RS flip-flop.....	50
Şekil 4.1. Işık karışımı	51
Şekil 4.2. Beyaz ışık.....	52
Şekil 4.3. Renk türü ve krominans	54
Şekil 5.1. PCI yazma sinyallerine göre oluşturulan diyagram.....	56
Şekil 5.2. Adres çözümleme devresi	61
Şekil 5.3. PCI sinyallerini düzenleyen GAL devresi	62
Şekil 5.4. Verilerin tutma işlemini yapan devre kısmı.....	63
Şekil 5.5. 8255'lerin bağlantı yapıları.....	64
Şekil 5.6. PCI kartının blok şekli	67
Şekil 5.7. Kart konektörünün yapısı.....	68
Şekil 5.8. Display kartının konektör şekli.....	69
Şekil 5.9. Satır ve sütun tarama.....	70
Şekil 5.10. Satır ve sütunun belirlendiği devre	70

Şekil	Sayfa
Şekil 5.11. LED’lerin sürülme devresi.....	71
Şekil 5.12. LED’lerin renk sıralaması.....	72
Şekil 5.13. LED’lerin görüntüleri	72
Şekil 5.14. Kırmızı renk elde edilmesi.....	73
Şekil 5.15. Mavi renk elde edilmesi.....	73
Şekil 5.16. Yeşil renk elde edilmesi.....	73
Şekil 5.17. Beyaz renk elde edilmesi	73
Şekil 5.18. Sarı renk elde edilmesi.....	74
Şekil 5.19. Turuncu renk elde edilmesi.....	74
Şekil 5.20. Mor renk elde edilmesi	74
Şekil 5.21. Kahverengi renk elde edilmesi.....	74
Şekil 5.22. Türkuaz renk elde edilmesi.....	74
Şekil 5.23. Siyah renk elde edilmesi	75
Şekil 5.24. Farklı renk tonu elde etme	75
Şekil 5.25. 4x6’lık Panelin satır-sütun yapısı	76
Şekil 5.26. Hücrelerin satır sütun yapısı	76
Şekil 5.27. Birden fazla panelin oluşturduğu yapı	77
Şekil 5.28. 8 Bitin kontrol ettiği hücre yapısı	78
Şekil 5.29. Kart ve panelin ortaklaşa çalışma	79

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	Açıklama
Hz	Hertz
KB	Kilobayt
KHz	KiloHertz
PC	Personel Computer (Kişisel Bilgisayar)
PCI	Peripheral Component Interconnect (Yan Bileşen Bağlantısı)
RAM	Random Access Memory (Rastgele Erişimli Bellek)
mA	Miliamper
mV	Milivolt
V	Volt
ms	Milisaniye
Mhz	MegaHertz
A/S	Analog Sayısal
dB	Desibel
nm	NanoMetre
LED	Light Emitting Diode
RD	Read(Okumak)
WR	Write(Yazmak)
IN	Input(Giriş)
I/O	Input/Output(Giriş/Çıkış)
IOR	Input/Output Read(Giriş/Çıkış Okuma)
IOW	Input/Output Write(Giriş/Çıkış Yazma)
IORQ	Input/Output ReQuest(Giriş/Çıkış İstemek)
OUT	Output(Çıkış)

Simgeler	Açıklama
TTL	Transistor-Transistor Logic(Transistör Mantığı)
CMOS	Complementary-symmetry/metal-oxide semiconductor
NMOS	Negative-channel metal-oxide semiconductor
STB	Strobe Input
IBF	Input Buffer Full Output(Giriş Tamponlu Tam Çıkış)
INTE	Interrupt Enable(Kesmeyi Engelle)
CPU	Central Processing Unit(Merkezi İşlem Birimi)
OBF	Output Buffer Full Output(Çıkış Tamponlu Tam Çıkış)
ACK	Acknowledge input(Girişi kabul et)
EISA	Extended Industry Standard Architecture (Genişletilmiş endüstriyel mimari standardı)
AGP	Accelerated Graphics Port(Gelişmiş Ekran Çıkışı)
MCA	Master of Computer Applications
VESA	Video Electronics Standards Association(Video Elektronik Mimari Standardı)
ISA	Industry Standard Architecture
PLD	Programmable Logic Devices(Programlanabilir Mantıksal Aygıt)
EECMOS	Electrically Erasable Complementary Metal Oxide Semiconductor
IRLED	Infrared Light Emitting diode
XOR	Exclusive Or
LSB	Least Significant Bit(En Önemsiz Bit)
MSB	Most Significant Bit(En Önemli Bit)

Simgeler**Açıklama****BCD**

Binary Coded Decimal(İkili Kodlanmış Onlu)

RS

Reset Set(Sıfırlama Birleme)

ABELAdvanced Boolean Equation Language
(Gelişmiş Booleen Eşitlik Dili)

1. GİRİŞ

Günümüzde çok hızlı gelişen teknolojilerin başında bilgisayar sektörü gelmektedir. İnsanlar günlük yaşamlarında bilgisayarı veya bilgisayar tarafından kontrol edilen ses, görsel, finansal, eğlence, güvenlik, tasarım vb... kullanmaktadır-lar. Bilgisayar tarafından kontrol edilen cihazlar bilgisayarla iletişim kurabilmesi için kendisi için yapılmış özel bir kart veya standart veri yolu kullanmak zorundadır.

Hazır mevcut olan kartlar tek bir işleve sahip, günümüzde ana kartlarda mevcut olmayan veri yolunu (ISA) kullanan veya üzerinde modifiye edilemeyecek kadar yetersiz olan özelliklere sahiptirler.

Gerçekleştirilen kart genel amaçlı, çok yönlü ve PCI veri yoluna sahiptir. Genelde PCI veri yoluna sahip kartlar, birden fazla sinyalin kontrolü amacıyla kullanılan mikro kontrollü veya FPGA karmaşık uç yapılarına sahip olduğundan dolayı maliyetleri oldukça yüksektir. Bu tezde gerçekleştirilen GAL daha basit, programlaması kolay ve diğer kartlara nazaran maliyeti düşüktür.

PCI kartımız çıkış amaçlı kullanılmaktadır. Basit bir değişiklik ile giriş amaçlı kullanılabilir. Kartımızda 8255 programlanabilen çıkış entegreleri kullanılmışken 8254, ADC veya ihtiyaç durumunda daha farklı bir entegre devreye monte edilerek çok yönlü amaçlarda kullanılabilir.

Günümüzde ortaöğretimde ve yükseköğretimde kurumlarında mikro işlemci veya bilgisayar deney setleri işlevselliğini yitirmiş, kullanılmayan özelliklere sahiptir. Kartımız günümüz şartlarına uygun geliştirilen bir deney seti ile bilgisayar arasında ara yüz görevi yapabilecek özelliklere sahiptir.

Gümüşkaya'nın çalışmasında üzerinde paralel, seri, ve sayısal analog dönüştürücüleri birlikte bulunan ISA kartı gerçekleştirmiştir. Ayrıca kartla birlikte birçok uygulama devresi ile deney yapma imkânı vermektedir. Günümüzde ISA yuvaları ana kartlarda bulunmadığından uygulama alanı kalmamıştır [1].

Bery'nin adres çözümlemede PAL kullanmaktadır. Böylelikle adres çözümlemede kullanılan birçok devre elemanı artık tek bir PAL ile yapılabilmektedir [2].

Ehud'un yapmış olduğu çalışmada PCI kartında CPLD ve FPGA kullanmıştır. FPGA PCI sinyallerinin çözümünde ve tak çalıştır özelliğine sahiptir. Kart PCI veri yolu kullanılarak hafıza alanı olarak kullanılmaktadır [3].

Bu çalışmada, gelen amaçlı kullanılabilen PCI kart ve kartın kontrol ettiği LED'lerden oluşturulan ekran modülü gerçekleştirilmiştir.

Tezin ikinci bölümünde, veri yolu standartları tanıtilen PCI veri yolunun çalışması, adresleme şekli ve sinyaller hakkında bilgiler verilmiştir.

Üçüncü bölümde PCI kartında kullanılan devre elemanları ve LED lerin çalışma şekli tanıtılmıştır.

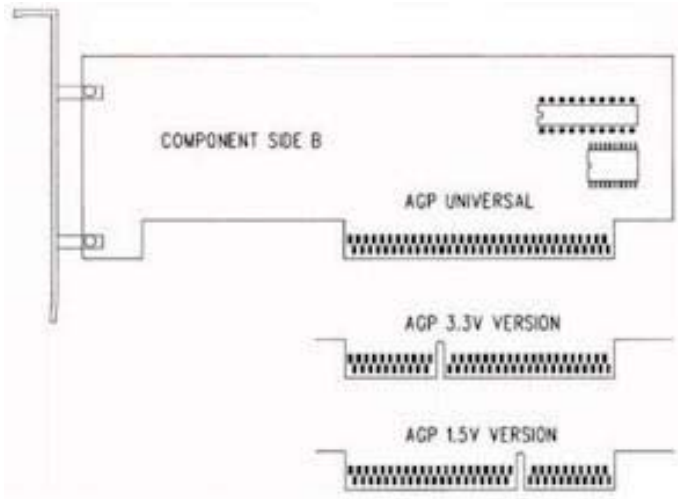
Dördüncü bölümde beyaz ışık bileşenleri ve rengin oluşumu hakkında bilgi verilmiştir.

Beşinci bölümde PCI sinyallerinin çözümde kullanılan GAL'ın programlaması, PCI kartını oluşturan sinyal çözümleme, adres çözümleme, paralel çıkış veren entegrelerin programlaması, ekran kartının çalışması ve LED'lerden renk elde etme gerçekleştirilmiştir.

Altıncı bölümde tezle ilgili genel sonuçlar verilmiştir.

2.VERİYOLU STANDARTLARI

2.1.AGP Veri Yolu



Şekil 2.1. AGP veri yolunun genel görüntüsü

Hızlandırılmış grafik port'u anlamına gelen AGP, ekran kartları için kullanılan yeni bir veri yoludur. AGP veri yolları Pentium II ve üstünü destekleyen ana kartlarda bulunur. PCI veri yolu ile aralarındaki temel fark; AGP'ler 128 KB'a varan büyük grafik dokularını (texture) ekran kartı belleğinin dışında, sistem belleğinden de yararlanarak işler. Bu sayede performansta artış sağlanır.

2.1.1.AGP veri yolunun özellikleri ve çalışması

AGP veri yolunu sadece ekran kartları kullanır. Bu nedenle veri yolunun tüm bant genişliği ekran kartları için çalışmış olur. Tüm bant genişliği sadece ekran kartı için kullanıldığından, bu yolu kullanan ekran kartlarının performansı PCI veri yolunu kullanan ekran kartlarına nazaran oldukça yüksek olur.

AGP veri yolunun bu özelliğine kenardan adresleme denilir. Normal durumlarda grafik kartı ile CPU arasında tek şeritli bir yol vardır. Veri ya CPU dan grafik

kartına, ya da grafik kartından CPU ya gider. Ama eğer ekran kartına doğru veri akarken CPU ya bir komut göndermek gerekirse, bu veri akışını kesip, CPU ya komutu gönderip veri akışına kaldığı yerden devam etmek gerekir. Buda performansın düşmesine sebep olur. İşte Sideband Addressing olayında işin içinde CPU ya komut göndermek için bir veri yolu daha bulunur. Komutlar bu yolu kullanarak her istendiğinde iletilebilir ve performans kayıpları önlenir.

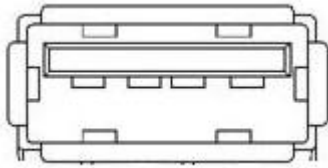
Bu özelliği bulunmayan kartlarda CPU, grafik kartının işlemcisine bir şeyler göndermek istediğinde önce bunları sistem belleğine yazar. Grafik kartı da kendisine gönderilen verileri sistem belleğine eriştikten sonra işleme koyar. Bu da zaman ve hız kaybına sebebiyet verir. Fast Write kullanabilen kartlarda ise CPU sistem hafızasını kullanmadan direkt olarak grafik kartının işlemcisine erişebilme yeteneği kazanır. Bu sayede veri akışı daha çabuk gerçekleşir. Bu özellik AGP veri yolunu kullanan ekran kartlarında bulunur.

Yine AGP veri yolunu kullanan ekran kartlarındaki bu özellik sayesinde ekran kartları bir komut yolladıktan sonra cevabın gelmesini beklemeden bir diğer komutu yollayabilir. Pipelining özelliği sayesinde cevabın gelmesini beklemeden bir sonraki komut ön belleğe aktarılır ve bir sonraki komutun işlenmesine geçilir. Bu sırada cevap zaten gelmiş ve ön bellekte bekleyen komut gönderilmiştir. Arka bellekte tutulan komut tekrar ön belleğe yüklenir ve bir sonraki hazırlanır. İşte bu şekilde 1'e 2 kat daha hızlı bir performans sağlanır [3].

Çizelge 2.1.AGP veri yolunun PCI veri yoluna göre hız farkı görülür.

	Veri Yolu Frekans	Örnekleme Hızı (Clock devri başına)	Bant Genişliği	Veri Transfer Hızı
PCI	33Mhz	1	33Mhz	133MB/s
AGP 1x	66Mhz	1	66Mhz	266MB/s
AGP 2x	66Mhz	2	133Mhz	512MB/s
AGP 4x	66Mhz	4	266Mhz	1.066GB/s
AGP 8x	66Mhz	8	533Mhz	2.133GB/s

2.2.USB Veri Yolu



Şekil 2.2. Usb veri yolunun genel görüntüsü

Giriş Harici aygıt kurulumu, hem yeni hem de usta bilgisayar kullanıcıları için zor bir iştir. Gerekli bağlantıları yapmanın yanısıra, aygıtı konfigüre etmek de zaman alır. IRQ ve I/O adresleri, bir aygıtın, bilgisayar sisteminde çakışma yaşamadan çalışması için doğru şekilde ayarlanmalıdır. Bazen, bir aygıt, sistemin boot işlemi esnasında takılı olmadığı için çalışmayabilir. Genel olarak, bu problemi çözmenin tek yolu, aygıtı bağlayıp sistemi yeniden başlatmaktır.

Universal Serial Bus (USB), yukarıda anlatılan zorlukların üstesinden gelmek için bulunmuş çözümlerden birisidir. Yukarıdaki eksiklikleri tamamlamak için USB, tek bir bağlantı tipi ve anında takma/çıkartma özellikleri sunar. Bu iki özellik sayesinde, USB tipi bir çevresel aygıtı takmak oldukça kolaylaşır. Sistemin açık veya kapalı olmasına dikkat etmeden, USB aygıtı sisteme bağlayın. Çalışan bir sistemde, aygıtın takıldığı anlaşılır ve aygıt konfigürasyonu sistem tarafından otomatik olarak yapılır.

Şimdiye kadar, USB bir değişim geçirmiştir. İki spesifikasyon sürümleri 1.0 ve 1.1'dir. USB 1.1'in asıl amacı, USB 1.0'da fark edilen problemlerin çözülmesi ve bazı belirsizliklere açıklık getirilmesidir. USB 2.0'ın ise 2000'in ilk çeyreğinde çıkması beklenmektedir. USB 2.0 sayesinde orijinal USB kablo ve bağlantıları değişmeden 480Mbps bant genişliği mümkün kılınacaktır. Bu, mevcut standarttaki genişliği 40 katına çıkartacaktır (USB 1.1'de 12Mbps). Ek olarak, USB 2.0 mevcut USB sürümleriyle de geriye uyumlu olacaktır. Bu yazıda "USB" terimi, ayrıca belirtilmedikçe USB 1.1 olarak anlaşılmalıdır [3].

Özellik Listesi USB, IEEE1394'ün benzeri özelliklere sahiptir. Aşağıda kısa açıklamalarıyla USB özelliklerinin özetini bulabilirsiniz:

- 1) Anında Takma/Çıkartma, PnP: USB aygıtları, aygıt takma/çıkartma işlemini tespit etme ve konfigürasyonunu otomatik olarak yapma yeteneğine sahiptir.
- 2) Seri Veri Yolu: Bir çift farklı sinyal, hem veri gönderimi hem de veri alımı için kullanılır.
- 3) Eşsüreli Aktarım: Kamera gibi video uygulamalarında, sabit bit oranına yakın aktarım gereklidir. Eşsüreli aktarım, veri kesinliği yerine ayrılmış aktarım oranı sözü verir.
- 4) Kablo Gücü: Kendi elektriğini sağlayamayan aygıtlar için, kablo gücü, USB'nin kablosu aracılığıyla mümkündür.
- 5) Tam veya Düşük Çalışma Hızı: USB' de iki çalışma hızı desteklenir: Tam hız için 12Mbps ve düşük hız için 1.5Mbps. Düşük hız tanımlama, üreticiler için maliyeti düşürme amaçlı bir opsiyondur.

USB Bağlantısı Bir USB kablosu, dört iletkenle oluşur: iki adet veri yolu (kablo) gücü için ve iki adet de farklı sinyal çifti için. Pin tanımlamaları aşağıdaki şekildedir:

Pin 1. VSUB: Veri yolu gücü, kaynakta +5V

Pin 2 ve 3. D- ve D+: Farklı sinyal çifti

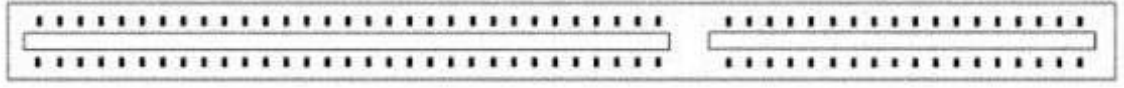
Pin 4. GND: Toprak

USB Konfigürasyon İşlemi USB, seçimli bir veri yoludur. Bütün veri aktarımları, sunucu kontrol aygıtı tarafından başlatılır. Göbekler, bir USB aygıtının kendi portlarından birine takıldığını belirten durum sinyallerine sahiptirler. Sunucu, bir aygıtın takılıp çıkartılmasını anlamak için, düzenli aralıklarla bütün göbekleri sorgular. Bir aygıt takıldığında, göbek, portu açması için sunucudan emir alır, bir ünite yükünü (100mA) hazır hale getirir ve ilgili porta bağlanan aygıtı yeniden başlatır. Bu anda, aygıt standart adrese (yani sıfır) cevap verir. Sunucu bu aygıtı özel bir adres atar, aygıttan konfigürasyon bilgisini okuyarak işleme sokar ve son olarak, aygıt durumunu “configured” yaparak aygıtı kullanıma hazır hale getirir.

USB aygıt durumları aşağıdaki listede belirtildiği gibidir:

- 1) Attached: Aygıt, sisteme takılmıştır.
- 2) Powered: VBUS, spesifik porta uygulanmaktadır.
- 3) Default: Aygıt güçlendirilmiştir ve standart adrese (sıfır) cevap vermektedir.
- 4) Address: Sunucu tarafından aygıtı özel bir adres atanmıştır.
- 5) Configured: Aygıt kullanıma hazırdır.
- 6) Suspended: Aygıt uyku durumundadır.

2.3.ISA Veri Yolu

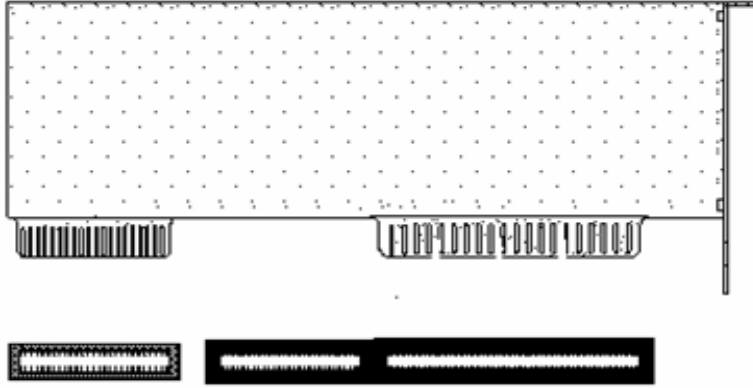


Şekil 2.3. Isa veri yolunun genel görüntüsü

Isa veriyolu ilk temel pc olan i8086 bu yana kullanılan en yaygın ve başarılı tabir edilebilecek kolay kullanımlı geniş amaçlı bir porttur. Kelime açılımı ISA Industry Standard (Endüstri standardı) dır, 8 mhz'lik bir clock hızına sahiptir,fakat düşük hızına ve yapısındaki yavaşlığa rağmen asıl önemini anakart üzerinde olmasından ve direkt olarak PC'nin kalbi olan CPU'ya bağlı olmasıyla kazandığı hız ve esnekliktir. Port temelde 64 bitlik bir pin bağlantı şeması yer almasına karşın sadece 8 adet data ve 16 bitlik bir adres bus'a sahiptir, Isa portunun bir sonraki versiyonunda ise Eisa (Extended ISA) adıyla bilinen 16 bit data ve 24 bit adres taşıyabilen portdur. Şuanki P3 ve P2 bilgisayarlarının anakartları temelde isa portu kullanabilmesine karşın artık gelişen diğer portlar yüzünden ISA portu devreden çıkarılmış ve tarihin kalın yapraklarına gömülmüştür. Yinede bilgisayarın merkezi işlem birimine en çok yaklaşabildiğimiz, kolay ve esnek kullanımlı bu portu incelemek hem PCI portunu hem de temel adres, data bus mantıklarını kavramamızda etkili olur.

Çok rahat ve sorunsuz bir kullanımı olan Isa veri yolu yapısı ne yazık ki günümüz uygulamaları için artık yavaş ve hantal kalmaktadır. Bu yüzden geliştirilen her yeni uygulamada veri yolu minimum PCI olarak tasarlanmıştır fakat USB basit yapılar için tercih sebebidir [1].

2.4.VESA Veri Yolu



Şekil 2.4. Vesa veri yolunun genel görüntüsü

80486 gibi 32 bitlik işlemcilerin yaygınlaşmasından sonra, ISA yolunun bu işlemciler için oldukça yetersi ve yavaş kaldığı görülür. Özellikle grafik uygulamalarında bu gerçek apaçık ortaya çıkmaktadır. Bunu telafi edebilmek amacıyla bazı firmalar, 90'lı yılların başlarında 16 bitlik ISA genişletme yuvaları yanına daha hızlı bir veya iki 32 bitlik yuva daha eklemenin yollarını aradılar. Sonuçta yerel veri yolu (Local Bus) adlı kavram ortaya çıktı. Bu ilk veri yollarının, ilk uygulamalarını özellikle grafik kartlarında görmekteydik. Ancak her firmanın kendi tasarımı ayrı bir çözüm ile ortaya çıkması, farklı üreticilerin kart ve genişletme yuvalarının birbirleri ile uyumsuz olmasına yol açtı. Böylece sadece kart üreten üreticiler her marka için farklı bir kart üretmek gibi bir zorluk içine girdiler.

Uyumsuzluk sorununu çözmek üzere tanınmış ana kart üreticileri birleşerek VESA yerel veri yolu standardını geliştirdiler. Bu veri yoluna “yerel” denmesinin nedeni, mikro işlemci ile genişletme kartı arasına işlemcinin performansını büyük oranda değerlendirebilecek kadar hızlı bir hat çekilmesiydi. Böylece işlemci, genişletme kartı ile olan işlemleri halletmek üzere bu karta hızlı bir veri yoluyla doğrudan transit olarak bağlanıyordu. VESA veri yolu bu nedenle işlemciden bağımsız olarak çalışmamaktadır. Veri yolunun hızı ise, kartın saat hızına göre (25, 33, 49 MHz gibi) farklı değerlerde olabilmektedir. Ancak bu mimari en fazla 2, 3 kartı destekleyebiliyor, eklenen her yeni kart ise veri yolunun performansını düşürüyordu.

Bu nedenle, VESA veri yolu, 486 sınıfı işlemcilerin hızlarını değerlendirebilmek için geliştirilmiş geçici bir çözümdü ve en yaygın olarak grafik kartlarında ve bazen de hard disk kontrol kartlarında kullanım alanı buldu. Geçen sene popüler olan 66 MHz'lik 486 DX2 ve 100 MHz'lik 486 DX4 işlemcili bilgisayarların birçoğunda bu veri yolu, ucuz olması ve bu veri yolunu kullanan bol miktarda kartın ucuza bulunabilmesi nedeniyle yaygınca kullanılmaktadır [1].

2.5. PCI Veri Yolu

PC kullanıcıları için, yüksek hızlarda çalışan 486 veya Pentium gibi bir mikroişlemci kullanan bilgisayardaki, EISA ve MCA yolları büyük bir hayal kırıklığı oluşturdu. Çünkü örneğin, 50 Mhz hızındaki bir bilgisayar, EISA'da 8Mhz, MCA'da 10MHz hızında, sistem uzantı yuvalarındaki kartlarla haberleşebilmekteydi. Bu hız, Windows gibi grafik kullanıcı ara yüzüne sahip işletim sistemleri için çok yavaştı[8].

İntel firması, eski sistem yollarındaki bu temel hız problemini çözmek amacıyla, kendi ürettiği yüksek performansa sahip mikroişlemciler için, PCI (Peripheral Component Interconnect) yol standardını geliştirdi. Aynı zamanlarda, daha önceden geçtiği gibi, VESA grubu da VESA VL yolunu üretti. Bu şekilde, Intel ve VESA birbirlerinden Bağımsız olarak yerel yol standartlarını ortaya koydular. Intel firması, PCI yolunu geliştirmesine rağmen, PC ve sistem yolu üreticilerine, ücretsiz olarak bu yol standardını vermiştir. Bunun sonucu PCI kısa sürede bir endüstri standardı oldu. Bugün için artık VL yolu PC'lerde kullanılmamakta, PCI ve ISA yolları kullanılmaktadır [1].

PCI yolunun en önemli özelliklerinden bazıları, yüksek yol genişliği, mikroişlemci bağımsız olması ve sağladığı bir köprü devresi ile ISA, EISA, veya MCA tabanlı bir kartın PCI yolu ile kullanılabilmesidir.



Şekil 2.5. 5volt PCI slot



Şekil 2.6. 3.3volt PCI slot



Şekil 2.7. 5volt 64 bit PCI slot



Şekil 2.8. 3.3volt 64 bit PCI slot

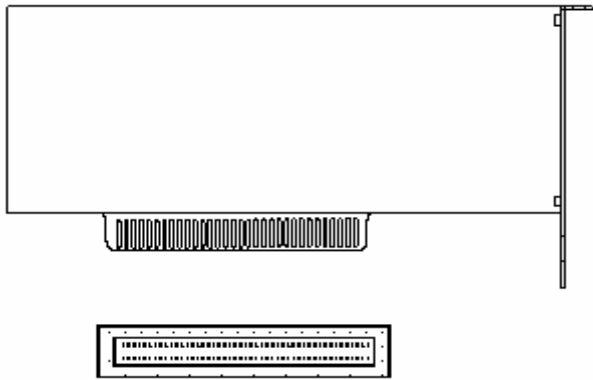
2.5.1. PCI yerel yolunun özellikleri

- 1) Maksimum 33 MHz hıza sahiptir.
- 2) 32-bit ve 64-bit veri yollarına sahiptir.
- 3) PCI yerel yolu işlemciden bağımsızdır. 80x86 ailesi haricinde herhangi bir mikroişlemci ile de kullanılabilir. Bu yüzden, DEC ve Apple gibi bazı firmalar, 80x86 ailesinden bir mikroişlemci kullanmayan kendi ürettikleri yeni bilgisayarlarında PCI desteği sağlamaktadır.
- 4) ISA, EISA ve MCA yolları ile uyumludur. PCI tasarımı, bir yol köprüsü kullanarak, yavaş olan ISA, EISA ve MCA yollarını desteklemektedir.
- 5) Bir PCI yolunda, diğer eski yollardan farklı olarak, ISA yolu için bir yer (konektör) bulunmamaktadır. Yani, bir ana kartta ISA yolu ayrı, PCI yolu ayrıdır.
- 6) 80486 ve Pentium gibi işlemciler tarafından kullanılan, hızlı moda (burst mode) veri aktarımını desteklemektedir.

7) Hem 5V hem de 3.3V iŖe alıŖan kartları desteklemektedir. Bu sayede 5V’lu sistemlerden 3.3V’lu sistemlere yumuŖak bir geiŖ saęlar.

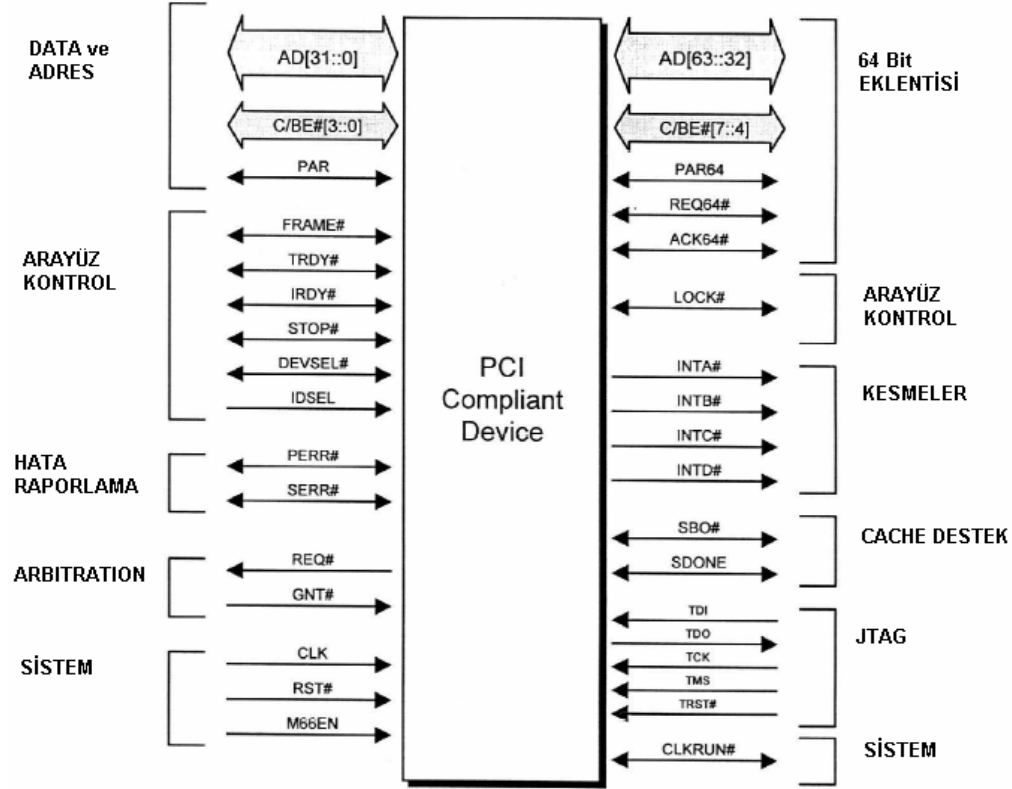
8) Toplam 10 taneye kadar evre birimi destekler. evre birimlerinden bazıları ana karta monte edilmelidir.

9) Kullanıcının sistem yuvasına yeni bir kartı yerleŖtirirken, DIP anahtarlar ile veya dięer Ŗekillerde ayarlar yapmadan, otomatik olarak kendini ayarlama zellięi saęlamaktadır. PCI donanım ayarlama (configuration) yazılımı, atıŖmaları özmek iin otomatik olarak, kullanılmayan adres ve kesmeleri seer.



Ŗekil 2.9. 5volt PCI slot ve kartı

2.5.2. PCI veri yolunun sinyalleri

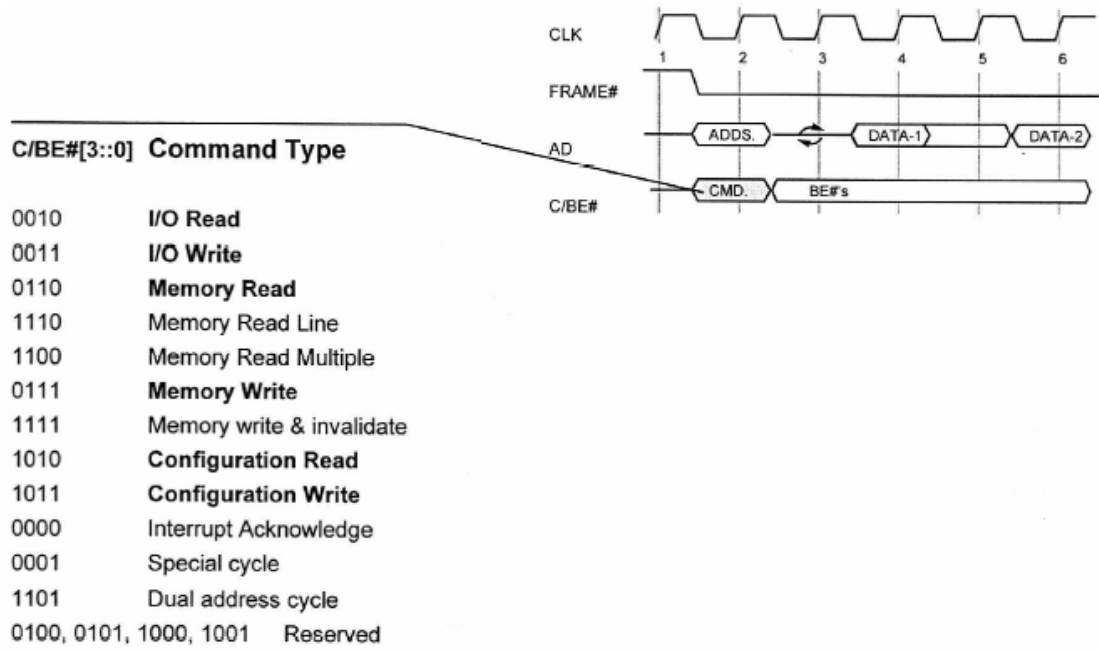


Şekil 2.10 PCI sinyalleri

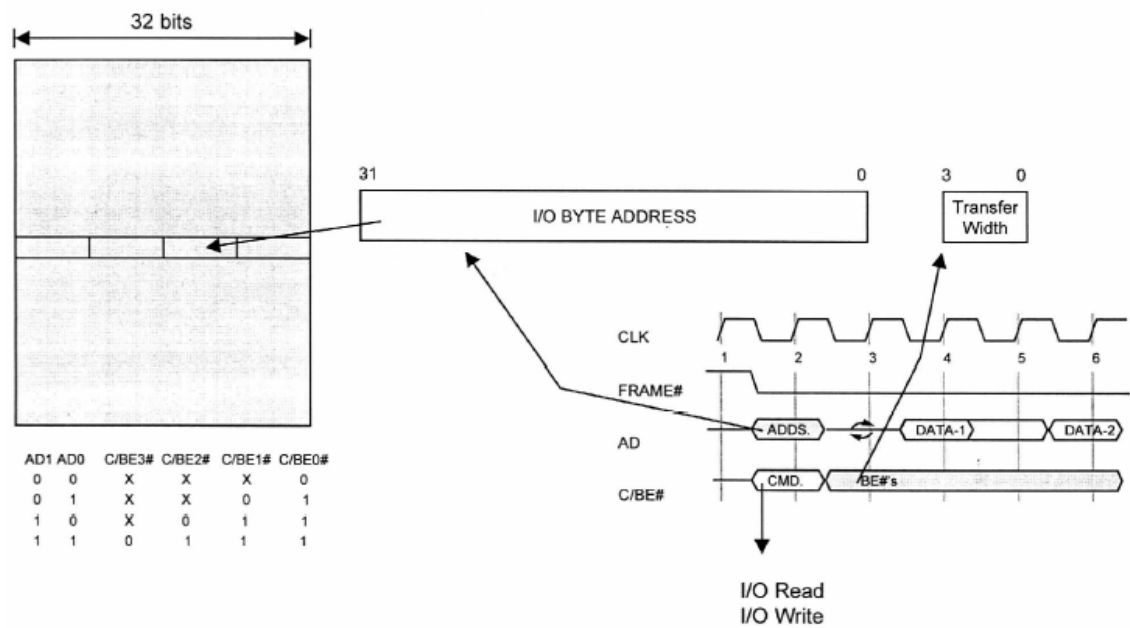
CLK: PCI yolu saat sinyali. Bütün sinyallerin bu sinyale göre çalışır. 0-33Mhz arasında çalışır.

RST: PCI yoluna bağlı bütün kayıtcı ve sinyaller ilk değerlerini almasını sağlar.

AD[31..00]: Veri ve Adres aynı hattı kullanılmaktadır. Adres iletim süresinde; I/O alanına byte, Hafıza(memory) alanına 4 byte (Dword), Ayarlama(config) alanına 4 byte (Dword) olarak erişilir. Veri iletim süresinde; AD[07..00] LSB...AD[31..24] MSB şeklinde çalışır.



Şekil 2.11. C/BE# komut yapısı



Şekil 2.12. C/BE# byte yetki yapısı

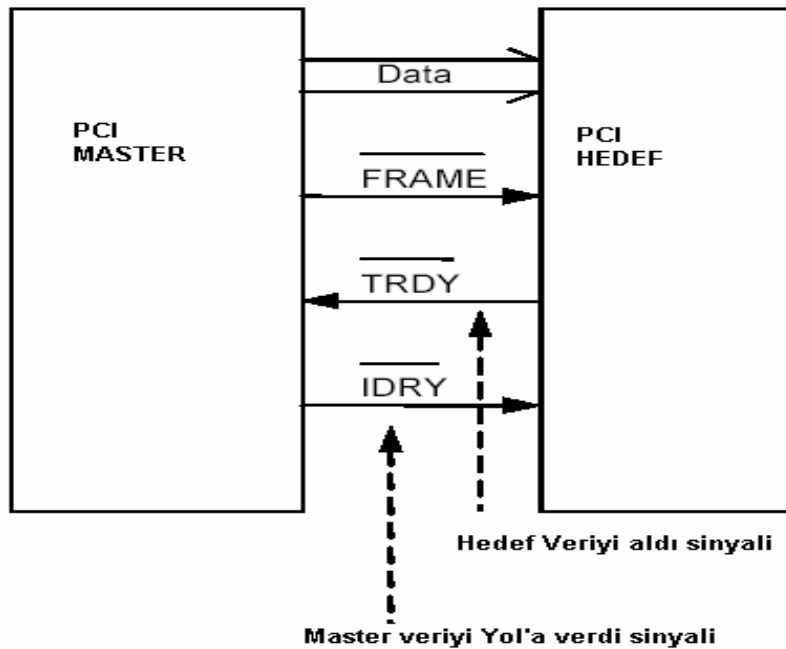
C/BE[0..3]: Komut/Byte yetki sinyalleri. Adres iletim süresinde; Yol komutları, Veri iletim süresinde; Byte yetkisi.

FRAME: Master(Ana)'nın işlem yaptığını bildirir. Adres evresinin başladığını gösterir. Tersinde ise Veri evresini işaret eder.

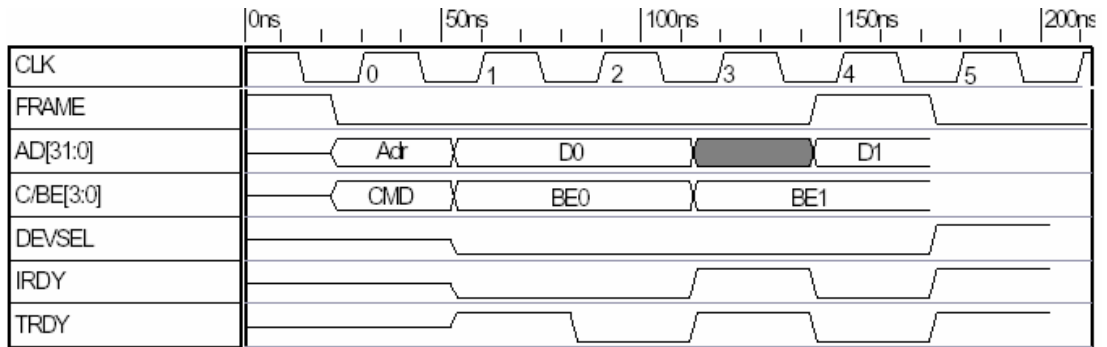
IRDY/TRDY: Master(Initiator)/Hedef(Target) karşılıklı iletişimde, Ayrıca genel bekleme durumunda kullanılır.

DEVSEL: Hedef(Target) kendisini işlem için seçer. (Master'dan sinyal aldığı zaman)

2.5.3. Basit yazma işlemi



Şekil 2.13. Yazma işleminde sinyallerin yönü



Şekil 2.14. Yazma işlemindeki sinyallerin zamanlaması

0. zamanda master ., AD [31:0] daki istenen adres ve C/BE#[3:0] daki istenen komut ile FRAME# i kullanarak işleme başlar (IO okuma, memory okuma, memory okuma çizgisi veya memory read katları).

1. zaman da hedef DEVSEL# low'u göstererek yanıt verir. Bu bir yazma işlemi olduğundan master AD[31:0] ı kullanmaya devam eder(bu sefer data ile), ve bus etrafında dönen çevrime ihtiyaç yoktur. Verinin yazıma hazır olduğunu göstermek için IRDY# düşük konumlandırılır. Hedef TRDY# yi yüksek konumlandığından veriyi kabul etmeye hazır değildir. C/BE#[3:0] tahsis edilen byte'larla yüklenir.

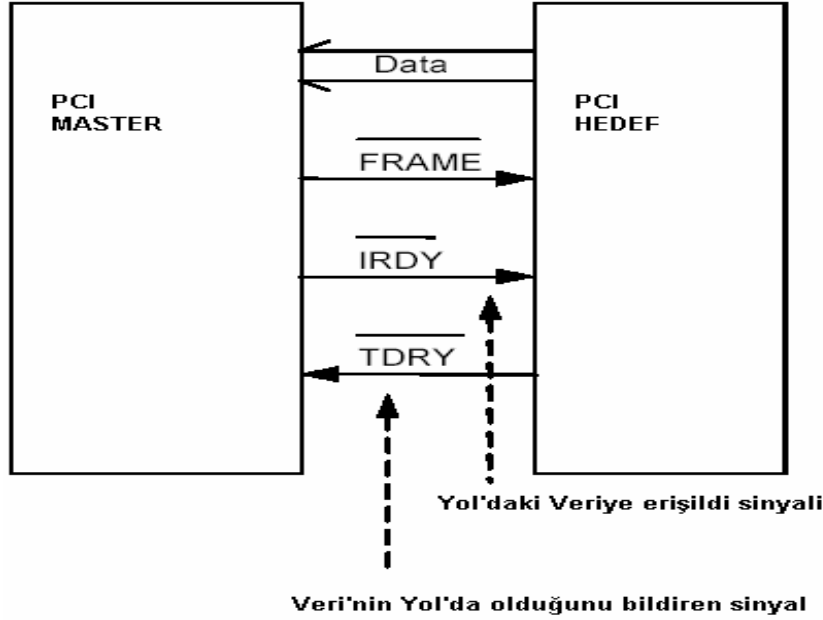
2.zamanda IRDY# ve TRDY# nin ikiside düşüktür, böylece 1. veri kelimesi yazılır.

3.zamanda IRDY# yüksek olduğundan master yeni veri göndermeye hazır değildir(AD[31:0] ın uygun olmayan içeriğinin uygun olmadığı gösterilir).TRDY# düşük olduğundan hedef yeni kelimeyi kabul etmeye hazırdır fakat IRDY# yüksek olduğundan veri transferi olmaz.

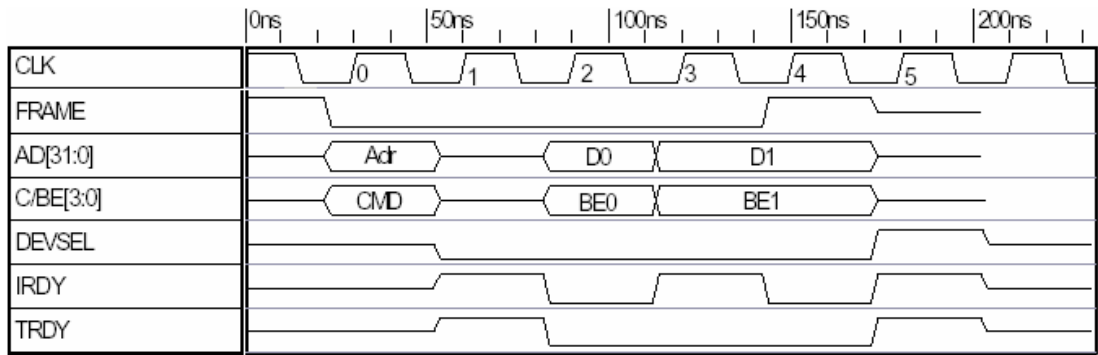
4.zamanda IRDY# düşük olduğundan master veriyi yeniden yazabilir ve veri transfer edilir. Bu son çevrim sırasında FRAME# yüksek olduğundan Master ve hedef 'in ikisi birden çevrimi bitirir.

5.zamanda hedef TRDY# ve DEVSEL# i yüksek konumlandırır. Master da IRDY# ı yüksek konumlandırır. Şimdi yeni bir işleme başlayabilir. (bu çoklu işlemler arasında dönen çevrimler olmadığı zaman arka,arkaya transfer olarak adlandırılır. Back to back transferler yalnızca dönen çevrimlere ihtiyaç olmadığı zaman bir yazma işleminden sonra olur) [3].

2.5.4. Basit okuma işlemi



Şekil 2.15. Okuma işlemindeki sinyallerin yönü



Şekil 2.16. Okuma işlemindeki sinyallerin zamanlaması

0. zamanda master, AD [31:0] daki istenen adres ve C/BE[3:0] daki istenen komut ile FRAME i kullanarak işleme başlar.(IO okuma, memory okuma, memory okuma çizgisi veya memory read katları)

1. zaman da hedef DEVSEL düşük göstererek yanıt verir. 1. zaman bus için read işleminde kullanıldığından beri (master AD[31:0] ' ı kullanmayı bırakır) hedef

AD[31:0] ı kullanmaya başlar, asıl veri transferi 2. çevrimde başlamalıdır, öyle ki hiçbir data henüz müsait olmadığından TRDY ve IRDY yüksek olmalıdır.

2. zamanda IRDY ve TRDY nin ikisi de düşüktür, bu şekilde 1.veri kelimesi okunur.

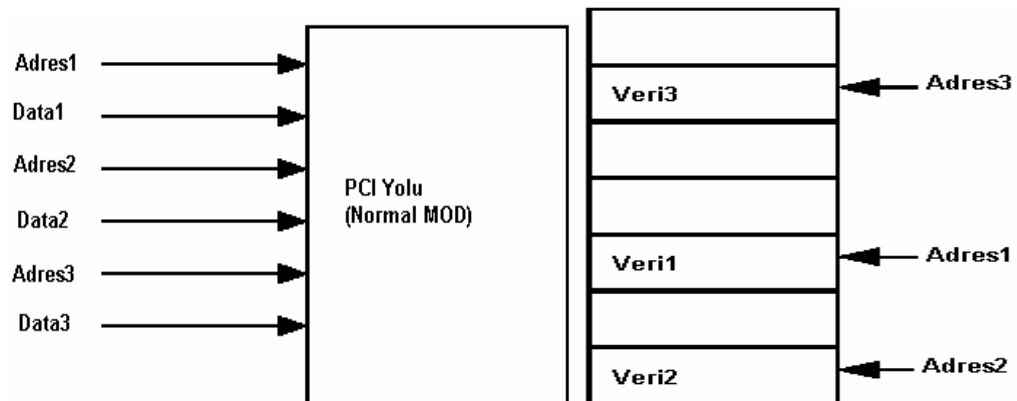
3. zamanda master IRDY yüksek olduğu için yeni kelimeyi kabul için hazır değildir. TRDY düşük olduğu için hedef sonraki kelimeyi transfer etmek için hazırdır. İkiside düşük olmadığından hiçbir veri transfer edilmez.

4. zamanda IRDY# düşük olduğundan master veriyi yeniden alabilir, va 2. veri kelimesi transfer edilir. Bu son çevrim sırasında FRAME yüksek olduğundan Master ve hedef 'in ikisi birden çevrimi bitirir.

5.zamanda hedef AD[31:0] üç durumludur ve TRDY yi yüksek konumlandırır. Master da IRDYyi yüksek konumlandırır. Bu TRDY ve IRDY nin üç durumlu çizgiyi güçlendirdiğinden yapılır ve onları bir çevrim önce üç durumlu yapabilmek için yüksek yapılmalıdır. Bu yüksek bir düzeyde üç durumlu olduktan sonra neden olduğu çizgiye vermek için kullanılır [3].

2.5.5. Transfer modları

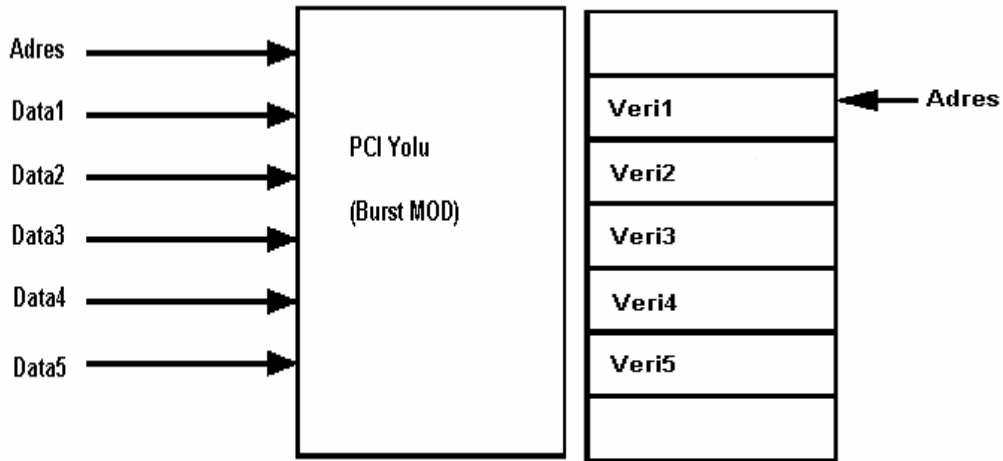
Normal mod



Şekil 2.17. Normal Mod'daki Adres ve Veri akış şekli

Ardışık veri transfer işleminde adres ve veri şeklinde tekrarlayan bir iletişim şeklidir. Adresler değerleri sırasal değil rasgele olduğundan bu işlemde hafızanın veya I/O cihazlarının farklı alanlarına veri gönderilebilir [3].

Burst mod



Şekil 2.18. Burst mod'daki adres ve veri akış şekli

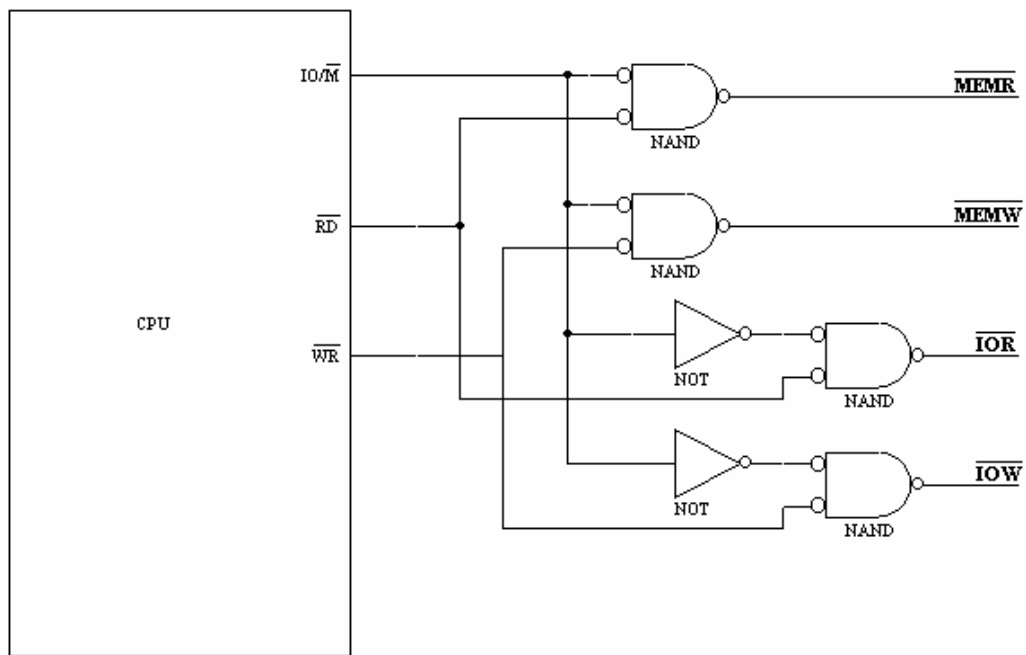
Burst transfer şekli bir adres değeri ve arka arkaya beş veri gönderilerek gerçekleştirilir. Burst transfer çoğunlukla hafıza girişinde uygulanır, yinede, I/O girişinde de kullanılır.

PCI burst transferlerde performans sağlamak için tasarlanmıştır. Bu adres ve verinin daha iyi çoklaması olarak hayal edilebilir. Burst transferde hedef adres veri transferinin başlangıcında yeterli olduğunu belirtir. Bununla PCI sinyal sayısını azaltır ve çalışmanın kararlılığını ve maliyeti azaltır [3].

3.PCI KARTINDA KULLANILAN DEVRE ELEMANLARI

3.1.Temel Giriş/Çıkış Portu

Bu bölüm X86 mikro işlemcisinin temel giriş/çıkış port işlemlerini açıklamaktadır. Ayrıca giriş/çıkış için gerekli kontrol sinyalleri ve giriş/çıkış işlemini gerçekleştirmek için kullanılan komutlar anlatılmaktadır.



Şekil 3.1. RD, WR' dan üretilen IOR ve IOW devresi.

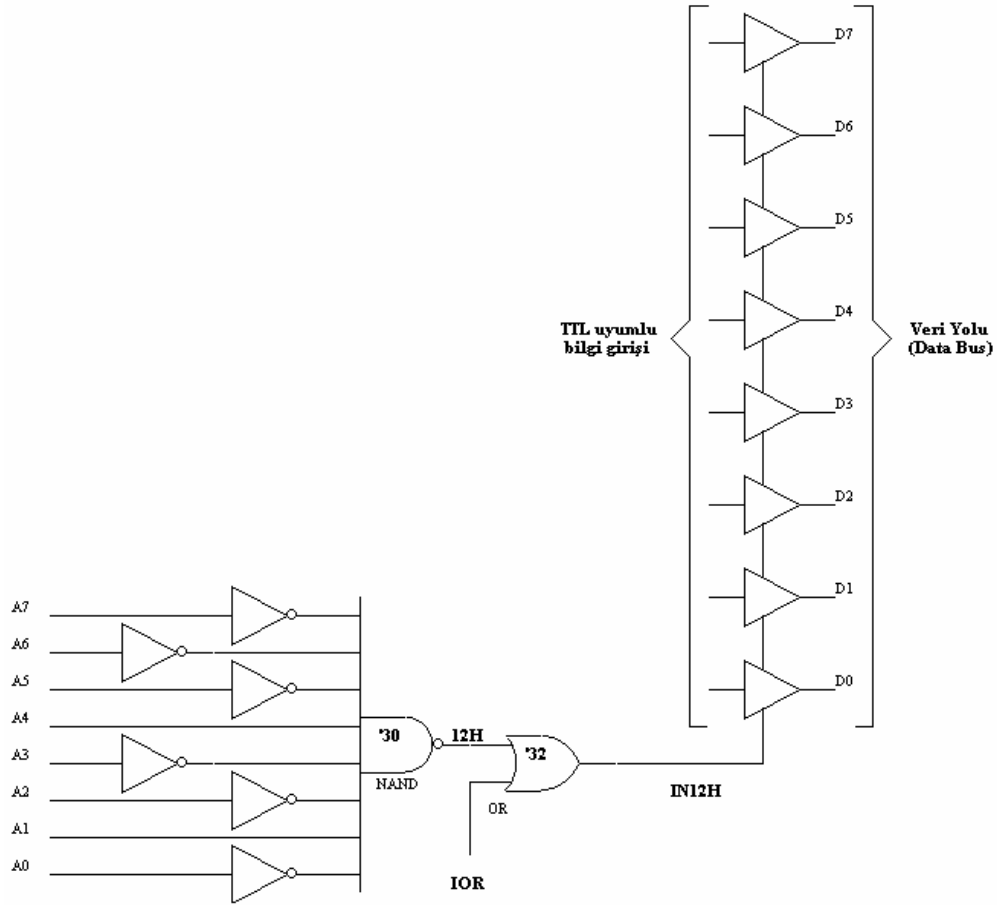
3.2.Temel I/O (Giriş/Çıkış) Komutları ve Kontrol Sinyalleri

I/O işlemleri için 8086 mikro işlemcisi 2 tane komut kullanılmaktadır. IN ve OUT, IN komutu dış I/O biriminden verinin akümülatörü girilmesi ile ilgilidir. Bu veriye ait 8 bitlik adresi adres hatlarında görülür. (A0-A7 uçları) 8086 giriş için 256 farklı adres tanımlayabilir. OUT komutu ise akümülatör içeriğini tamamen adres hatlarında adresi verilmiş (bağlantıda A0-A7) cihaza gönderilir. Yine 8086 çıkış için 256 farklı adresi tanımlayabilir. Her iki komutta akümülatördeki veri sistem veri yoluna transfer edilir.

Giriş yada çıkış kontrol edebilmek için iki sinyal vardır. IOR ve IOW Bu kontrol sinyalleri 8086 tarafından WR ve RD sinyalleri ile IORQ sinyalinin kombinasyonundan üretilir. Şekil 3.1. basit olarak bu sinyallerin üretilmesiyle ilgili devreyi göstermektedir. 8086 mikro işlemcisinin gönderdiği IORQ sinyali ya IN yada OUT komutu ile işletilir. Eğer IN komutu işletilirse RD sinyali ile IORQ sinyalinin kombinasyonu olduğu zaman IOR IN için aktif olurken, WR ve IORQ kombinasyonu olduğu zaman IOW sinyali OUT komutu için aktif olur [4].

3.3.Temel Giriş Portu

Şekil 3.2. 8086 Temel giriş portunu göstermektedir. Buradaki 74LS244 veri girişini veri hatlarına giriş komutu esnasında uygulanmaktadır. Bu devre ile TTL ver mikro işlemciye uygulanır. Eğer TTL olmayan bir veri girişi yapılacaksa, bir analog/dijital çevirici ile verinin şekli değiştirilerek yada başka seviyelere çevirme birimleri ile veri uygunlaştırılarak işlem gerçekleştirilir. Şeklide görüldüğü gibi kodu çözülmüş giriş/çıkış portu 12H , adres yolunda (A0-A7), IN 12H komutu işletildiği sürece görülür. Kodu çözülmüş adres ile IOR sinyalinin kombinasyonundan. Buna göre IN 12H komutu veriyi akümülatöre verir.(gönderir). Böylece TTL uyumlu veri, komutun işlenmesinden sonra akümülatör içinde görülür (bulunur). Temel giriş portu, dış birimlerin yazma ve okuma işlemlerinin anahtarlanması ile ilgili tüm uygulamalarında kullanılmaktadır. Bu devre genellikle giriş birimleri içindir [4].



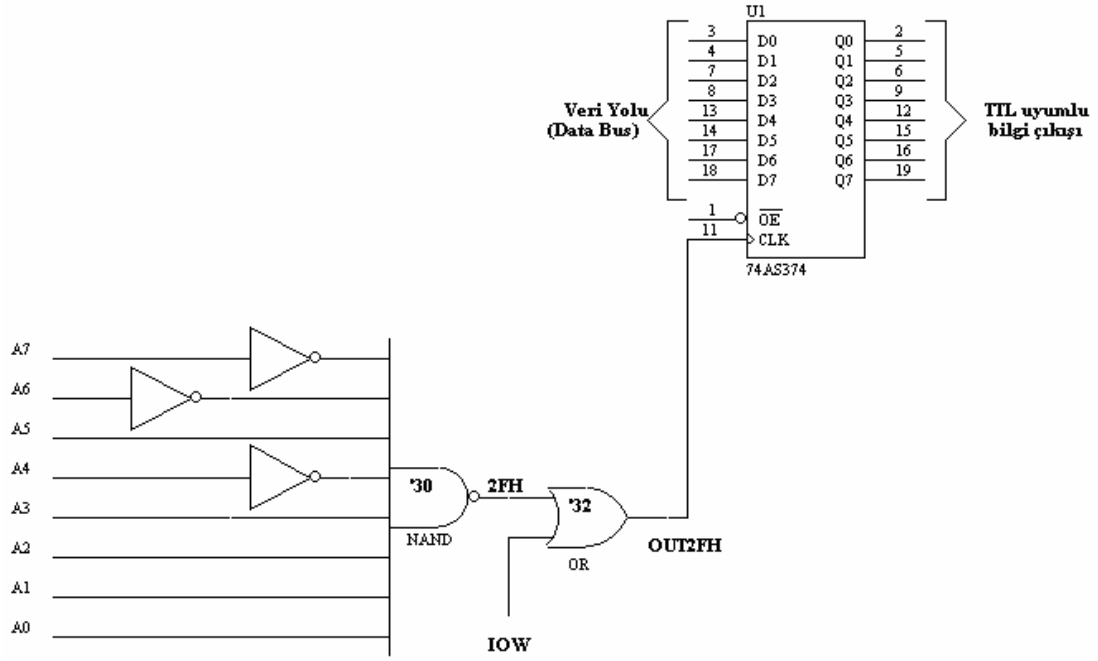
Şekil 3.2. 12H I/O portundan TTL data girişinin yapıldığı basit bir giriş portu

3.4. Temel Çıkış Portu

8086 mikro işlemcisinin temel çıkış portu bazı dış birimlere akümülatörde bulunan TTL verinin gönderilmesinde kullanılır. Tüm durumlarda, çıkış portuna, akümülatördeki veriyi veri yolundan almak çıkış araçlarına göndermeden önce bir süre bekletmek için kilit devresine gerek vardır. Bu devre sayesinde çıkış portu bu bilgileri, çıkış birimleri için bekletir(tutar). Eğer çıkış verisi analog bilgi isteniyorsa dijital/analog çeviriciler kullanılarak bilgi analog hale getirilir.

Şekil 3.3. temel çıkış portunu 2FH'a karşılık olan çıkış portunu göstermektedir. OUT 2FH komutunu işletildiği zaman adres (port numarası) kodu Nand kapıları ile çözülür, Nand kapılarının çıkışı ile IOW sinyalinin kombinasyonu 74LS374 kilidi

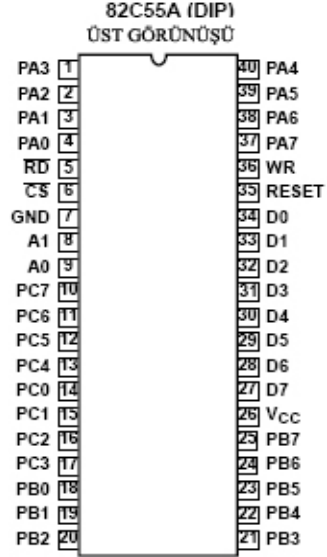
için clock palslerini üretir. OUT 2FH komutunu her icrasında kilit devresi için de clock palsleri üretir. Anlaşıldığı gibi OUT komutu, akümülatör içeriğini işlem sırasında veri yoluna göndermektedir, böylece bu devre akümülatördeki veriyi dış birim (cihaz) için alır ve tutar.



Şekil 3.3. Giriş/Çıkış port numarası 2FH'ta bilgiyi tutan basit bir çıkış portu

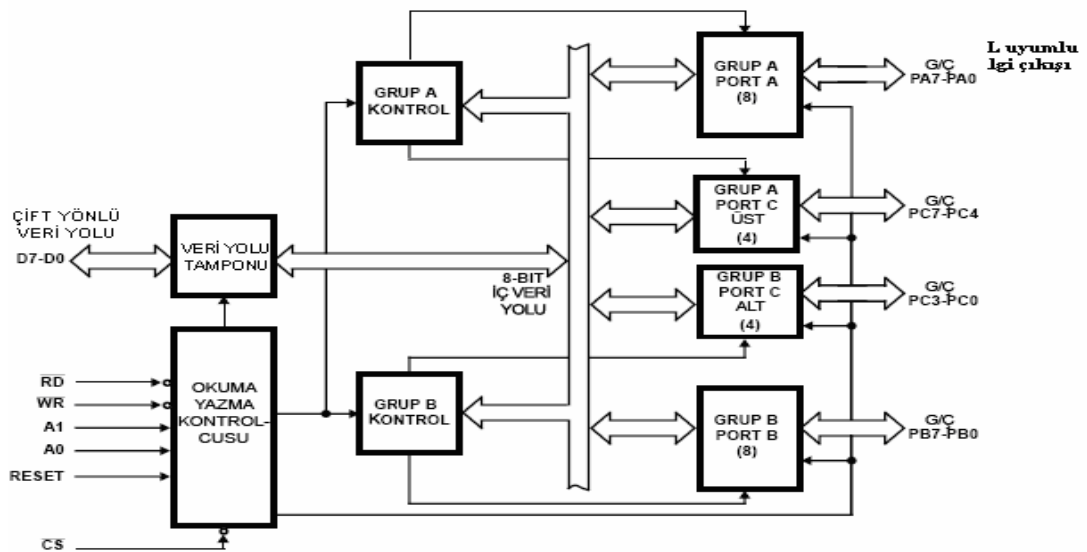
Aslında bütün çıkış birimleri (cihazları) bu kilit devresini içermektedir. Bu kilit devresi belki çıkış portundaki cihazın içinde olabilir belki de ayrı olabilir. Ancak bütün durumlarda bu devre dış birimin içinde bulunmaktadır. Kilit devresi olmayan sistemler de çıkış verisi veri yolunda sadece birkaç mikro saniye içinde görülür ki bunun anlamı da kilitsiz sistemde veri, sadece birkaç cihaz için (hızlı olanlar uyumludur. Çünkü bu hızda veri kabul edebilecek cihaz sayısı çok azdır [4].

3.5. 8255 Giriş Çıkış Entegresi



Şekil 3.4. 8255 entegresinin ayak yapısı

Günümüzde programlanabilen devre elemanları bir çok tasarımda düşük maliyetlerinde dolayı kullanılmaktadırlar. Bunların en popülerleri hiç şüphesiz 8255 PPI'dır. Şekil'de bacak bağlantıları, Şekil 3.4'deyse bacak fonksiyonları gösterilmektedir.



Şekil 3.5. 8255 entegresinin blok şeması

Yongada, 24 bit kapasitesinde giriş/çıkış ve çevre birimleriyle iletişimde kullanılan özel kontrol ve el sıkışma sinyalleri vardır. Intel,8255 yongasını 8085 mikroşlemcilerle birlikte kullanılmak üzere üretmişse de hem 8052 hem de başka mikroşlemcilerle de kullanılmasıyla popülerliğini günümüze kadar korumuştur. Uyumlu yongalar AMD, OKI, Toshiba ve NEC(UPD71055) tarafından da üretilmektedir.

3.5.1. 8255 yapım çeşitleri

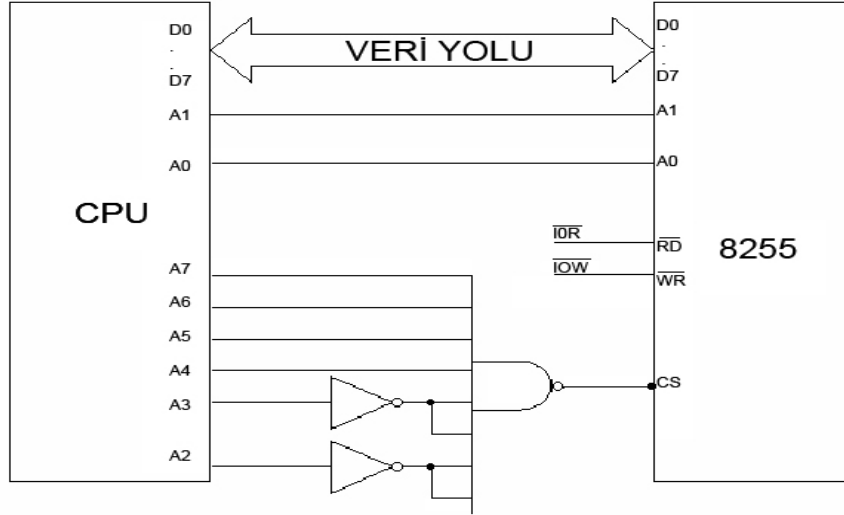
Hem NMOS 8255 hem de CMOS 82C55 kullanabilirsiniz. CMOS sürümü NMOS sürümüne kıyasla bazı avantajlara sahiptir. Daha düşük enerji gereksinmesi (CMOS 10 mA. NMOS sürümde 120 mA) bunun en önemlilerindendir.82C55 CMOS uyumlu çıkışlara sahip olduğundan hem TTL hem de CMOS girişleri rahatlıkla sürebilmektedir. CMOS girişleri sürerken ve eğer NMOS 8255 kullanıyorsanız,8255 çıkışlarının en az 3.5V seviyesinde olmasını garantilemek için yukarı çekme dirençleri kullanmayı ihmal edilmemelidir.

82C55 CMOS sürümü kullanmanın diğer bir avantajı da daha fazla akım verme kapasitesine sahip olmasındadır. Eğer transistör veya birkaç mA gereksinen bir yükü sürmek isterseniz bunun büyük önemi vardır. CMOS 82C55 3V'ta 2.5mA verebilirken, NMOS 8255 2.4V'ta yalnızca 0.5mA verebilmektedir. Her ikisi de 0.45V'ta 2.5mA akım çekebilmektedir [5].

3.5.2. 8255'in çalışma hızı

8255 değişik hızlarda temin edilmektedir.5 MHZ.lik 8255,8255-5 olarak da bilinmektedir.12 MHZ.lik 8086 sistemiyle karşılaştırıldığında 8255 çok yavaş gibi görünse de, 8255 aslen 8085 işlemcileriyle birlikte kullanılma üzere geliştirildiğinden, buradaki -5'le gösterilen hızla 8085'in hızı kastedilmektedir.8086'nin zamanlaması 8085'ten büyük farklılıklar gösterdiğinden 5 MHZ veya daha yüksek hızda bir yongayı 12 MHZ' lik 8086 sisteminde rahatlıkla kullanabilirsiniz [5].

3.5.3. 8255'in çalışma şekli



Şekil 3.6. CPU ,8255 bağlantısı

Şekil 3.6'de 8086 sisteminde 00F0h adresinde erişilen bir 8255 devresi gösterilmektedir. D0-D7 sisteminin veri yoluna, A0 ve A1 en düşük iki adres hattına $\overline{RD}$ ve $\overline{WR}$ girişleri kontrol yolunun $\overline{IOR}$, $\overline{IOW}$ karşılık gelen çıkışlarına bağlanmaktadır. $\overline{CS}$ ise (A2-A7) adres seçme devresine bağlanmaktadır. Bu bağlantı 8255 yongasını 00F0h-00FFh adresleri arasında seçmektedir. 8255 bu portlardan dördünü 00F0h-00F3h adreslerini kullanmaktadır

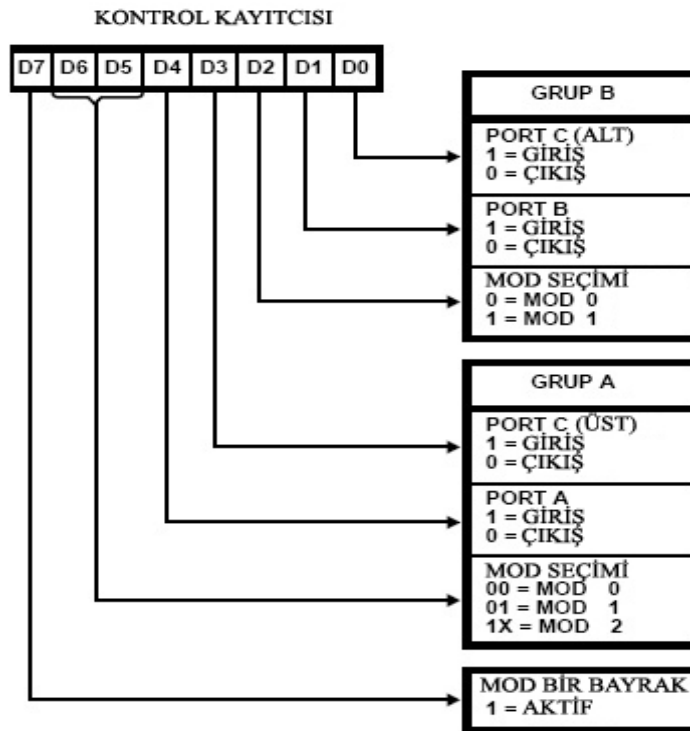
Çizelge 3.1'de görüldüğü gibi PortA, PortB ve PortC üç yeni giriş/çıkış portudur. A0 ve A1 adres hatları erişilecek portu belirlemekte kullanılmaktadır. Buna göre PortA 00F0h'de PortB 00F1h'de be PortC ise 00F2h'dedir. Port bacakları -0.5V ile +6.5V arasındaki gerilimlere bağlanmalıdır.

Çizelge3.1. 8255 adres ve kontrol yolu ile port secimi

A1	A0	RD	WR	CS	GİRİŞ İŞLEMİ (OKUMA)
0	0	0	1	0	Port A → Veri Yolu
0	1	0	1	0	Port B → Veri Yolu
1	0	0	1	0	Port C → Veri Yolu
1	1	0	1	0	Control Word → Veri Yolu
ÇIKIŞ İŞLEMİ (YAZMA)					
0	0	1	0	0	Veri Yolu → Port A
0	1	1	0	0	Veri Yolu → Port B
1	0	1	0	0	Veri Yolu → Port C
1	1	1	0	0	Veri Yolu → Control
ENGELLEME FONKSİYONU					
X	X	X	X	1	Veri Yolu → Üç-Durum
X	X	1	1	0	Veri Yolu → Üç-Durum

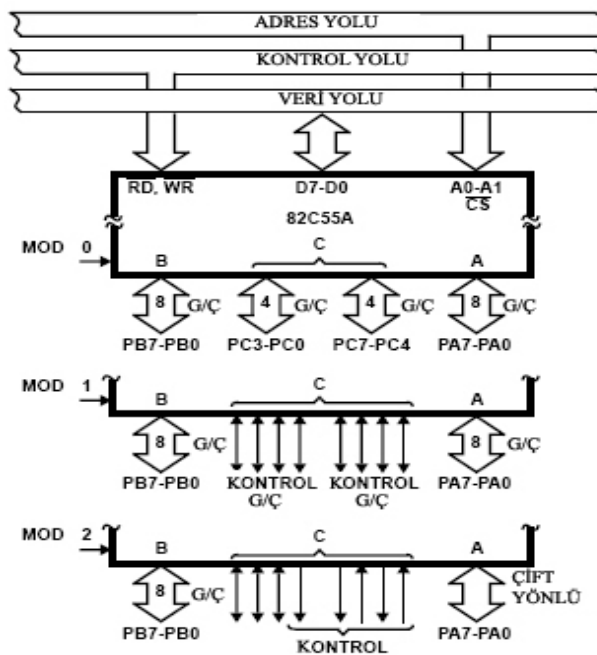
3.5.4. Kontrol kayıtcısı

00F3h adresindeki kontrol kayıtcısına bir kontrol sözcüğü yazarak 8255'i yapılandırabilirsiniz. Kontrol kayıtcısının çalışma modunun seçimi ve port bitlerinin 1/0 olmak üzere iki işlevi vardır. Kontrol yazmacına yazılan bir byte'ın 7.biti 1'se, kontrol kayıtcısı her bir port için çalışma modunu ve portun giriş mi çıkış mı yoksa iki yönlü mü olacağını belirler. Modların ve giriş/çıkışların değişik kombinasyonu olabilir. Kontrol yazmacına yazılan bir byte'ın 7.biti 0'sa, kontrol sözcüğü PortC'nin her bir bitini bağımsız olarak 1'leyebilir ya da 0'layabilir. Kontrol kayıtcısı salt yazılabilir, okunamaz.



Şekil 3.7. 8255 kontrol kayıtcısı

3.5.5. Mod ayarları



Şekil 3.8. 8255 mod secimi

Şekil 3.8’de kontrol kayıtcısıyla mod ayarları gösterilmektedir. Portlar iki gruba bölünmüştür. Grup A, PortA’nın 8 biti ve PortC’nin 4-7.bitlerini, Grup B ise PortB’nin 8 bitiyle birlikte Port C’nin 0-3.bitlerini içermektedir. Bu grupta PortA ve PortB’nin her birinin PortC’nin yarısını el sıkışma veya kontrol sinyalleri için kullanabilmesini sağlamaktadır [5].

Mod ayarı yapabilmek için kontrol kayıtcısının 7.bit mutlakla 1 olmalıdır. 2.bit Grup B için mod 0 veya mod 1’i, 5 ve 6.bitlerse Grup A için mod 0,1 veya 2’yi seçmekte kullanılırlar. 0,1,3 ve 4.bitlerse portu giriş veya çıkış olarak seçmeye yararlar. PortC’nin her yarısı yani alt ve üst 4 biti bağımsız olarak seçilebilmektedir.

```
Mov dx, 00F3h ; dx=00F3h
Mov al, 09Bh ; al=9Bh
Out dx, al ; 00F3 portuna 9B değeri yüklendi.
```

Yukarıdaki komutlarla mod 0’da tüm portları giriş yapmaktadır.
Başka bir şekilde tarif edene kadar 8255 reset durumunda bu moddadır.

Port A’yı okumak için:

```
mov dx, 00F0h ; dx=00F0h
in al, dx ; al’nin içersinde portA’daki değeri oluşmaktadır.
```

Deyimlerini kullanabilirsiniz. PortB veya PortC’yi okumak için yukardaki deyim aynısını sırasıyla 00F1h ve 00F2h adresiyle kullanabilirsiniz.

Tüm portları mod 0’da çıkış olarak tanımlamak için;

```
mov dx, 00F3h ; dx=00F3h
mov al, 080h ; al=80h
out dx, al ; 00F3 portuna 80h değeri yüklendi
```

deyimini kullanabilirsiniz. Daha sonra örneğin PortA’ya yazmak içinse;

```

mov dx, 00F0h ; dx=00F0h
mov al, 015h
out dx, al ; al'nin deęerini portA'ya yazar..

```

Deyimlerini kullanarak 015h deęerinin PortA'ya yazılacaktır. PortB için 00F1h, PortC içinse 00F2h adreslerini kullanın.

3.5.6. Giriş ve çıkış kombinasyonları

Şekil 3.7'de gösterildięi gibi kontrol kayıtçısında 0,1,3 ve 4.bitleri set ederek veya temizleyerek 14 deęişik giriş ve çıkış kombinasyonu oluşturmak olanaklıdır. Mod ayarını deęiştirmek için kontrol kayıtçısında hangi bitlerin 0 veya 1 yapılacağını tespit ettikten sonra karşılık gelen sayıyı ondalık veya onaltılık olarak 00F3h adresine yazılır.

8255'in güzel özelliklerinden biri de çıkış portuna en son yazdığınız deęeri okuyabilmenizdir. Portlar çıkış olarak yapılandırıldığında, portları okumak için kullandığınız 8086 deyimlerinin aynısını kullanarak porta en son yazdığınız deęeri okuyabilirsiniz. PortB ve PortC'de okuduğunuz deęer çıkış mandalında saklanan deęerdir ki bu da porta en son yazılan deęerdir. PortA'daysa durum biraz farklıdır. Çıkış mandallarını okumak yerine PortA'nın bacaklarının o anki mantıksal durumları okunmaktadır. Eğer PortA'daki bir bit topraęa kısa devre edilmişse, o bit için son yazılan deęer 1 olsa da yine de o bit için 0 okunacaktır [5].

3.5.7. Mod 1

Bir çok uygulamada Mod 0 yeterli olmaktadır. Mod 1 ve Mod 2 ise, gereksinen devre elemanlarına el sıkışma veya kontrol sinyalleri sağlamaktadır.

Mod 1 strobe'lu giriş/çıkıştır. 8086 ve çevre birimleri veri gönderirken veya alırken uygun olup olmadıklarını ve gönderilen/alınan bilgiyi birbirlerine teyid etmede el sıkışma sinyali olarak PortC'yi kullanmaktadırlar. Mod 1 giriş bilgisini de


```

mov al,09h      ; al=09h
out dx,al        ;00F3 portuna 09 değeri yüklendi

```

Port B içinse;

```

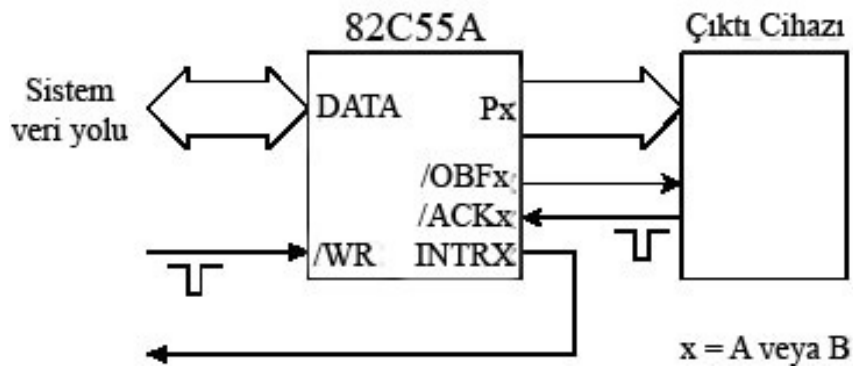
mov dx, 00F3h   ; dx=00F3h
mov al,05h      ; al=05h
out dx,al        ;00F3 portuna 05 değeri yüklendi

```

deyimini kullanın. Bu adreslere bir şey yazdığınızda 8255'in dahili bitlerini set edersiniz. Bu yazma işlemlerinin Port C ile aynı bit adreslerini paylaşan $\overline{STB}$ girişlerine bit etkisi yoktur.

INTR(interrupt request output).INTE set edildiğinde $\overline{STB}$ ile IBF 1 olduğunda 1 olmakta, böylelikle 8255'te okunmayı bekleyen veri olduğu bilgisini vermektedir. INTR, 8086'nın 18.bacağı olan kesme girişine bağlanabilir.Böylece devreye giren kesme rutini 8255'e yeni mandallanan veriyi okuyacaktır.CPU veriyi okuduktan sonra $\overline{RD}$ 'nin alçalan kenarı INTR'yi yeniden 0 yapacaktır [6].

Çıkış kontrolü



Şekil 3.10. 8255 çıkış modu bağlantı sinyalleri

$\overline{OBF}$ (output buffer full output). CPU 8255'e veri yazdığında 0 olur. $\overline{OBF}$, 8255'in okunacak bir bilgi tuttuğunu belirtmek amacı ile çevre birimlerine (örneğin bir gösterge) sinyal göndermek için kullanılabilir.

$\overline{ACK}$ (acknowledge input). 0'a düşerek çevre biriminin veriyi okuduğunu 8255'e bildirir. $\overline{ACK}$ 'nın alçalan kenarında $\overline{OBF}$ 1 olur.

INTE (interrupt enable). INTR'yi devreye almada kullanılır. İşlevi Mod 1'in girişleri gibidir. Grup A Port C'nin 6. bitini, Grup B ise Port C'nin 2. bitini kullanmaktadır.

INTR (interrupt request output). Çevre birimi 8255'e yazılan veriyi kabul ettiğinde 1 olmaktadır. $\overline{WR}$ 'nin alçalan kenarı INTR'yi 0 yapmakta, INTE 1 edildiğinde ve $\overline{ACK}$ ile $\overline{OBF}$ 1 olduğunda yeniden 1 olarak çevre birimin veriyi okuduğunu ve onayladığını bildirmektedir.

Giriş ve çıkışlar için INTR'nin kullanımı opsiyon eldir. Mod 1'de Port C'nin kullanılamayan 2 biti kalır. Port A girişse kullanılmayan 6 ve 7. bitlerdir. Port A çıkışsa 4 ve 5. bitler kullanılmamaktadır. Bu bitleri genel amaç giriş/çıkış olarak kullanabilirsiniz. Mod seçimi kontrol kayıtcısının 3. biti bu iki bit için de giriş veya çıkış işlevinin seçiminde kullanılır [6].

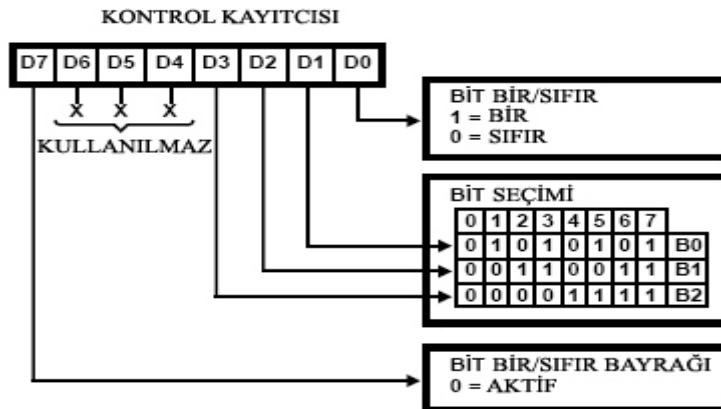
3.5.8. Mod 2

Mod 2 (Strobed Bidirectional Bus I/O). Mod 2, Mod 1'e çok benzemektedir. Ama bu modda veri 2 yönlü akmaktadır. Mod 1'in tüm giriş ve çıkış modlarındaki kontrol sinyallerinin kullanan Mod 2 yalnız Port A için geçerlidir. Port A Mod 2'deyken, Port B mod 0 veya 1'de giriş yada çıkış olabilir. Port C'nin kalan üç biti olan 0-2. bitler kontrol kayıtcısıyla giriş veya çıkış olarak yapılandırılabilir.

Çizelge 3.2. 8255 entegresinin mod seçimleri

	MODE 0		MODE 1		MODE 2
	IN	OUT	IN	OUT	GROUP A ONLY
PA ₀	IN	OUT	IN	OUT	↔
PA ₁	IN	OUT	IN	OUT	↔
PA ₂	IN	OUT	IN	OUT	↔
PA ₃	IN	OUT	IN	OUT	↔
PA ₄	IN	OUT	IN	OUT	↔
PA ₅	IN	OUT	IN	OUT	↔
PA ₆	IN	OUT	IN	OUT	↔
PA ₇	IN	OUT	IN	OUT	↔
PB ₀	IN	OUT	IN	OUT	_____
PB ₁	IN	OUT	IN	OUT	_____
PB ₂	IN	OUT	IN	OUT	_____
PB ₃	IN	OUT	IN	OUT	_____
PB ₄	IN	OUT	IN	OUT	_____
PB ₅	IN	OUT	IN	OUT	_____
PB ₆	IN	OUT	IN	OUT	_____
PB ₇	IN	OUT	IN	OUT	_____
PC ₀	IN	OUT	INTR _B	INTR _B	I/O
PC ₁	IN	OUT	IBF _B	OBF _B	I/O
PC ₂	IN	OUT	STB _B	ACK _B	I/O
PC ₃	IN	OUT	INTR _A	INTR _A	INTR _A
PC ₄	IN	OUT	STB _A	I/O	STB _A
PC ₅	IN	OUT	IBF _A	I/O	IBF _A
PC ₆	IN	OUT	I/O	ACK _A	ACK _A
PC ₇	IN	OUT	I/O	OBF _A	OBF _A

3.5.9.Bit kontrolü



Şekil 3.11. 8255 bit kontrol secimi

8255'in kullanışlı özelliklerinden biride PortC'nin tekil bitlerinin ayrı ayrı 1'lenmesi veya 0'lanabilmesidir. Bunu Şekil 3.10'de gösterildiği gibi yapabiliriz. Bu sefer 7.biti 0 olmalıdır. 0. bit ilgili biti 0 yada 1 e ayarlamakta kullanılmaktadır. 1-3.bitlerse ayarlanacak bitin seçiminde kullanılır. Örneğin PortC'nin 7.bitini 1 yapmak için;

```
mov dx, 00F3h ; dx=00F3h
mov al, 0Fh ; al=0Fh
out dx, al ; 00F3 portuna 0F değeri yüklendi
```

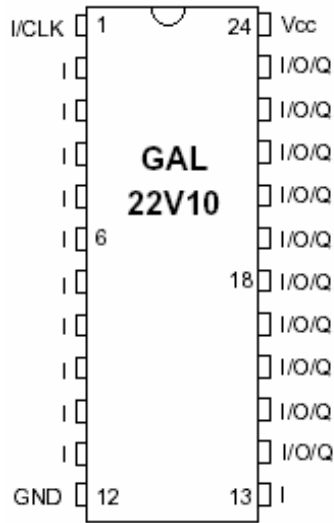
deyimlerini kullana bilirsiniz, temizlemek için yani 0 yapmak içinse

```
mov dx, 00F3h ; dx=00F3h
mov al, 0Eh ; al=0Eh
out dx, al ; 00F3 portuna 0E değeri yüklendi
```

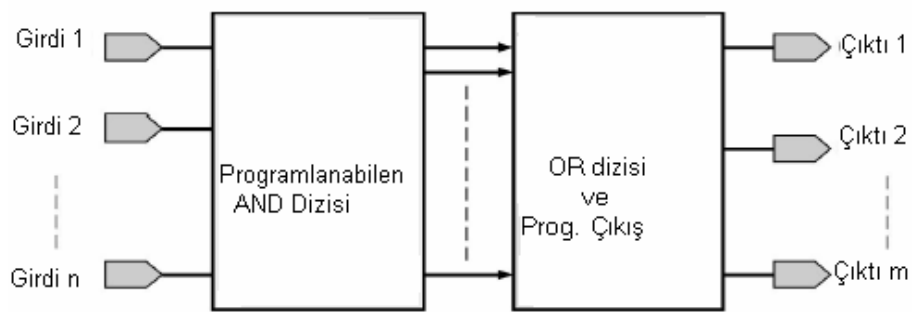
deyimini kullanabilirsiniz. Farklı bir biti 0 yada 1'e ayarlamak için, biti seçin ve karşılık gelen kontrol sözcüğünün ondalık veya onaltılık değerini 00F3h'ye yazın.

3.6. Programlanabilir Kombinezonsal Devreler (GAL)

Programlanabilir kombinezonsal devreler, PLD (Programmable Logic Devices) genel adıyla anılırlar. Bu devrelerde amaç karmaşık lojik işlevleri tek bir veya birkaç devre elemanı kullanarak gerçekleştirmektir; Buradaki programlanabilir olma özelliği, lojik kapılar bağlamında istenilen işlevin sağlanması anlamına gelir. Programlanabilir bir kombinezonsal devre, kendi bünyesinde programlanabilir özellikte VE-VEYA kapıları ve programlama işinin yapılmasını sağlayan sigortalar veya EECMOS devreler barındırır. Programlama işlemi, giriş değişkenleriyle kapı girişlerine bağlantısını sağlayan devrelerin (sigorta ve EECMOS) açık, kapalı bırakılması ile sağlanır. Dolayısıyla, bir değişkenin kapı girişinde etkilenmesi isteniyorsa kapalı, eğer bağlantısı istenmiyorsa açık olur [7].

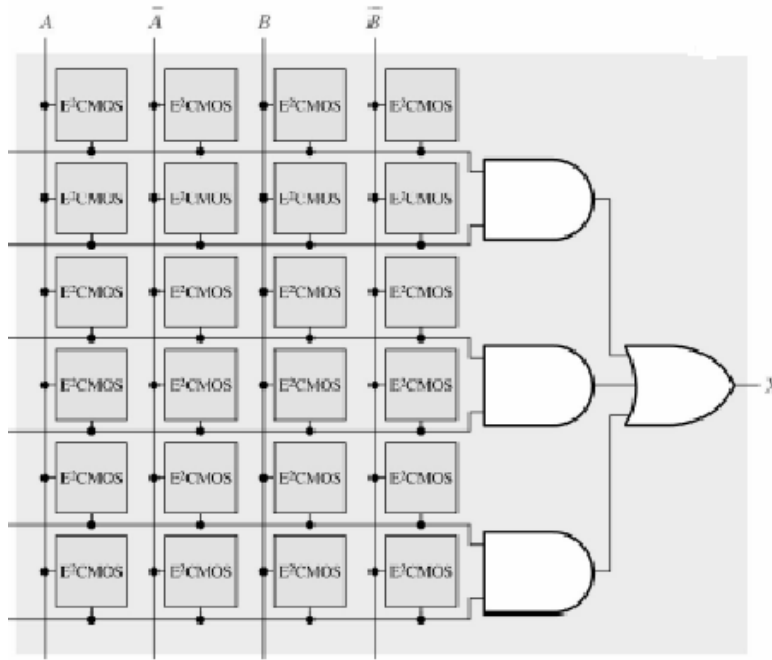


Şekil 3.12. GAL'in pin yapısı



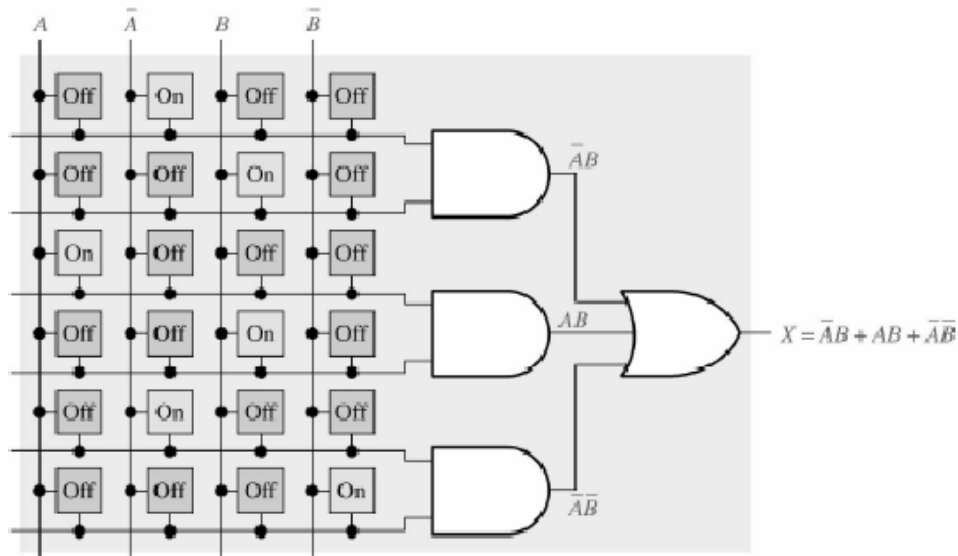
Şekil 3.13. GAL'in iç yapısı

GAL Programlanabilen VE kapı dizisiyle sabit VEYA kapı dizisi şeklinde bir mimariye sahiptir.

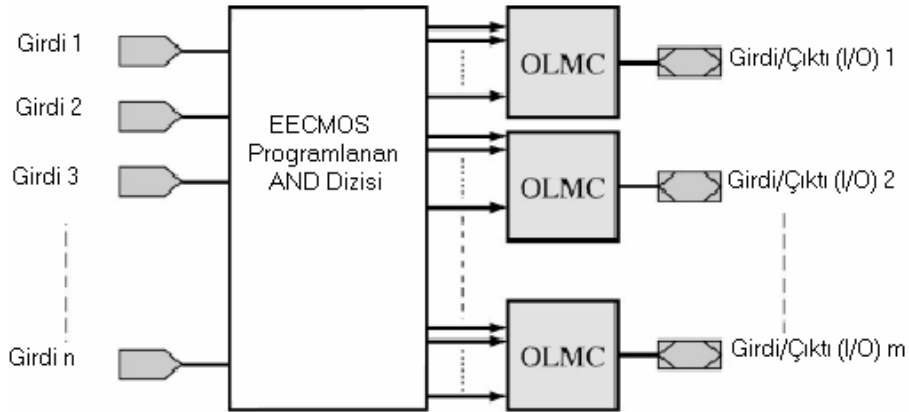


Şekil 3.14. GAL'in programlanabilen hücreler

GAL diğer PLD'de bulunan sigortalar yerine EECMOS devrelere sahiptir böylelikle defalarca tekrar programlanabilmektedir. Programlamadan önce daha önceki programın elektroniksel olarak silinmesi gerekmektedir [7].



Şekil 3.15. GAL'in programlanmış hücre durumu



Şekil 3.16. GAL'in programlanabilen çıkış birimi

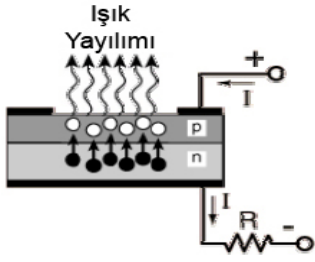
GAL programlanabilen çıkış birimi sayesinde oluşan bir çıktı devre içinde girdi olarak kullanılabilmektedir. Ardışık devre uygulamaları yapılabilmektedir. 12 giriş ucu, 10 giriş/çıkış ucuna sahiptir. Programlanan bir GAL 20 yıl süre ile üzerindeki programı koruyabilmektedir.

3.7. Işıklı Diyot Kullanımı

Dikkatli kullanılmaları halinde ışıklı diyotların ömürleri, ampullerinkinin 100 katına kadar uzayabilir. Fakat bir ışıklı diyoda ampullerinkine benzer şekilde gerilim uygulanırsa (pil,akü yada benzeri bir kaynak üzerinden), o zaman diyot genellikle o kadar çabuk yanar ve kısa bir süre ışık vermesi dahi gözlenmez. Buna rağmen ışıklı diyotların kullanımı, ampullere göre daha zor değildir. Bunun için sadece kendine has özelliklerini bilmek yeterlidir. Aşağıda bu konuya açıklık getirilmektedir [8].

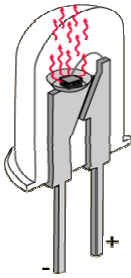
3.7.1. Işıklı diyotların yapısı ve çalışma şekli

İletim yönünde akım geçiren bir diyotta hızlanan elektron ve oyukların nötrleşmesi, bunlara verilen kinetik enerjinin ısıya dönüşmesi sebep olur. Bu enerji diyot a uygulanan hızlandırıcı gerilimle doğru orantılıdır.



Şekil 3.17. Yarı iletken malzemedan ışık yayılımı

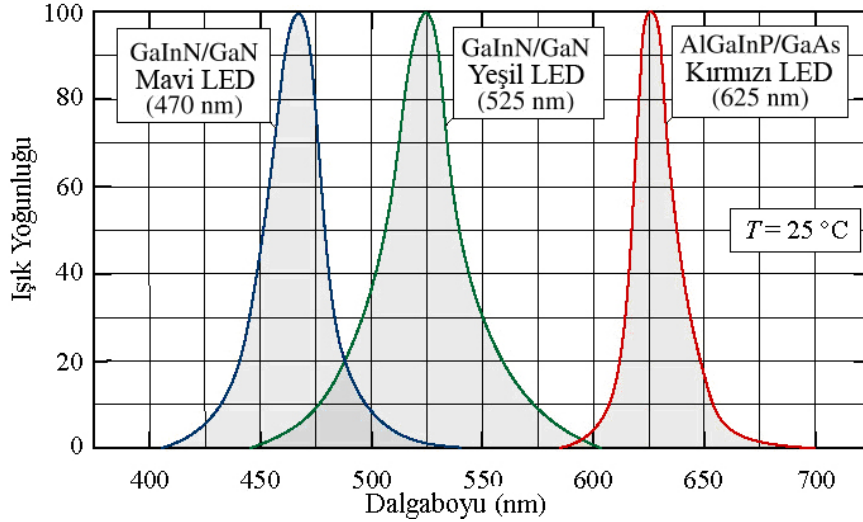
Bu enerji diyot a uygulanan hızlandırıcı gerilimle doğru orantılıdır. Elde edilen ısı radyasyonun dalga boyu bu gerilimin büyüklüğü ile ters orantılıdır. Gerilim arttıkça dalga boyu kısalır. İletimde p-n birleşim yüzey gerilimi yeterince yüksek olan bir diyotun yayacağı radyasyonun dalga boyu ışık olarak gözle görülebilecek kadar kısa olabilmektedir. Bu durumda p-n birleşim yüzeyinde dalga boyuna bağlı belli renkte ışık görülür [8].



Şekil 3.18. LED kesiti

Diyotun dış cephesi şeffaf bir koruyucu içine alındığından ışık kolaylıkla dış ortama yayılabilmektedir.

3.7.2. Işıklı diyotlarda renk



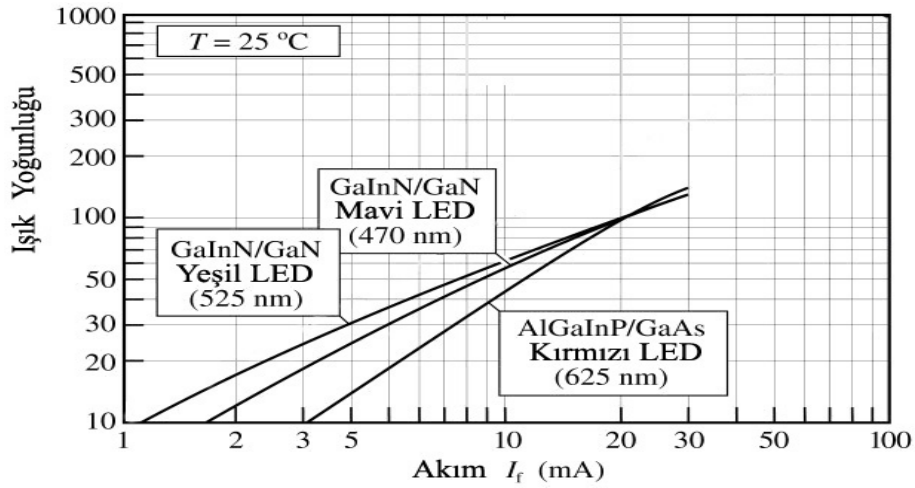
Şekil 3.19. Renklerin dalga boyu

İletim yönünde (Anod pozitif, Katod negatif) polarmalandırıldıkları takdirde ışık yayan bu diyotlara "Light Emitting Diode" kelimelerinin baş harfleri olan L.E.D adı verilir. IRLED'lerin (Infrared Light Emitting Diode) ışınma dalga boyu 700nm den daha büyüktür. Ledlerin ışınma renkleri yapılmış oldukları yarı iletken maddeyle yakından ilgili olup renklere göre yarı iletken madde örnekleri aşağıdaki tablodaki gibidir [8].

Çizelge 3.3. Yarı iletken malzemeye göre renkler

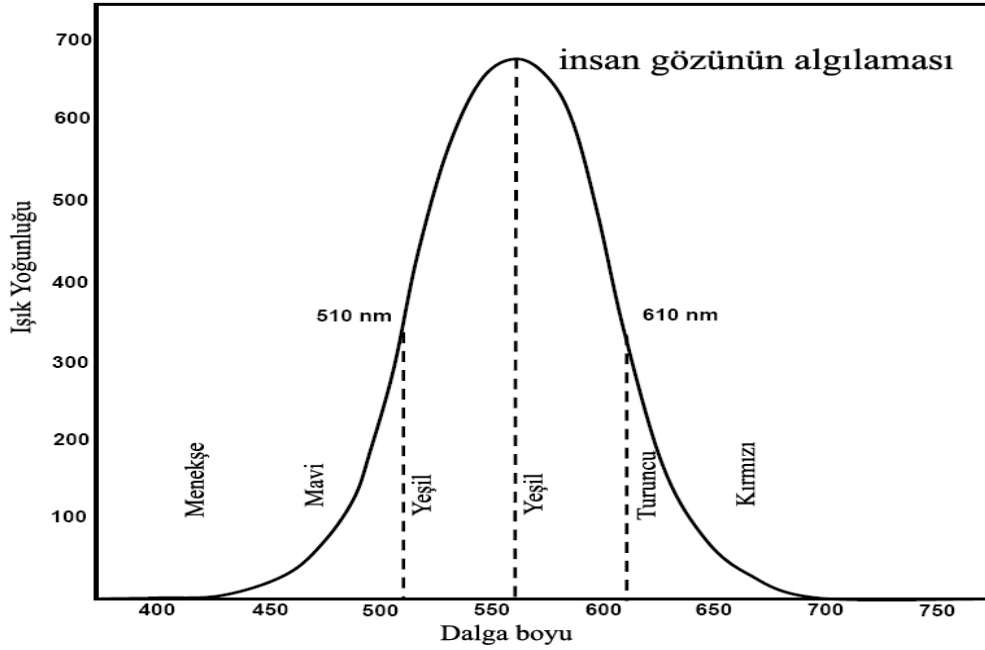
Tipik LED Karakteristiği			
Yarı İletken Eleman	Maksimum Dalga Boyu	Yaklaşık Rengi	VF (20mA)
GaAs	910nm	(Kızılötesi)	1.2V
AlGaInP/GaAs	625nm	Kırmızı	1.8V
GaAsP	580-620nm	Sarı	2.1-2.2V
GaInN/GaN	525nm	Yeşil	3.3V
GaInN/GaN	470nm	Mavi	3.4V
GaInN	460 + 570nm	Beyaz	3.6-4.0V

Led'lerin görünür ışıma bölgelerinde çektikleri akımlar, renklerine göre farklı olup 1,5 mA ile 100 mA arasında çalışabilmektedir. Ancak 40mA'in üstündeki akımlar devamlı çalışmalar doğru akım için kullanılmaz. LED'in en ideal çalışma akımı 20mA olarak seçilir.



Şekil 3.20. Işık yoğunluğu ve ideal akım (20mA)

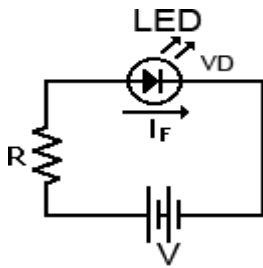
Şekil 3.21'de insan gözünün algılayabildiği dalga boyu sınırları ve algılama hassasiyetini gösteren grafik görülmektedir. Buna göre 400nm ile 700nm arasındaki dalga boyuna sahip ışınlarını görebilmekteyiz. Bu bölgenin üst sınırında kızıl ötesi ışınlar başlar. Kızıl ötesi ışınlar görebilmemiz mümkün değildir.



Şekil 3.21. İnsan gözünün algıladığı renk dalga boyları

3.7.3. Işıklı diyotlarda R direnci hesabı

Şekil 3.22’da LED’in sembolü, de ise 5V’luk bir kaynaktan akım sınırlayıcı bir dirençle çalıştırılma şekli görülmektedir. Led’lerin akımları mutlaka bir dış devre direnciyle sınırlanmalıdır. Bu direncin değeri kullanılan kaynağın gerilim değerine bağlı olup aşağıdaki formülle hesaplanır.



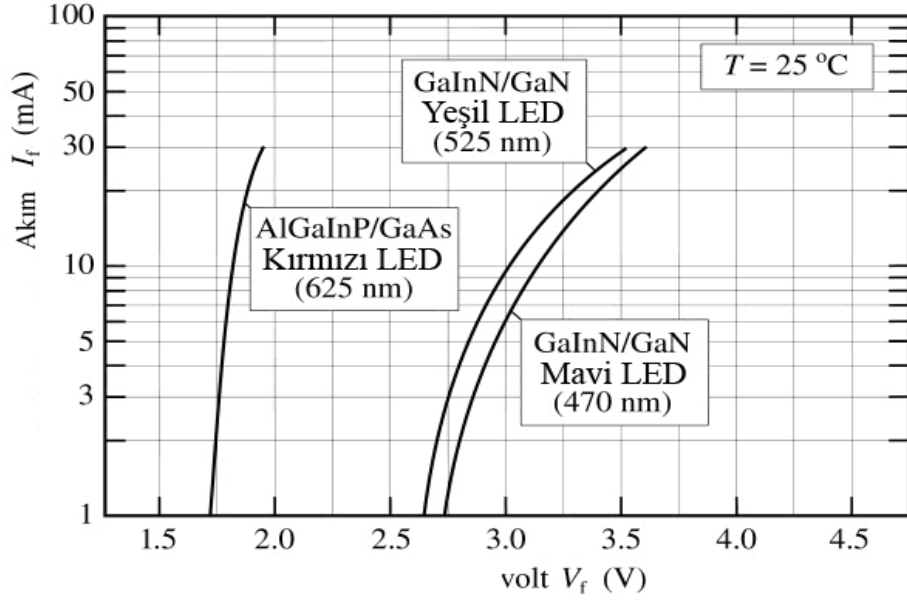
Şekil 3.22. LED devresi

$$R = (V - V_D) / I_F$$

V = Kaynak gerilimi (V)

V_D = Kullanılan LED ‘lerin çalışma gerilimi (1,6 V-2,7V)

IF=LED 'in çalışma akımı (10mA-30mA arasında seçilmelidir)



Şekil 3.23. Akım ve gerilim bağlantısı

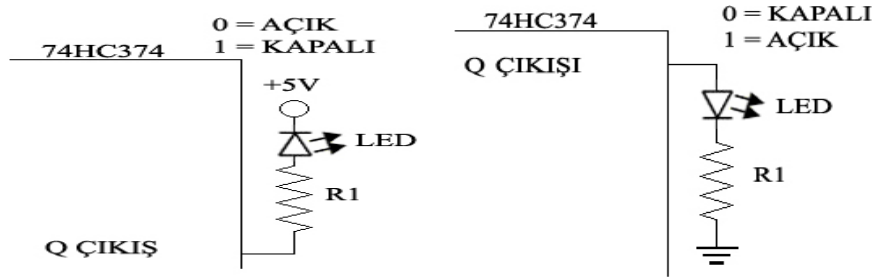
Şekil 3.23.'ye bakıldığında, diyot üzerinden geçen akımla volt arasındaki bağlantıyı göstermektedir. 20mA ideal diyot akımını göz önüne alarak kırmızı led için 1.85V, yeşil led için 3.3V, mavi led için 3.4V olduğu görülmektedir.

Kırmızı Led için R değeri;

$$R = (5V - 1.8V) / 20mA$$

$$R = 157.5 \text{ Ohm}$$

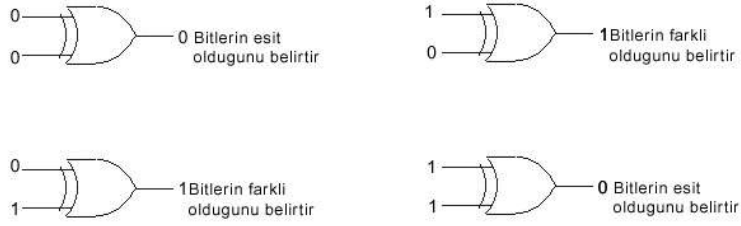
Aşağıdaki şekillerde Led'lerin entegreye iki türlü bağlantı şekli vardır. Akım yönü entegreden toprağa doğru diğerinin akım yönü ise güç devresinden entegreye doğrudur.



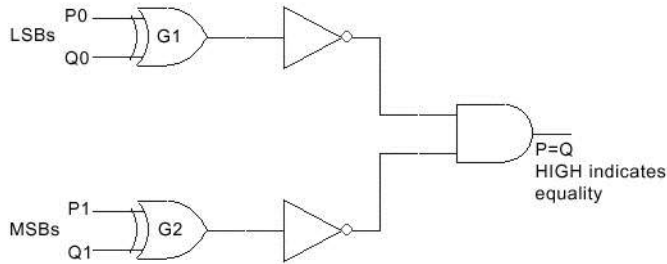
Şekil 3.24. LED'in mandal entegresi(74HC374) ile sürme devresi

3.8. Karşılaştırıcılar

Bir karşılaştırıcının temel fonksiyonu iki niceliğin birbiriyle olan ilişkisini saptamak için bu iki niceliğin büyüklüğünü karşılaştırmaktır. Kendi basit yapısı içerisinde bir karşılaştırıcı iki sayının eşit olup olmadığını saptar. Bir XOR kapısı basit bir karşılaştırıcı olarak kullanılabilir. Çünkü çıkışının 1 olması için girişlerin farklı 0 olması içinde girişlerin eşit olması gerekir. Şekil 3.25.'de bir XOR'un iki bit karşılaştırıcı olarak kullanılması gösterilmektedir. Her biri iki bit içeren iki adet binary sayıyı karşılaştırmak için fazlasından bir XOR kapısı daha gerekir. İki sayının iki en küçük (LSB) değerlikli biti G1 ve iki en büyük (MSB) değerlikli bitde G2 ile karşılaştırılmaktadır. Şekil 3.26. Eğer iki sayı eşitse bu sayıları benzer bitleri aynı ve XOR kapısının çıkışının 0 olması anlamına gelir. Eğer karşılıklı bitler eşit değilse XOR kapısının çıkışında 1 seviyesi meydana gelir. Şekil 3.26.'da gösterildiği üzere iki sayının eşit olup olmadığını tek çıkışta göstermek için iki inverter ve bir AND kapısı kullanılmaktadır. Her bir XOR kapısının çıkışı invert edilip AND kapısının girişine uygulanmaktadır. Her XOR kapısının girişlerindeki bitler eşit olduğunda, rakamların karşılıklı bitleri eşit olur ve her iki XOR kapısının çıkışı inverterler ile 1 seviyesine çekilir. Böylece AND kapısının her iki girişinde 1 sinyali sağlandığı için çıkışta 1 olur. Rakamlar eşit olmadığından en az bir çıkış 0 olacağı için AND kapısının çıkışında 0 olur. Bu şekilde eşitlik=1 ve eşitlik=0 olarak tanımlanır.



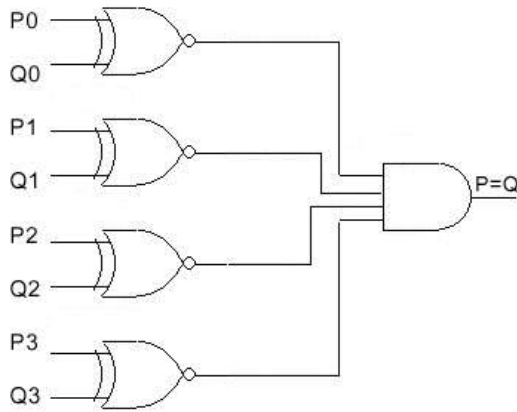
Şekil 3.25. Basit karşılaştırma işlemi



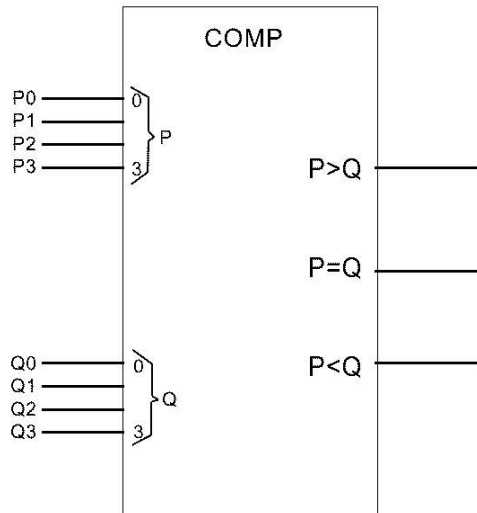
Şekil 3.26. 2 Bitlik binary karşılaştırma işlemi

Temel karşılaştırıcı devre herhangi bir uzunluktaki sayıları karşılaştıracak şekilde genişletilebilir. Şekil 3.27’de iki 4 bitlik sayının karşılaştırılması gösterilmektedir. AND kapısı karşılıklı bitlerin eşit olması şartını kontrol eder. Herhangi karşılıklı iki bitin farklı olması AND kapısının en az bir girişini 0 yapacağından sonuç eşitsizlik=0 olur [9].

Bazı entegre devre karşılaştırıcılar Büyüktür ($P > Q$) ve Küçüktür ($P < Q$) koşullarını kontrol eden fazladan çıkışlara sahiptir. Şekil 3.28’de 4 bitlik bir MSI karşılaştırıcının sembolü gösterilmektedir.



Şekil 3.27. 4 bitlik binary karşılaştırma işlemi



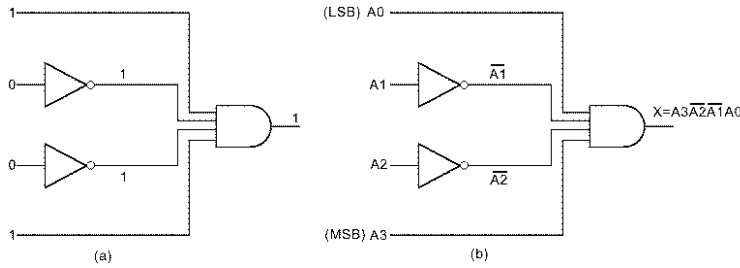
Şekil 3.28. 4 bitlik karşılaştırma yapan entegrenin genel görüntüsü

3.9. Çözümler

Temel fonksiyonu girişindeki bitlerin durumunu tespit etmek ve bu durumu tanımlı olduğu şekilde çıkışlarına yansıtmaktır. Genel olarak decoderler n adet girişi 2^n adet çıkışa tanımlanan şekilde çoğullar [9].

Bir dijital devrenin girişindeki 1001 verisi meydana geldiğinde bunu saptamak istediğimizi varsayalım. Bir AND (ve) kapısı temel bir dekode olarak kullanılmaya uygundur. Çünkü ancak tüm girişleri 1 olunca çıkışı 1 seviyesine almaktadır. Bu

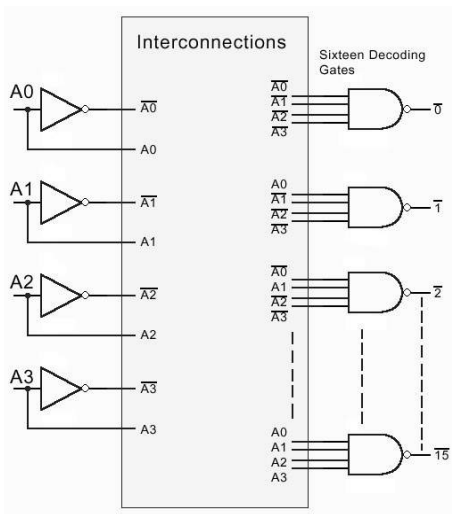
nedenle 1001 değeri için dekoderimizin tüm girişlerinin 1 olduğundan emin olmalıyız. Bura da ortadaki iki biti invert ederek sağlarız.



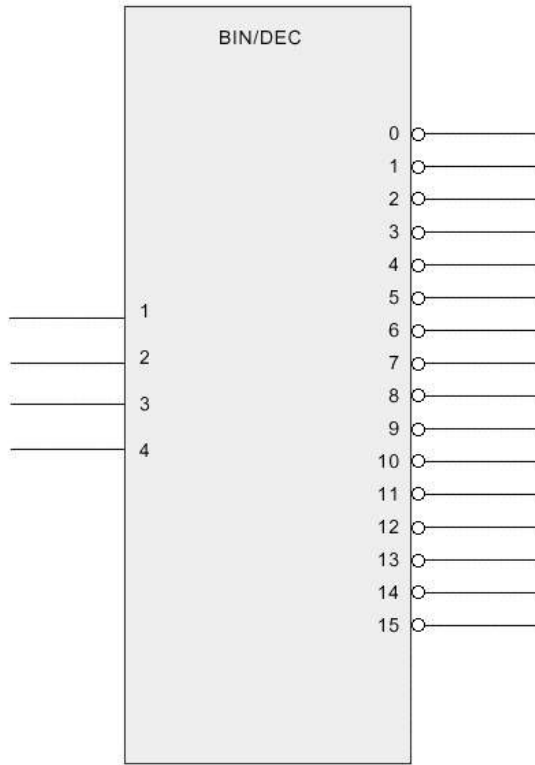
Şekil 3.29. Devrenin lojik denklemi gösterilmektedir.

Aksi belirtilmediği sürece LSB daima yatay sıralamadan en sağdakini ve dikey sıralamada ise en üstteki bittir. Şekil 3.29 de 1001 verisi için Aktif-Sıfır çıkışlı devre şeması gösterilmektedir.

4 Bit Binary Decoder birim tüm olası kombinasyonlarını çözmek gerektiğinde ($2^4=16$) adet çıkışı gerekir. Bu tip çözümler genelde 4-hattan 16-hatta çoğullayıcı olarak bilinir. 8 binary kod verisi ve bunlara karşı gelen çözüm fonksiyonları Tablo 6-4 te verilmektedir. 4 ten 16 ya çoğullayıcı için kullanılan mantık Şekil 3.30 ve mantıksal sembolü Şekil 3.31 aşağıda gösterilmektedir.



Şekil 3.30. 4 -16 decoder giriş ve çıkış şekli

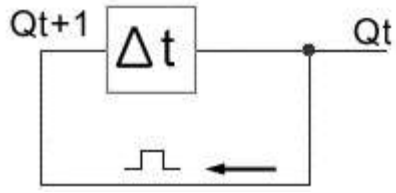


Şekil 3.31. 741574 4-16 hata decoder

74154 TTLMSI decoderlere iyi bir örnektir. Mantıksal diyagramı Şekil 3.31 de gösterilmektedir. Girişlerindeki fazlalık inverterlar aşırı yükleri sürmek ve bu yolla oluşabilecek hasarları gidermek içindir. Burada entegre üzerinde ek olarak bir yetkilendirme fonksiyonu da mevcuttur. Eğer bu yetki girişlerinde 0 durumu oluşmazsa hiçbir çıkış aktif olmayacak ve 0 durumunu korur. BCD' den Decimale çözücülere konumuz dışında olduğundan değinilmemektedir.

3.10. Basit Bir Hafıza Hücresi(Mandal)

Temel hafıza hücresi içerisindeki bilgiyi Δt anındaki $\Delta t+1$ süre sonra tekrar içerisinde barındırabilen bir devre elemanıdır. Bu olay devre içerisinde gerçekleştirilen bir loop ile gerçekleştirilmektedir. Q_t anındaki durum varolan durum olarak kabul edilir ise Q_{t+1} anındaki durum q_{t+1} olarak kabul edilir.

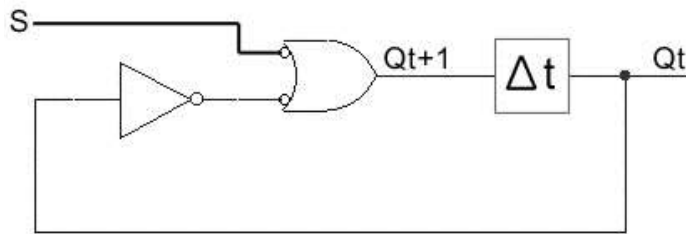


Şekil 3.32. Basit bir hafıza hücresi prensibi

Bir elemanın hafıza hücresi olarak işlevini sürdürebilmesi için bilginin giriş olarak kabul edilecek bir devre elemanı olması gereklidir. Bunu da girişte bir AND kapısı ile gerçekleştirebiliriz.

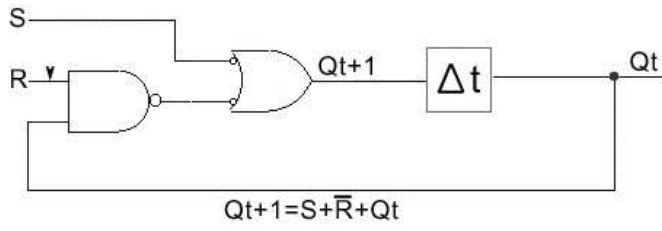
Çizelge 3.4. Hafıza hücresi çıkış değerleri

S	Qt+1
0	Qt hold
1	1 set



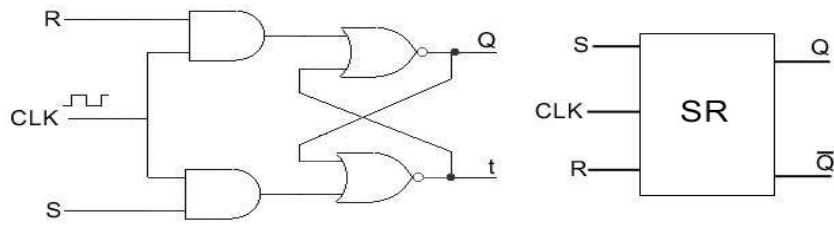
Şekil 3.33. Hafıza hücresinin S tipi flip-flop'la çalışma

Görüldüğü üzere S girişi 0 iken $Qt+1=Qt$ yani varolan durum korunacaktır. S girişi 1 olunca devre Set edilir yani 1 durumuna getirilir. Anlaşılacağı üzere ilk Set edildikten sonra S girişi ne olursa olsun Qt ve $Qt+1$ hep olarak devam edecektir. Devrenin daha doğru olması için girişlerin Set edildikten sonra istenilen bir anda Reset edilmesi gereklidir.



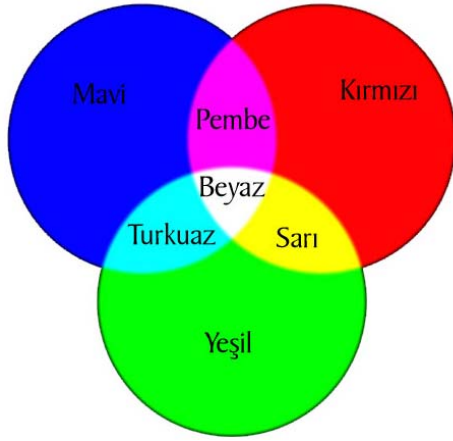
Şekil 3.34. Hafıza hücresinin RS flip – flop tipi

Yukarıda oluşturduğumuz devre Flip Flopların temelini teşkil etmektedir. Aşağıda clock sinyali olan basit bir RS (Reset Set) Flip-Flop blok diyagram ve dizaynı görülmektedir [9].



Şekil 3.35. RS flip - flop

4. IŞIK VE RENK



Şekil 4.1. Işık karışımı

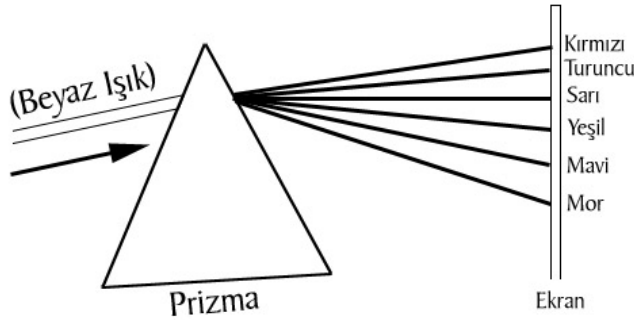
Geçmiş günlerde, henüz çamaşır makinesi yokken, “çamaşır günlerinde” özel bir olay göze çarpardı. Ev hanımı yıkanmış çamaşırları en sonunda içinde mavi renkli çivitli su bulunan bir tekneye sokardı. Bu banyo işleminden çıkan çamaşırlar, günümüzün yıkama malzemesi imalatçılarının iddia ettikleri kadar parlak ve beyaz olurdu. Daha yakından yapılan bir inceleme, çamaşırların bu işleminden önce az bir miktar sarıya çaldıklarını, bunun da büyük bir olasılıkla o zamanlar çamaşır yıkama için kullanılan yumuşak sabundan kaynaklandığını ortaya koymuştur. Sarı ve çivit mavisi bir araya gelince saf bir beyaz renk oluşturmaktaydılar. Benzer bir olay, Şekil 4.1’da sol tarafta görülen renkli iki ışık demeti ile de gerçekleştirilebilir. Her iki ışık beneğinin kesiştikleri bölge, daha fazla ışık düştüğünden daha aydınlıktır. Ayrıca bu bölge renkli değil, beyazdır. Bu yol izlenerek, başka başka renklere sahip ışık demetleri ile farklı renk izlenimleri oluşturulabilir; üç veya daha fazla sayıda renkli ışık demetinin karıştırılması halinde de beyaz elde edilebilir. Dolayısıyla, iki önemli sonuç ortaya çıkmaktadır [10].

Bir renklenme, başka renkli ışık demetlerinin uygun şekilde bir araya getirilmeleriyle kompanze edilebilmektedir.

Renkli ışık demetleri karıştırılarak başka renkler veya beyaz renk elde edilebilmektedir.

4.1. Renk ve Işık İki Ayrı Şeydir

Cisimleri üzerindeki renk ışık değildir. Görülebilen ışık, dalga boyu 780 nm ile 390 nm arasında olan elektromagnetik dalgalardan oluşur. Parlaklığı sağlayan bu madde, günümüzde de yıkama malzemeleri kapsamında yer almaktadır; mavi tonlanma mor ötesi bölgesine kaymıştır, ancak etkisi aynı olmaktadır.



Şekil 4.2. Beyaz ışık

4.2. Beyaz Işık

Beyaz ışık, renkli ışığın bir karışımıdır. Bu durum, güneş ışınlarının Şekil 4.2'deki gibi bir prizma ile ayrılmaları halinde kolayca görülebilir. Renkli ışık demetlerinin karıştırılmasıyla yeniden beyaz ışık oluşturulabilir Şekil 4.1. Ancak, boya kutularındaki “renklerle” bu iş yapılamaz. Böyle bir durumda çok sayıda rengin karıştırılması halinde hiçbir şekilde beyaz renk elde edilemez, bunun yerine kirli bir gri kahverengi ton oluşur. Işık kaynaklarının tersine, bu cisim renkleri adeta birer seçici reflektör görevi yaparlar. Bir renk izlenimi, sadece ışık ışınlarının gönderilmesi halinde elde edilebilir (geceleri bütün kediler gri olarak gözükürler). Işık demetinde cisim üzerindeki renk tarafından iyi bir şekilde yansıtılan bir dalga boyunun bulunması halinde, bir renk izlenimi oluşur. Bunun dışındaki bütün diğer ışık dalgaları zayıflatılır veya yutulur.

Televizyon kamerasına renkler değil, ışık ulaşmaktadır. Bu ışık elektriksel işaretlere dönüştürülmeli ve birçok biçim değiştirmeden sonra ekran üzerinde yeniden ışığa çevrilmektedir. Ekran renkli ışık verir, ancak kendi rengi yoktur. Cihazın kapalı olması halinde, ekran koyu gri olarak gözükür. Toplam transmisyon yolu üzerinde de her hangi bir yerde renkler bulunmamaktadır. Renkli televizyon üzerine literatürde verilen renkli resimlerle bu kitapta karşılaşılan resimler, sadece yetersiz karşılaştırmalardır [10].

4.3. Renk İşaretleri

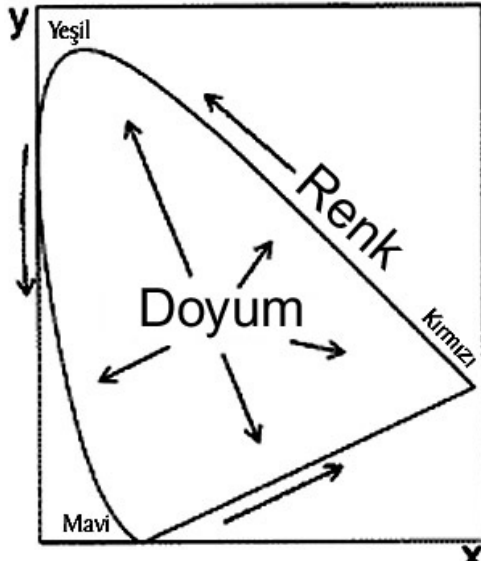
Televizyonda renkli resmin oluşması, çocuklar için hazırlanan boyama kitaplarının prensibine benzemektedir. Söz konusu kitaplarda önceden basılmış siyah-beyaz bir resim bulunmaktadır. Çocuk yüzeyleri resmine göre renkli kalemle boyarsa, resim elde edilir. Buradan şu sonuçlar çıkartılabilir:

Siyah-beyaz resim bütün resim ayrıntılarını içermelidir; diğer bir deyişle bu işaret büyük bir band genişliğine sahip olmalıdır.

Renk işaretleri daha kaba, yani daha dar bandlı olabilirler. Buna rağmen, bütün işaretlerin bir araya getirilmeleri halinde, güzel bir resim elde edilmektedir.

Renkli televizyonda üç renk işareti, yani mavi, kırmızı ve yeşil renklere ilişkin üç ayrı işaret kullanılmaktadır. Diğer bütün renkler bu üç temel rengin karıştırılmasıyla oluşturulur. Tersine, renkli ışık da bu üç temel renk bileşenine ayrılabilir. Renkli televizyon için temel renkler olarak şunlar seçilmektedir [10].

4.4. Renkler ve Bunların Kademelendirilmesi



Şekil 4.3. Renk türü ve krominans

Bir renk koordinat sistemi bir tür üçgenden oluşmaktadır Şekil 4.3. Üçgenin her bir köşesi, mavi, yeşil ve kırmızı renklerinden birine karşı düşürülmektedir. Üçgenin kenarları boyunca ilerlenirse, bir renkten yavaşça diğerine geçilir. Yani, gökkuşağında olduğu gibi, yeşil ve kırmızı arasında renk dizisi yeşilden sarıya geçer ve bunu kırmızıya geçilmeden önce turuncu renk izler. Alışlagelen renkler olarak isimlendirilen bu renkler, renk körü olmayan herkese doğru olarak hitap ederler. Renk körü olmayan bir kimse, çekinmeksizin turuncu ve kırmızı renkleri ayırt edilecektir. Renkler yerine daha çok renk türü demeye alıştırmalıdır.

Üçgenin ortasına doğru gidildikçe renkler solgunlaşır. Merkezinde ise beyaz noktası bulunmaktadır. Bu arada, R (kırmızı), G (yeşil) ve B (mavi) renklerinden her birinin resim tüpünün üç ayrı fosforlu maddesinden birine karşı düştüğünü belirtmekte yarar vardır. Bu üç ışık kaynağının yaydığı renk ışınlarının karşılaştırılmasıyla, siyah üçgenin içinde bulunan bütün renk çeşitleri oluşturulabilir.

Üçgen biçimli renk koordinatları şebekesi, renkli televizyonda bir daireye dönüştürülmektedir. Şekil 4.3'deki R-G-B üçgeni, üç çivi arasına gerilmiş ve renkli tonlara sahip bir ipten oluşuyor gibi düşünülür. Çiviler yerlerinden çıkartılır ve ip bir daire oluşturacak biçimde yayılırsa, renk tonları Şekil 4.3'daki gibi bir α açısı ile verilebilir. Buna göre, $\alpha = 100^\circ$ açısı kırmızı bir renk tonuna, $\alpha = 240^\circ$ ise yaklaşık olarak yeşil bir renk tonuna karşı düşecekti. Erguvan ve gök mavisi renk tonları, renk literatüründe Açık Kahverengi ve Turkuaz olarak da isimlendirilmektedir. Renk dairesinde karşı karşıya gelen renk çiftleri, yani sarı ve mavi veya kırmızı ve gök mavisi, eşlenik renkler olarak adlandırılır [10].

4.5. Renk Doyması

Aynı renk tonu ve canlı olarak gözükebileceği gibi yumuşak ve pastel biçimi de olabilir. Kırmızı mürekkep dolu bir şişe, sert kırmızı bir renk tonuna karşı düşer. Mürekkebin içine su katılırsa, renk tonu değişmez. Ne mor ne de sarımtırak olur; ancak, ne kadar fazla su katılırsa, renk tonu da o derecede donuklaşır. Aynı renk tonunun bu şekilde kademelere ayrılmasına o renk tonunun doyması veya doyurulmaması adı verilir. Şekil 4.3'deki renk dairesinde çevre doymuş renkleri göstermekte ve merkeze doğru gidildikçe renkler soluklaşmaktadır. Doyurulmamış olduğundan, dairenin ortası beyazdır.

4.6. Parlaklık, Parlaklık Yoğunluğu

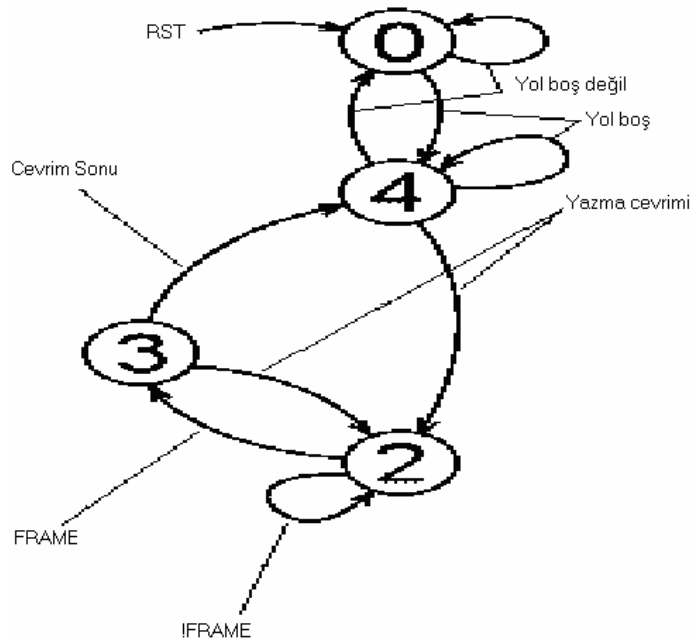
Renk doyması renk parlaklığına eşdeğer tutma eğilimi oldukça yaygındır. Ancak bu doğru değildir. Parlaklık kavramı aşağıdaki örnek yardımıyla daha iyi açıklanabilir: Bir fotoğraf makinesi objektifinin diyaframı küçültülürse, resim düzleminin parlaklığı azalır. Bu olay bir aynalı refleks kamera yardımıyla kolayca izlenebilir. Ancak bu parlaklık değişimi, hiçbir şekilde renk tonuna ve renk doymasını değiştirmez. Kırmızı renk kırmızı, benzer şekilde soluk mavi bir renk soluk mavi ve beyaz renk de beyaz olarak kalır. Uygun şartların sağlanması halinde, resmin küçük veya büyük diyafram kullanılarak çekilmiş olup olmadığına bakılmaksızın, renkli fotoğraflarda daima aynı renkler elde edilir.

5. PROJENİN GERÇEKLEŞTİRİLMESİ

5.1. GAL'ın Programlaması

PCI tam bir senkronize yol olduğu için bütün işlemler sistem saatine göre yapılır. Yapılmak istenilen kartın PCI olması birden fazla sinyalin girdi olarak kullanılmasından dolayı sadece PLD ile gerçekleştirilir(GAL 22V10).

5.1.1. Durum diyagramı



Şekil 5.1. PCI yazma sinyallerine göre oluşturulan diyagram

Durum 0: Yol boş değildir. Herhangi bir diğer işlem uygulanmaktadır. Yol boş hale gelinceye kadar bekler.

Durum 4: Yol boş ise: Bu durumda yeni yol evresinin başlangıcı bulunur. Bu hedef alete bir I/O cihazı yazmak için aktif edildiğinde durum 2'e taşır. Yol evresi hedef aletle bir ilişki kuramadığı zaman tekrar durum 0'a geçilir.

Durum 2: Yeni bir I/O yazma evresi çalıştırılır. Bu alet DEVSEL i bir yapar ve Yol'a cevap verir. Aynı zaman da hedef alet TRDY yi bir yapar ve yazılması için veriyi kabul edebileceğini gösterir. Bu anda veri'yi kilitler ve IRDY bir yapıldığı zaman ve FRAME sıfır olduğu zaman durum 3 e geçirir.

Durum 3: DEVSEL ve TRDY sıfır yapılır. Boş durumuna geçirir(durum 0). Fakat boş duruma geçmeden I/O cihaza yazma işlemi tekrarlanıyorsa direk olarak durum 2'e geçirir.

Durum 1: Durum numarası tanımlanmamaktadır. Eğer uygulamada bir hata oluşacağı düşünüldükçe konulmaktadır [10].

5.1.2.Gal'in abel dili ile programlanması

Yukarda anlatmış olduğumuz diyagramın program olarak uygulamasını yüksek seviyeli bir dille yapacağız.

ABEL (Gelişmiş Boolean Eşitlik Dili), sizin mantık devrelerinin yapısal formuna girmenize izin verir. ABEL , Data I/O şirketi tarafından programlanabilir mantık cihazlarının (PLD) kontrolü için geliştirilmiş endüstriyel standart donanım açıklama dili (HDL) dir. ABEL aynı amaç için üretilmiş diğer dillerden yapısal olarak daha basittir.

CLK, !FRAME, !IRDY, CB0, CB1, CB2, CB3 pin 1,2,3,6,7,8,9;
!EQL, !EQH, !RES pin 4,5,13;

CLK(saat sinyali),!FRAME(tersini al),!IRDY(tersini al),CB0,CB1, CB2, CB3(Yol Komut tipi ve Byte kullanımı),!EQL(tersini al),!EQH(tersini al),!RES(tersini al)'in giriş pinlerini tanımlamış olduk.

```
!DEVSEL, !TRDY          pin 22,23  istype 'reg,invert';
```

!DEVSEL(tersini al),!TRDY(tersini al) çıkış pinlerini tanımlarken ‘reg’ ifadesi ile kayıt edici (flip-flop) tanımlanır. Ayrıca çıkışa değer verilirken ‘invert’ tersi gönderilir.

```
WC1, WC0                pin 14,15  istype 'reg';
```

WC1 ve WC0 ise çıkış olarak kayıt edici (flip-flop) tanımlanıyor.

```
CMD_IOW=([FRAME,IRDY,CB3,CB2,CB1,CB0,EQL,EQH] == [1,0,0,0,1,1,1,1]);
```

CMD_IOW sabiti tanımlanırken birinci köşeli parantez içindeki pin değerleri ikinci köşeli parantez içindeki değerlerle karşılaştırılarak eğer eşit ise CMD_IOW sabiti ‘1’ yapılıyor değilse ‘0’ olur. Bu ifade program içinde her kullanıldığında işlem tekrarlanarak sonuç CMD_IOW’ye atanır.

```
BUS_IDLE = ([FRAME, IRDY] == [0,0]);
```

BUS_IDLE sabiti FRAME ve IRDY ‘0’ ise ‘1’ oluyor değilse ‘0’ olur. Bu işlemde programda kullanıldığında karşılaştırma sonucu her defasında BUS_IDLE’ a ‘0’ veya ‘1’ atanır.

```
[DEVSEL,TRDY,WC0,WC1].clk = CLK;
```

DEVSEL,TRDY,WC0 ve WC1 her clk sinyalinde değeri çıktı olarak verilir.

State 0:

```
if    RES    then 0;
else if BUS_IDLE then 4;
           else 0;
```

State 0 durumunda RES değeri '1' ise reset durumu var demektir. Kendine tekrar konumlanır. RES değeri '0' ise ve BUS_IDLE '1' ise state 4'e gidilir değilse kendine konumlanır.

State 4:

```

if    RES    then 0;
else if CMD_IOW then 2 with DEVSEL := 1; TRDY := 1;
      else if BUS_IDLE then 4;
            else 0;

```

State 4 durumunda RES '1' ise state 0 gidilir. Değilse CMD_IOW değeri '1' ise (alınan yol komutu yazma işlemi anlamına gelmektedir.) DEVSEL ve TRDY kayıtlarına (flip-flop) '1' değeri verilir ve state 2'ye gidilir. CMD_IOW '0' ise tekrar BUS_IDLE kontrol edilerek state 4 veya state 0 gidilir.

State 2:

```

DEVSEL.oe = 1; TRDY.oe = 1;
if    RES    then 0;
else if !FRAME then 3 with WC0 := !CB0; WC1 := !CB1;
      else 2 with DEVSEL := 1; TRDY := 1;

```

State 2 durumunda state 4 veya state 3'den gelindiği için DEVSEL ve TRDY kayıtlarına verilen değer çıkışta görünmesi için çıkışlar aktif edilir.

RES değeri '1' ise state 0 değilse FRAME kontrol edilir. !FRAME '1' ise veri Yolda demektir ve alması istenilen devreleri kontrol eden WC0 ve WC1 kayıtları '0' yapılarak state 3 e gidilir. !FRAME değeri '0' ise DEVSEL ve TRDY kayıtlarına '1' atanarak state 2 ye gidilir.

State 3:

```
DEVSEL.oe = 1; TRDY.oe = 1;
if RES then 0;
else if CMD_IOW then 2 with DEVSEL := 1; TRDY := 1;
      else 4;
```

State 3 durumunda state 2 den gelindiğinden DEVSEL ve TRDY kayıtlarına verilen değer çıkışta görünmesi için çıkışlar aktif edilir. RES değeri kontrol edilir '1' ise state 0 değilse CMD_IOW '1' ise yazma işlemi komutu alındı anlamına gelmektedir bu durumda DEVSEL ve TRDY kayıtları '1' yapılarak state 2'ye CMD_IOW değeri '0' ise state 4'e gidilir.

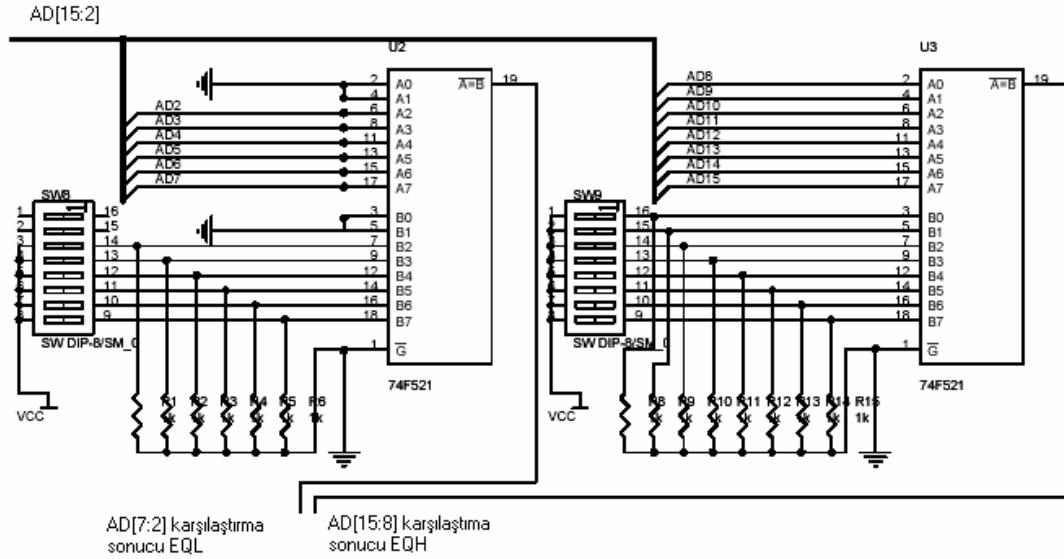
State 1:

```
goto 0;
```

State 1 diyagramda gösterilmemektedir. Karta beklenmedik bir hata ile karşılaşırsa işlemler state 0'dan başlaması için konulmaktadır.

5.2. Pci Kartını Yapımı

5.2.1. Adres çözümleme işlemi



Şekil 5.2. Adres çözümleme devresi

Devremizin adres çözümleme kısmında iki adet 8 bit karşılaştırıcı kullanılmaktadır. Çünkü PCI veri yolu I/O işlemlerinde 16 bit adres yoluna sahiptir.

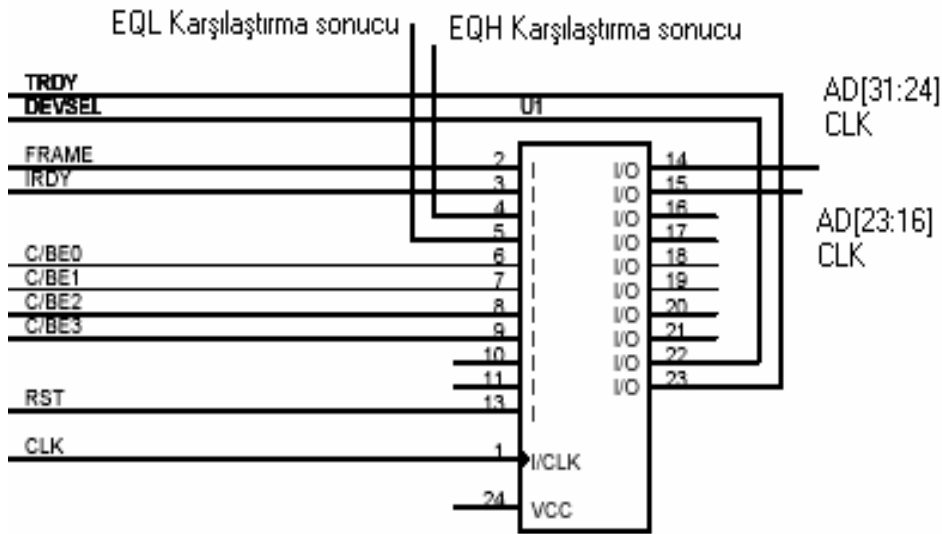
Birinci 8 bitlik karşılaştırma işleminde AD0 ve AD1 değerleri kullanılmamaktadır.(bu değerler byte değeri çözümlemede kullanıldığından dolayı) AD2-AD7 arasındaki gelen değerlerle bizim karşılaştırıcıya anahtarlarla uyguladığımız değerler eşit ise EQL (U2'nin 74F521'in 19. pin) '0' olacaktır.

İkinci 8 bitlik karşılaştırma işleminde AD8-AD15 değerleri ile bizim karşılaştırıcıya anahtarlarla uyguladığımız değerler eşit ise EQH (U3'ün 74F521'in 19 pin) '0' olacaktır.

Genelde prototip kartlarda uygulanan port adresleri 300-31F arasında olduğunda bizde kartımızda anahtarlar vasıtası ile karşılaştırıcılara uyguladığımız 0300H

değeridir. Fakat AD0 ve AD1 kullanılmadığından dolayı gerçekte kullandığımız port adresi 0300H-0303H(4 port) arasında olacaktır. 32 bit veri gönderildiğinden memory kullanımından farkı her 8 bit bir port'a gelmektedir.

5.2.2. Gal'in çalışması

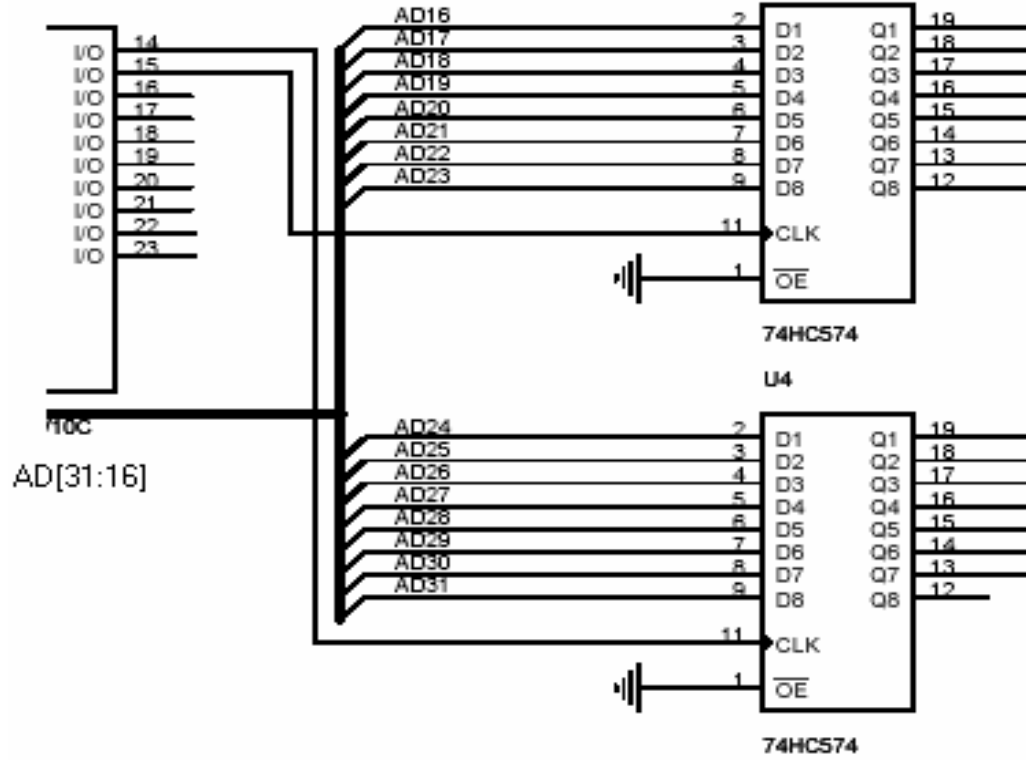


Şekil 5.3. PCI sinyallerini düzenleyen GAL devresi

Devremizin bu kısmında PCI veri yolu üzerinden gelen CBE[3:0] komut ve byte yetki değeri, FRAME, IRDY, RST ve CLK gelmektedir. Gelen sinyallerin senkronizasyonu sağlayan CLK sinyalidir.

Devremizin bir parçası olan karşılaştırma kısmından gelen EQL ve EQH değerlerine bağlı olarak üretilerek PCI veri yoluna iletilen DEVSEL ve TRDY sinyalleridir. Bu sinyalleri kullanarak bilgisayar üzerinde çalışan 32 bit özellikli bir programın kartımıza erişmek istediğini ve değer göndereceğini anlarız. Gönderilen değerlerin alınması için üretilen iki sinyalimiz daha mevcuttur bu sinyaller AD[31:16] 16 bitlik veri'leri almak için entegremizin 14 ve 15 pin'lerinde oluşan sinyallerdir.

5.2.3. Verilerin alınması

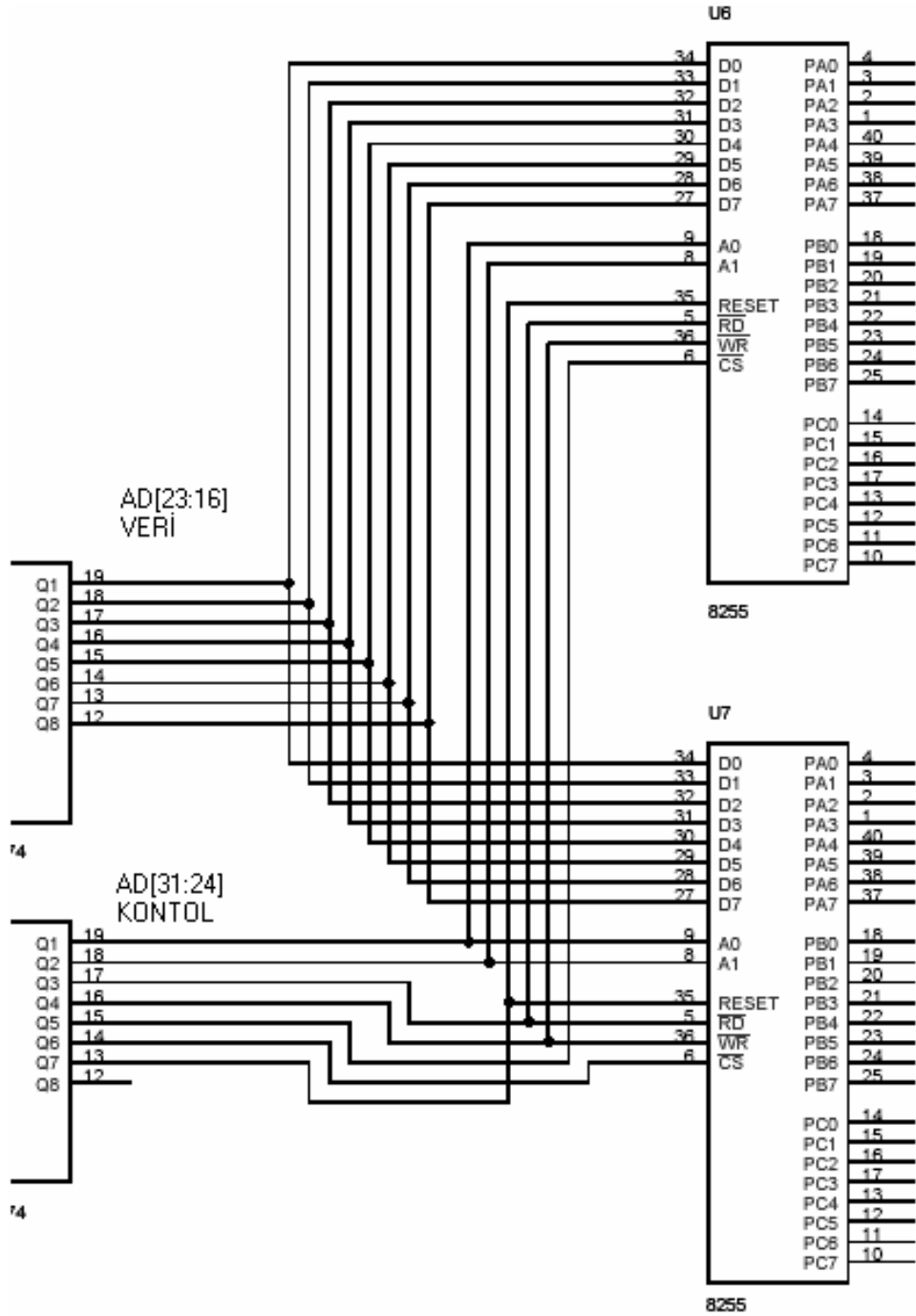


Şekil 5.4. Verilerin tutma işlemini yapan devre kısmı

Devremizin veri alma kısmında ise AD[31:16] toplam 16 bit olarak sinyal çözümleme entegresinden(GAL22V10) 14 ve 15 nolu bacaklarından alınan sinyallerimiz 74HC574 entegrelerimizin 11 nolu (clk) pin'lerine uygulanır.

PCI veri yolundan gelen sinyal değerlerine bağlı olarak, adres çözümleme işlemi ile kartımızın port'una gönderilen veriyi almak(tutmak) için 14 ve 15 nolu pin'lerde oluşan sinyaller AD[31:16] değerler 2 adet 8 bitlik tutucularımız tarafından alınması sağlanır.

5.2.4. 8255 entegrelerin programlanması



Şekil 5.5. 8255'lerin bağlantı yapıları

Devremizin bu kısmı ise kartımıza gönderdiğimiz 16 bit uzunluğundaki veriyi kullanarak 8255 entegrelerinin programlaması olacaktır. 8255 entegreleri 8 bit tabanlı çalışmaktadır. Gönderilen 16 bit'lik verinin ilk 8 bit kısmı veri diğer 8 bitlik kısmı ise 8255 entegrelerin kontrolü, port seçimi için kullanılacaktır.

AD[23:16] değerlerini tutan 74HC574 8255 entegrelerinin ortak veri yolu olarak kullanılmaktadır. Gelen veri her iki 8255 (U7 ve U8) aynı anda veri bacaklarında bulunmaktadır.

AD[31:24] değerlerini tutan 74HC574 entegresi ise iki adet 8255 entegresinin kontrol sinyalleri olarak kullanılmaktadır. Ortak olan kontrol sinyalleri;

AD24-AD25 değerleri sırası ile 8255 entegrelerimizin A0 ve A1 pin'lerine. AD26-AD27 değerleri sırası ile 8255 entegrelerimizin RD ve WR pin'lerine. AD28 değeri birinci 8255 entegremizin(U8) CS pin'ine. AD29 değeri ikinci 8255 entegremizin(U7) CS pin'ine. AD30 değeri ise her iki 8255 entegresinin RESET pin'ine bağlanmaktadır.

Çizelge 5.1. AD[31:24] bitlerinin kullanımı ve oluşan port numaraları

	AD31	AD30	AD29	AD28	AD27	AD26	AD25	AD24	
PORT ADI	X	RESET	2.CS	1.CS	WR	RD	A1	A0	PORT (HEX)
1.A	0	0	1	0	0	1	0	0	0x24
1.B	0	0	1	0	0	1	0	1	0x25
1.C	0	0	1	0	0	1	1	0	0x26
1.CW	0	0	1	0	0	1	1	1	0x27
2.A	0	0	0	1	0	1	0	0	0x14
2.B	0	0	0	1	0	1	0	1	0x15
2.C	0	0	0	1	0	1	1	0	0x16
2.CW	0	0	0	1	0	1	1	1	0x17

```
outb(VERİ,0x302);
outb(PORT,0x303);
outb(PORT+08,0x303);
```

0x302 port'una yüklenen değer seçilecek olan 8255 entegresine gönderilen değerdir. 0x303 port'una yüklenen değer hangi 8255'in seçildiğini belirleyen değer olacaktır. Daha sonra WR değeri '1' yapılarak 8255 VERİ değerini alması sağlanır.

```
outb(0x80,0x302);
```

0x80 hex değerini 0x302 port'una göndermektedir. Bu değer AD[23:16] 8 bit'i ifade etmektedir. Bu 8 bitlik değer her iki 8255'in veri alanına yüklenir.

```
outb(0x27,0x303);
```

0x27 hex değeri 0x303 port'una göndermektedir. Bu değer AD[31:24] 8 bit'i ifade etmektedir. Bu 8 bitlik değer birinci 8255'in kontrol kayıtçısını seçmiş oluruz.

Her iki komut sıra ile çalıştığında birinci 8255'in kontrol port'una 0x80 değerini yüklemiş oluruz. Bu değer birinci 8255'in port'larını çıkış olarak seçilmiş olur.

```
outb(0x80,0x302);
outb(0x17,0x303);
outb(0x17+0x08,0x303);
```

Bu komutlar çalıştığında ikinci 8255'in kontrol kayıtçısına 0x80 değeri yüklendiğinden dolayı ikinci 8255'in port'ları çıkış olarak seçilmiş oluruz.

```
outb(0xAA,0x302);
outb(0x24,0x303);
outb(0x24+0x08,0x303);
```

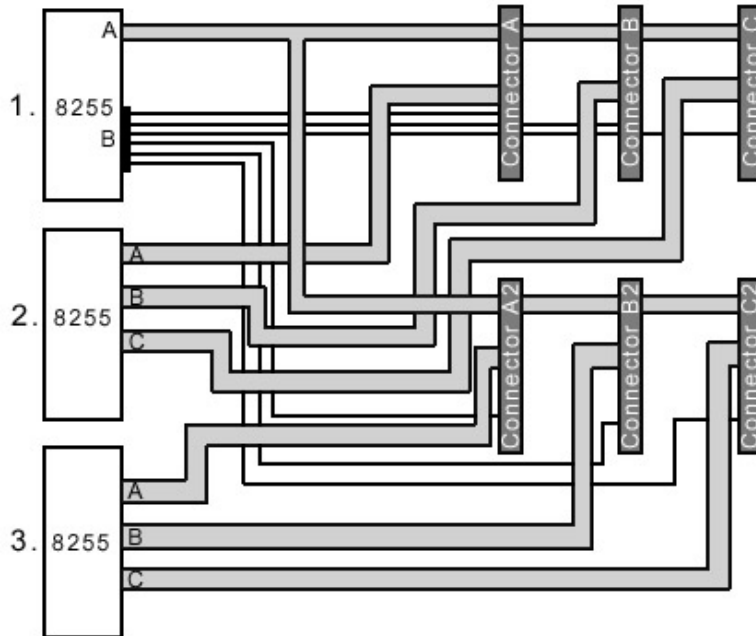
Bu komut yapıları çalıştırıldığında birinci 8255'in (bütün port'lar çıkış olarak seçilmişti) A port'una 0xAA değeri yüklemiş oluruz.

```
outb(0xBB,0x302);
outb(0x16,0x303);
outb(0x16,0x08,0x303);
```

Bu komutların yapmış olduğu işlev ise daha önceden çıkış olarak(kontrol kayıtcısı 0x80 değerini atayarak) seçilmiş olan ikinci 8255'in C port'una 0xBB değerini yüklemiş oluruz.

5.3. Kartın Çıkış Portu

Kartımızda 3 adet 8255 kullanılmaktadır. 1. 8255A kontrol amaçlı diğer 8255'ler ise veri amacı ile kullanılır.

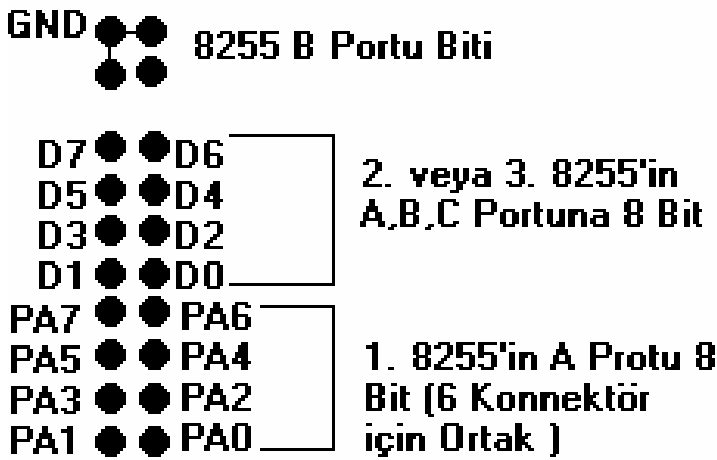


Şekil 5.6 PCI kartın blok şekli

Kartımızın 6 adet 20 pinlik çıkışları mevcuttur. 1. 8255'in A portu 6 çıkışında ortaktır. 2. ve 3. 8255'lerin A,B,C portları olmak üzere 6 adet port sırası ile çıkışlara bağlanmaktadır. 1. 8255'in B portunun $B_0 - B_5$ bitlerinde bu çıkışları birer bit olmak üzere bağlanmaktadır.

Buradaki konnektörlerin bu şekilde yapılmasındaki amaç sürülecek devrenin özellikleri göz önüne alınarak yapılmaktadır.

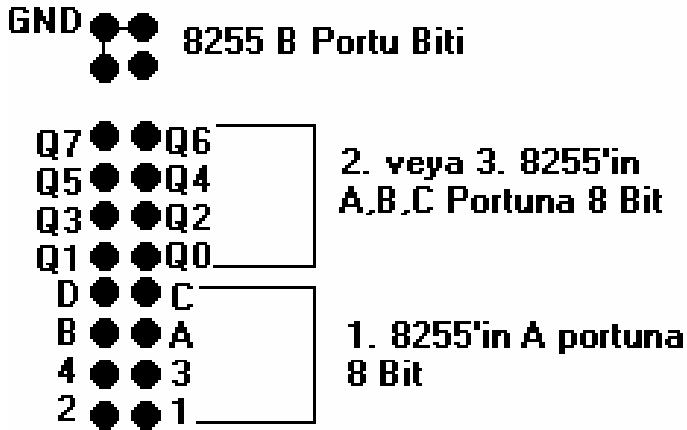
Fakat 8 bit ortak olmak üzere 8 bit ortak port (A,B,C Portlar), 1 bit ise 1.8255'in B portundan toplam 17 bitlik çıkış veya giriş amaçlı kullanılabilecek bir yapı elde edilmiş olunur. Buradaki yapı bu şartlar göz önüne alınarak genel amaçlı kullanılabilir (Gerçekte de bu devre EKG ve metallerine aşınma işlemlerinde step motor sürülmesi, Robot kolu kontrol işlemlerinde kullanılabilir.).



Şekil 5.7. Kart konnektörün yapısı

5.4. Display Kartının Gerçekleştirilmesi

5.4.1. Konektör

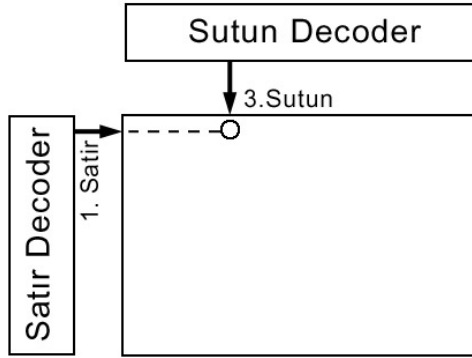


Şekil 5.8. Display kartının konektör şekli

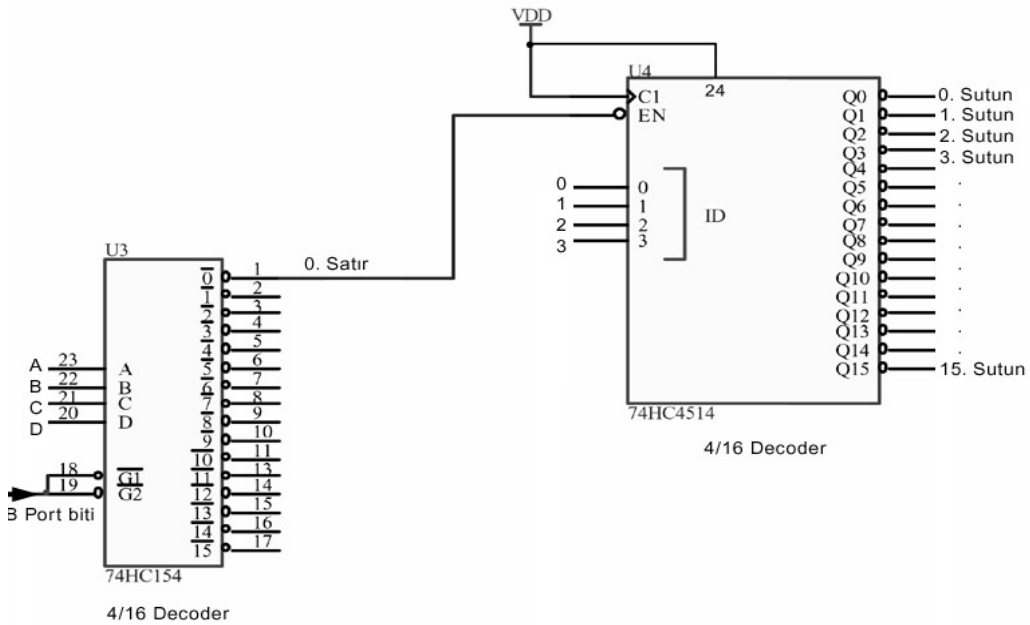
Display kartındaki konnektörle dizayn edilen ISA kartının konnektörü birebir bir şekilde bağlanmaktadır. Buradaki bağlantının neden bu şekilde olduğu devrenin birlikte (PCI ve display kartı) çalışma şeklinde daha iyi anlaşılacaktır. Display (Matrix) Satır-Sütun Kontrolü Displayın oluşturduğu hücrelerin kontrolü satır sütun şeklinde kontrol edilmektedir.

1.Satırdaki 3 sütundaki Hücre aktif edilecek ise; satır decoderine gönderilen 4 bitlik veri ile 1 seçilir. Sütun decoderine de gönderilen 4 bit ile sütun seçilir bu şekilde bir tarama metodu ile hücreler aktif hale getirilir.

5.4.2. Satır ve sütun kontrolü



Şekil 5.9. Satır ve sütunun tarama



Şekil 5.10. Satır ve sütunun belirlendiği devre

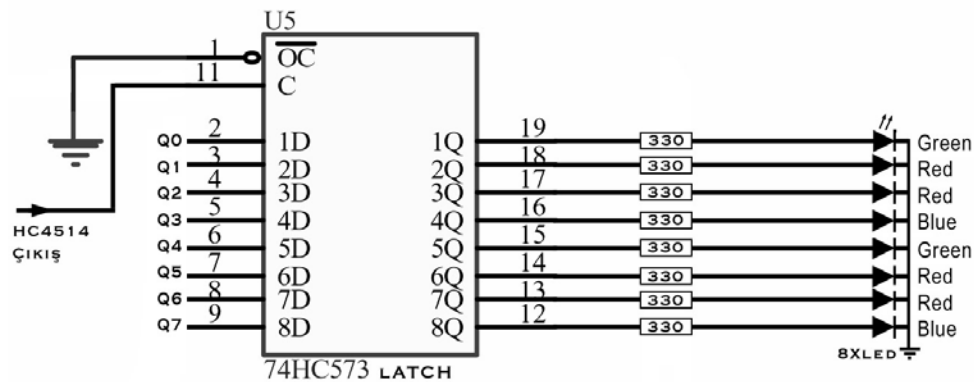
Her display panelinde (kart) bizim yapmış olduğumuz panoda 16 satır 32 sütun (Hücre) şeklinde 16 satır, 16 sütun gibi görülse de sütunun kontrol eden decoderin her biri çıkışı 2 adet hücreyi kontrol ettiğinden dolayı bu şekilde 32 hücreye (sütun) çıkmaktadır.

Her display panelini kontrol etmek için 1. 8255'in portundan bir bit kullanılmaktadır. Bu kullanılan bit 74HC154 4/16 decoder entegresini aktif yapmaktadır. Böylelikle panel aktif olmuş olur. Daha önce ki konnektör'ün açıklamasında A,B,C,D isimlendirilmiş bulunan 4 bit gerçekte 1. 8255'in A portunun 4,5,6,7 nolu bitleridir. Bu 4 bitin görevi paneldeki 16 adet satırların seçilmesinde kullanılmaktadır.

Konnektör deki 1,2,3,4 nolu bitler ise 1.8255'in A portunun 0,1,2,3 nolu bitlerine karşılık gelir ki bu 4 adet bitle 32 sütun hücresinin kontrol edilmesinde kullanılır.

Satır sütunların kontrol işleminde 74HC154 ve 4514 (CMOS) farklı iki tip entegrenin kullanılmasındaki neden hücreleri sürme işleminde kullanılacak olunan Latch'in aktif olabilmesi için değerinin gönderilmesi gerekir eğer ucuz 2 adet 74HC154 4/16 decoder arka arkaya kullanılmış olsa idi. Latch ile 4/16 decoder arasında NOT işlemi yapan bir entegre kullanılması gerekirdi. Bu şekilde bir çözüm ise panelin fiziki şartlarını zorlaştırır ve maliyeti artırır.

5.4.3. LED hücrelerinin sürülmesi

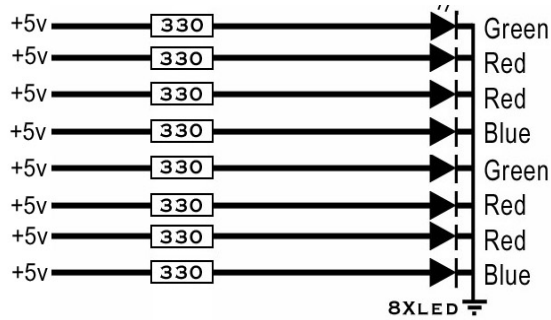


Şekil 5.11. LED'lerin sürülme devresi

2.veya 3. 8255 'in A,B veya C portundan gelen Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7 nolu 8 bitlik data Latch'in girişine uygulanmakta, satır ve sütunun seçilerek sütun hücresine aktif sinyali gönderildiğinde Latch'in girişine uygulanan 8 bitlik data Latch'in

çıkışına uygulanmaktadır. Burada ikinci bir 8 bitlik data gelse de çıkış etkilenmemekte fakat satır, sütun sinyali aynı hücreye uygulanıncaya kadar.

Daha önce de kısaca açıklanmıştıkt sütunu kontrol eden decoder (4514) 16 çıkışlı olmasına rağmen 32 hücreyi kontrol ettiğini bunun nedeni 8 bitlik Latch kullanılması ve her bir hücre 4 bitte kontrol edilmesinden dolayı sütun sayısı 32 adettir.



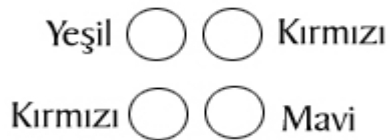
Şekil 5.12. LED'lerin renk sıralaması

Devrede 330ohm'luk direnç kullanılmasının nedeni Latch'in aşırı yüklenmesine engel olmak ve özellikle Yeşil Ledlerin düşük gerilim Mavi Ledlerin ise yüksek (9V) gerilimde çalışmasından dolayı standart bir değer seçilmektedir.

5.5. Devrenin Programlanması

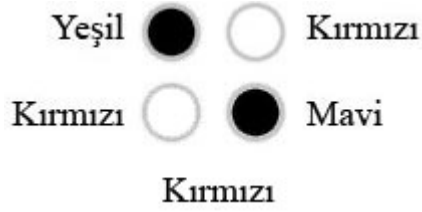
5.5.1. LED'lerle ana ve ara renklerin elde edilmesi

Bir hücreyi oluşturan Ledlerin yerleşik şekli aşağıda görülmektedir.

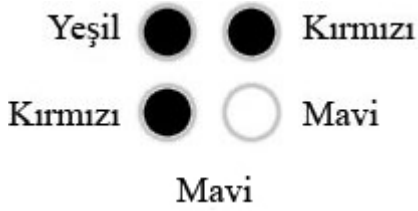


Şekil 5.13. LED'lerin görüntüleri

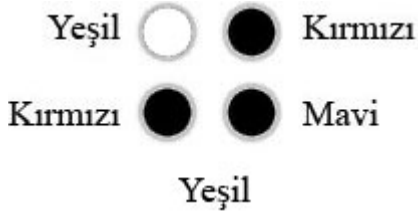
Burada 4 adet Led kullanılmaktadır. Kullanılan Led sayısı arttıkça elde edilen renk sayısı da artmaktadır.



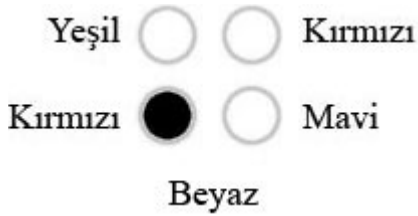
Şekil 5.14. Kırmızı renk elde edilmesi



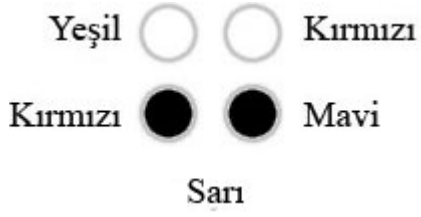
Şekil 5.15. Mavi renk elde edilmesi



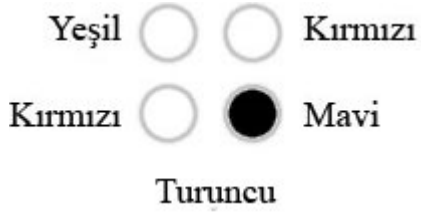
Şekil 5.16. Yeşil renk elde edilmesi



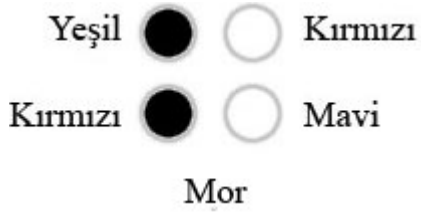
Şekil 5.17. Beyaz renk elde edilmesi



Şekil 5.18. Sarı renk elde edilmesi



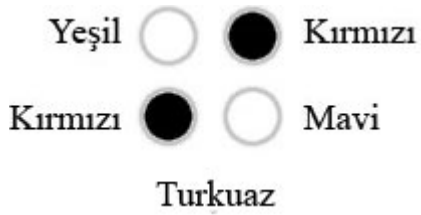
Şekil 5.19. Turuncu renk elde edilmesi



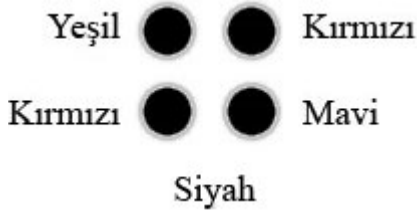
Şekil 5.20. Mor renk elde edilmesi



Şekil 5.21. Kahverengi elde edilmesi

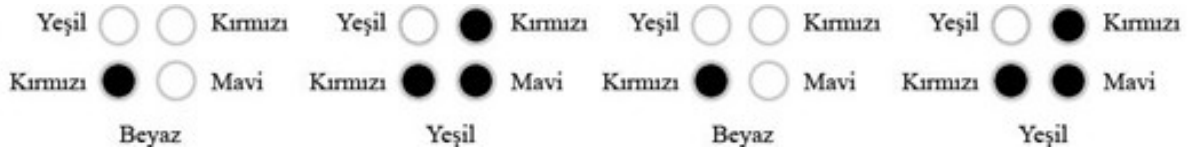


Şekil 5.22. Turkuaz renk elde edilmesi



Şekil 5.23.Siyah renk elde edilmesi

Şekillerde görüldüğü gibi Ledlerin belirli şekilde yanması ile çeşitli renkler elde edilmektedir. Burada gösterilen renklerden daha açık ve çeşitli renk elde edilebilir.



Şekil 5.24.Farklı renk tonu elde etme.

Yukarıda gösterildiği gibi yeşil, beyaz gibi dizilmesi ile yeşilin daha açık tonu veya pastel renkler elde edilebilir.

5.5.2. 8255 programlanması

8255 port düzenlenmesi

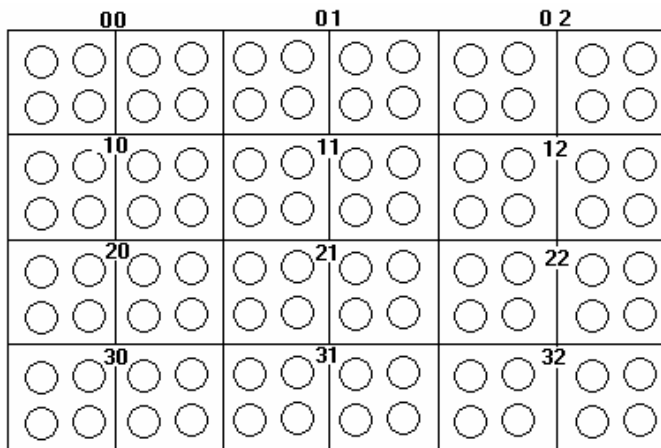
Devremizde kullandığımız portların hepsi panel uygulamasın da çıkış yapacak şekilde ayarlamaktır.

```
outb(0x80,0x302);
outb(0x27,0x303); //1.8255'in port'ları çıkış olarak ayarlandı.
outb(0x27+0x08,0x303);
outb(0x80,0x302);
outb(0x17,0x303); //2.8255'in port'ları çıkış olarak ayarlandı.
outb(0x17+0x08,0x303);
```

Yukarıdaki C kodunda da gösterildiği gibi 3 tane 8255'in A,B,C portları da çıktı yapacak şekilde ayarlanmaktadır. Fakat bizim yapmış olduğumuz kartta 1.8255 ile 2.8255 kullanılmaktadır. 1.8255'in A,B ve C portları da kullanılmakta ama 2.8255'in A portunun dışındaki B ve C portları kullanılmamaktadır.

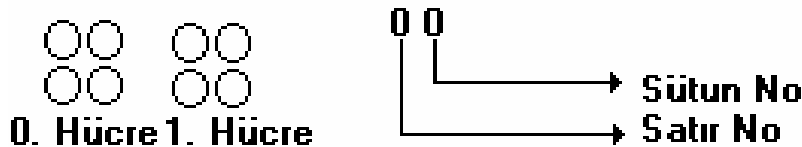
Fakat bizim yaptığımız panelin dışında bir uygulamada her üç 8255 'de bütün portları da kullanılabilir. Sadece bizim bu uygulamada yukarıda belirtilmiş portlar kullanılmaktadır.

Panelin programlanması



Şekil 5.25. 4x6'lık panelin satır – sütun yapısı

Yukarıdaki şekilde 6x4'lük bir Led Matrix Panel görülmektedir. Ama gerçekte ekonomik kısıtlamalardan dolayı böyle oluşur. Orijinal boyutları 32x16 hücredir.



Şekil 5.26. Hücrelerin satır – sütun yapısı

32x16	32x16	32x16
32x16	32x16	32x16

Şekil 5.27. Birden fazla panelin oluşturduğu yapı

Burada gösterdiğimiz şekil bir örnektir ama bizim yapmış olduğumuz PCI kartı ile 6 adet 32x16'lık panel kontrol edilebilir.

1. 8255'in B portu ile Panel kontrol edilmektedir.

1. 8255'in A portu paneldeki sütun no ile satır no belirtilmektedir.

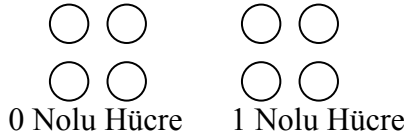
Bizim uygulamasını yaptığımız panelde ise 0. Hücre ile 1. Hücre 'nin programlanmasında 2. 8255'in A portu kullanılmaktadır.

```
outb(0x00,0x302);
outb(0x25,0x303);
outb(0x25+0x08,0x303);
outb(0x00,0x302);
outb(0x24,0x303);
outb(0x24+0x08,0x303);
```

Buradaki kodla paneldeki 00'nolu iki adet hücre (pixel) seçilmiştir. Buradaki ilk (LSB) 4 bit Sütun No'sunu belirlerken son (MSB) 4 bit Satır No'sunu belirlemektedir.

```
outb(0x01,0x302);
outb(0x14,0x303);
outb(0x14+0x08,0x303);
```

Bu kodla Satır No 0 , Sütun No 0 olan hücrenin 0 nolu hücresinin yeşil Ledinin yanması sağlanmaktadır.



Şekil 5.28. 8 Bitin kontrol ettiği hücre yapısı

Çizelge 5.2. Renklerin hex kodları

0 Nolu Hücredeki Ledlerin Hex ifadeleri		1 Nolu Hücredeki Ledlerin Hex ifadeleri	
01	Yeşil	01	Yeşil
02	Kırmızı	02	Kırmızı
04	Kırmızı	04	Kırmızı
08	Mavi	08	Mavi

Örnek Bir Uygulama:

```

outb(0x80,0x302);
outb(0x27,0x303); //1.8255 portları çıkışa ayarlanmaktadır.
outb(0x27+0x08,0x303);
outb(0x80,0x302);
outb(0x17,0x303); //2.8255 portları çıkışa ayarlanmaktadır.
outb(0x17+0x08,0x303);
outb(0xFE,0x302);
outb(0x25,0x303); //1.8255 B portundan panel seçilmektedir.
outb(0x25+0x08,0x303);
outb(0x00,0x302);
outb(0x24,0x303); //1.8255'in A portu 00 satır-sütun seçilmektedir.
outb(0x24+0x08,0x303);

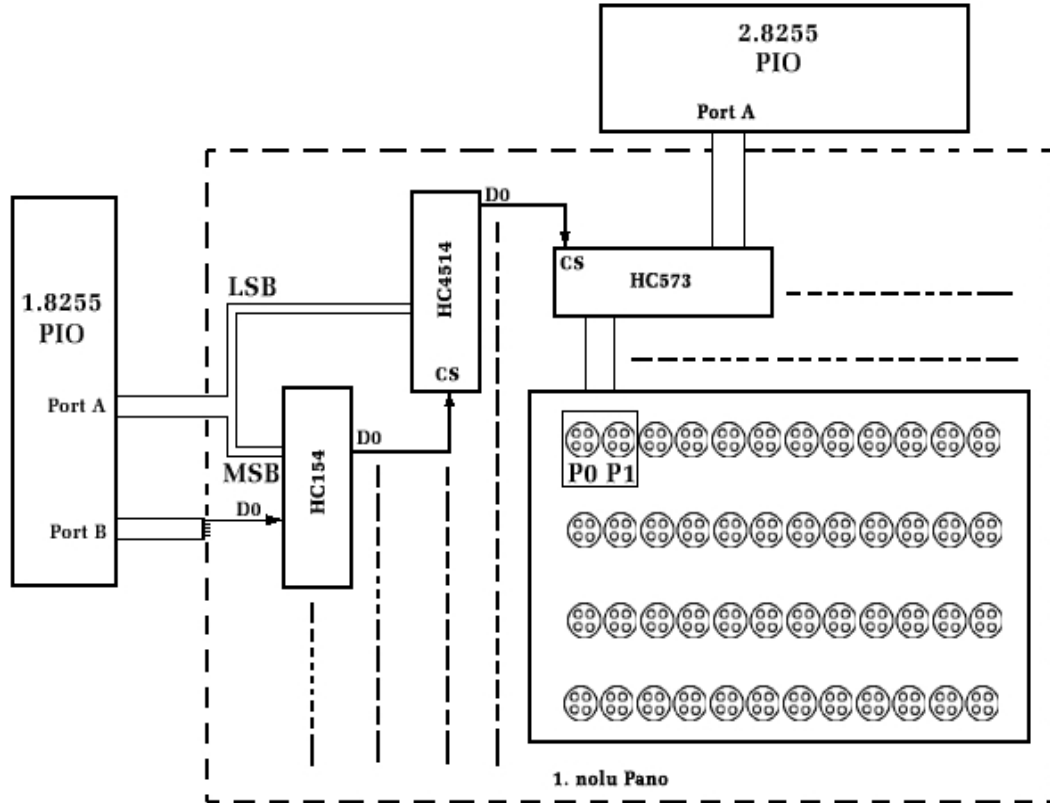
```



```
outb(0x0B,0x302);
```

```
outb(0x14,0x303); // 2.8255 A portu (yeşil,kırmızı,mavi) beyaz renk
```

```
outb(0x14+0x08,0x303);
```



Şekil 5.29. Kart ve panelin ortaklaşa çalışma

6. SONUÇ VE DEĞERLENDİRME

Yapılan bu tez konusu birbirlerinden çok farklı iki çalışmadan meydana gelmektedir. ISA kartlarını incelediğinde belirli bir frekansın üzerine çıkamamakta yani yapılan ISA kart en fazla 12MHz gibi bir frekansı destelemektedir. Fakat günümüzde bu çalışma frekansı çok küçük bir değerde kalmaktadır. Bu çalışmada daha önceden yapmış olduğumuz ISA kart yerine PCI kartı yapmış bulunmaktayız. Fakat PCI veri yolu bağlantılarının ISA'ya göre daha ince olması ve bazı baskı devre çıkarma imkanlarından dolayı zor olmaktadır. Bu tez ticari bir projeye dönüştürülmek istendiğinde ISA yerine PCI tercih edilmesi yerinde olacaktır. Yeni üretilen anakartlarda ISA veri yolu standardı yoktur. Çalışmamızda PCI kartı yaparken daha çok bu tezin ikinci parçası olan ekran parçası yerine istenildiği zaman farklı genel amaçlı işlerde de kullanılabilecek şekilde dizayn etmeye çalıştık. Tezin ikinci parçası olan bir ekran modülünü dizayn etmeye çalıştık. Yapılan bu modüller birleştirilerek dev ekranlar yapılabilir. Pratikte bu modüllerin çoğaltılması bilgisayarın ve bunu kontrol eden kartın yapısına bağlıdır.

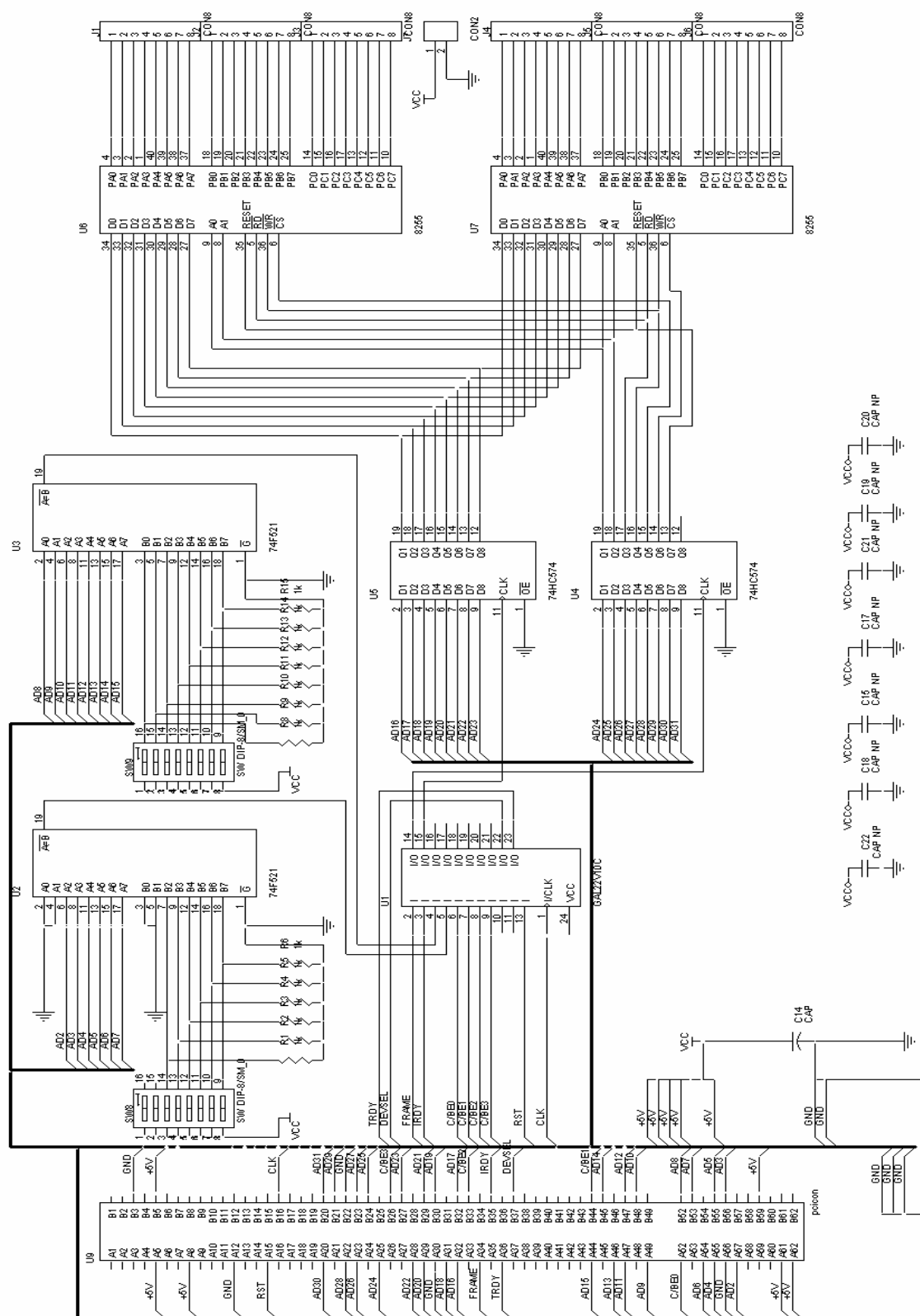
Taiwan gibi bir ülke bu alanda yılda 10 Milyar Dolar gibi bir kazanç elde etmektedir. Bu çalışmada bizi zorlayan unsurlar baskı devrenin istenildiği kalitede çıkarılamaması ve kullanılan Led'lerin kalitesi ögleki farklı ital edilen Led'ler arasında ışık ve renk farkı oluşmakta, Led'ler özellikle mavi Led ülkemizde kullanılmadığından bu Led'lerin alımı 3 ay gibi bir süre beklemek zorunda kaldık. Bu modülde kullanılan tarama metodu bu tür ekranaların büyük kısmında kullanılmaktadır. Fakat elde edilecek renk sayısı artıkça kullanılan (biz 4 Led kullandık). Led sayısı artıyor Led sayısı 8'in üzerine çıktığı zaman özel Lacth kullanılması daha etkili olacaktır. Hp'nin özel ürettiği Led'ler ile 65535 renk elde edilebilmektedir. Bizim yapmış olduğumuz bu çalışmada özellikle ekran modülünün dizaynı gerçek amaçına ulaşmaktadır.

KAYNAKLAR

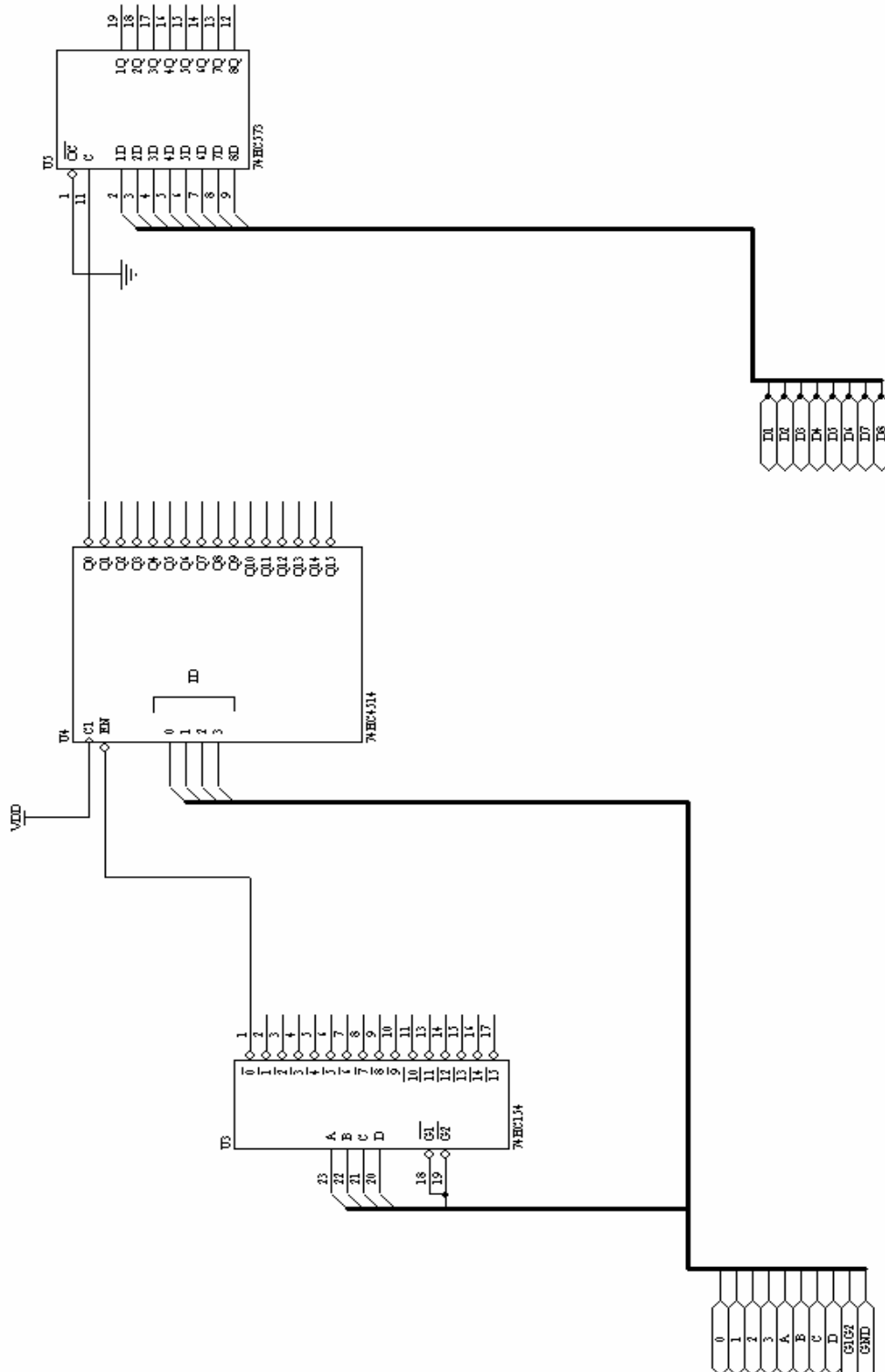
1. Gümüşkaya,H.,”Mikroişlemciler ve Bilgisayarlar”,*Alfa Basım Yayın Dağıtım A.Ş.*,İstanbul,271-287 (1999).
2. Brey,B.,“The Intel Microprocessors”,*Macmillan Publishing Company*,New York,337-409 (1994).
3. Finkelstein,E., “Design and Implementation of PCI bus based Systems”, Yüksek Lisans Tezi, *Tel Aviv University*,Tel Aviv,30-64 (1997).
4. Liu, Y.C. ve Gibson, G., “Microcomputer System:8086/8088 Family”, *Prentice – Hall*,New York, 369-372 (1986).
5. Intel, “82C55A Programmable Peripheral Interface”, *Intel Data Books*, Santa Clara,1-20(1986).
6. Axelson,J.,”The Microcontroller Idea Book“,Dinçer G.,*Lakeview Research*, İstanbul,109-111 (1997).
7. Arsan,T., ve Çölkesen,R., “Lojik Devre Tasarımı”, *Papatya yayıncılık.*, İstanbul, 203-211 (2001).
8. Schubert,E. F., “Light-Emitting diyotes”, *Jonh WILEY& SONS*,New York,275-345 (1982).
9. Floyd,T.,”Digital Fundamentals”,*Merrill Publishing Company*,New York,244-320 (1990).
- 10.Limann, O. ve Kuntman, H., “Renkli Televizyon Temelleri”, *Yüce Yayın A.Ş.*,İstanbul,40-45 (1985).

EKLER

EK-1 PCI Veri Yolu ve PCI Kartı Devresi.



EK-2 Led Ekran Modülü Devresi.



EK-3 PCI Kart üzerinden Led Ekranı Kontrol eden C programı.

```

#include <asm/io.h>
//Verilerin gonderilecegi port
#define DATA 0xF302
//PIO larin secilmesi,portlarin belirlendigi port
#define STATUS 0xF303
//PIO portlari cikis olarak aktif
#define ACTIVE 0x80
//Birinci PIO Portlari
#define PIO_1_CW 0x27 //Kontrol portu Adresi
#define PIO_1_A 0x24 //A portu Adresi
#define PIO_1_B 0x25 //B portu Adresi
#define PIO_1_C 0x26 //C portu Adresi
//Ikinci PIO Portlari
#define PIO_2_CW 0x17 //Kontrol portu Adresi
#define PIO_2_A 0x14 //A portu Adresi
#define PIO_2_B 0x15 //B portu Adresi
#define PIO_2_C 0x16 //C portu Adresi
//Renk Tanimlari
//Hucre bir
#define h1G 0x1 //yesil
#define h1R 0x2 //kirmizi
#define h12R 0x4 //2xkirmizi
#define h1B 0x8 //Mavi
//Hucre bir ara renkler
#define h1GR h1G+h1R //sari
#define h1G2R h1G+h1R+h12R //Turuncu
#define h1RR h1R+h12R //Koyu Kirmizi
#define h1GB h1G+h1B //Turkuaz
#define h1RB h1R+h1B //Mor
#define h1RRB h1R+h12R+h1B //Erguan
#define h1RGB h1R+h1G+h1B //Beyaz
#define h1RRGB h1R+h1G+h1B+h12R //Pembe
//Hucre iki
#define h2G 0x10 //yesil
#define h2R 0x20 //kirmizi
#define h22R 0x40 //2xkirmizi
#define h2B 0x80 //Mavi
//Hucre iki ara renkler
#define h2GR h2G+h2R //sari
#define h2G2R h2G+h2R+h22R //Turuncu
#define h2RR h2R+h22R //Koyu Kirmizi
#define h2GB h2G+h2B //Turkuaz
#define h2RB h2R+h2B //Mor
#define h2RRB h2R+h22R+h2B //Erguan

```


EK-3 (Devam) PCI Kart üzerinden Led Ekranı Kontrol eden C programı

```

#define h2RGB h2R+h2G+h2B //Beyaz
#define h2RRGB h2R+h2G+h2B+h22R //Pembe
#define PIO_RESET 0x40
#define PIO_PASSIVE 0x30
int main()
{
//Ekran tarama portlari
unsigned char
portlar[]={0x00,0x01,0x02,0x10,0x11,0x12,0x20,0x21,0x22,0x30,0x31,0x32};

void send_data(unsigned int port_adres,unsigned char data)
{
    outb(data,DATA);
    outb(port_adres,STATUS);
    outb(port_adres+0x8,STATUS);
    outb(PIO_PASSIVE,STATUS);
}

void reset(void)
{
    int i;

    for (i=0;i<12;i++)
    {
        send_data(PIO_1_A,portlar[i]);
        send_data(PIO_2_A,0x0);
    }
}

void test(unsigned int color1,unsigned int color2)
{
    int i;
    for(i=0;i<12;i++)
    {
        send_data(PIO_1_A,portlar[i]);
        send_data(PIO_2_A,color1+color2);
        sleep(1);
    }
}

send_data(PIO_1_CW,ACTIVE);
send_data(PIO_2_CW,ACTIVE);
reset();

printf("Kırmızı\n");
test(h1R,h2R);

```

EK-3 (Devam) PCI Kart üzerinden Led Ekranı Kontrol eden C programı.

```
reset();
printf("Yeşil\n");
test(h1G,h2G);
reset();

printf("Mavi\n");
test(h1B,h2B);
reset();

printf("Koyu Kırmızı\n");
test(h12R,h22R);
reset();

printf("Sarı\n");
test(h1GR,h2GR);
reset();

printf("Turuncu\n");
test(h1G2R,h2G2R);
reset();

printf("Koyu Kırmızı\n");
test(h1RR,h2RR);
reset();

printf("Turkuaz\n");
test(h1GB,h2GB);
reset();

printf("Mor\n");
test(h1RB,h2RB);
reset();

printf("Erguan\n");
test(h1RRB,h2RRB);
reset();

printf("Beyaz\n");
test(h1RGB,h2RGB);
reset();

printf("Pembe\n");
test(h1RRGB,h2RRGB);
reset(); }
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : BAŞ, Hasan Hüseyin
Uyruğu : T.C.
Doğum tarihi ve yeri : 22.06.1968 Sivas
Medeni hali : Evli
e-mail : corps58@hotmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Gazi Üniversitesi/ Elektronik-Bilgisayar	1994
Lise	Sivas E.M.L.	1987

İş Deneyimi

Yıl	Yer	Görev
1994-1997	Trabzon Anadolu E.M.L.	Öğretmen

Yabancı Dil

İngilizce

Hobiler

Bilgisayar teknolojileri, Basketbol