

KOCAELİ ÜNİVERSİTESİ

FPGA İLE PCI KART TASARIMI

LİSANS BİTİRME TEZİ

Ahmet KALE

Bölüm: Elektronik ve Haberleşme Mühendisliği

Danışman: Yrd. Doç. Dr. Mehmet YAKUT

KOCAELİ, 2007

FPGA İLE PCI KART TASARIMI

Ahmet KALE

Anahtar Kelimeler: PCI, FPGA, VHDL, Serializer, Deserializer, LVDS, SERDES

Özet: PCI (Peripheral Component Interconnect) yerel veri yolu genişleme kartları, tümleşik G/Ç denetleyici yongaları ve bilgisayarın ana hafıza ve işlemcisi için yüksek hızlı bir bağlantı sağlamaktadır. Günümüzde çok büyük veri bloklarının yüksek hızlarda iletiminin gerçekleştirilmesi bir zorunluluk haline gelmiştir. Bilgisayar ile dış dünyadaki çevre birimleri arasında yüksek hızlı veri iletişimi için PCI tabanlı cihazlar günümüzde oldukça yaygın bir şekilde kullanılmaktadır.

FPGA'ler donanımları kullanıcı tarafından yapılandırılabilir kapı dizileridir. Yeniden yapılandırılabilir FPGA'ler günümüzde ağ teknolojilerinden telekomünikasyona, sayısal işaret işlemeden gömülü mikro-denetleyicilere, otomotiv endüstrisinden tüketici elektroniğine kadar birçok alanlarda kullanılmaktadır.

Bu tezde yapılan çalışmanın amacı FPGA tabanlı genel amaçlı bir PCI I/O kartı tasarlamaktır. Kartın şematik tasarım ve baskı devre çizimleri gerçekleştirilmiştir.

FPGA BASED PCI CARD DESIGN

Ahmet KALE

Keywords: PCI, FPGA, VHDL, Serializer, Deserializer, LVDS, SERDES

Abstract: The Peripheral Component Interconnect (PCI) local bus provides a high-performance interconnection for expansion cards, integrated I/O controller ICs, and the computer's main memory and processor. In today's operating environments, it is absolutely necessary that large blocks of data be transferred expeditiously. PCI devices are commonly used for high speed data transmission between computers and peripherals.

The FPGAs are configurable gate arrays by user. Nowadays, reconfigurable FPGAs are being used in mainstream products, from network technology to telecom, from DSP to embedded microcontroller, and from automotive industry to consumer electronics.

The purpose of the work in this thesis is for designing general purpose an FPGA based PCI I/O Card. Schematic design and printed circuit board design of the card have been implemented.

ÖNSÖZ ve TEŞEKKÜR

PCI veri yolu mimarisi adres ve veri yollarının birlikte kullanıldığı yüksek performanslı 32 yâda 64 bitlik bir veri yoludur. Bu veri yolu, tümleşik çevre birimi kontrol cihazları ile çevre birimi ek kartları ve işlemci/hafıza sistemleri arasında bir bağlantı mekanizması olarak kullanmak için tasarlanmıştır.

FPGA'ler genellikle ASIC tasarımların ilk örnekleri veya yeni bir algoritmanın fiziksel gerçekleştirilebilirliğini doğrulamak amacıyla bir donanım ortamı sağlamak için kullanılırlar. Bununla birlikte düşük geliştirme maliyetleri ve kısa sürede pazara sunulabilme özellikleri sayesinde gittikçe son ürün yelpazesindeki yerlerini almaktadır.

Bu çalışmada önce PCI veri yolu, FPGA ve SERDES yapısı açıklanarak FPGA tabanlı bir PCI kartının şematik ve PCB tasarımının EAGLE programı kullanılarak yapılışı anlatılmaktadır.

Öğrenim hayatım boyunca bana her türlü maddi ve manevi desteklerini esirgemeyen ve beni bugünlere getiren aileme ve desteklerinden dolayı tüm dost ve arkadaşlarıma çok teşekkür ederim.

Bitirme tezini almamda beni teşvik eden ve destekleyen değerli hocam Yrd. Doç.Dr. Mehmet YAKUT'a, kartın tasarım ve imalatı aşamasında verdiği destekten dolayı sn. Bekir ASLAN'a ve beni FPGA ve VHDL ile tanıştıran ve her türlü desteğini esirgemeyen Öğr. Gör. Suhap ŞAHİN hocama gönülden teşekkürlerimi sunarım.

Ahmet KALE

İÇİNDEKİLER

ÖZET	ii
ABSTRACT.....	iii
ÖNSÖZ ve TEŞEKKÜR.....	iv
İÇİNDEKİLER	v
ŞEKİLLER LİSTESİ	viii
TABLOLAR DİZİNİ	x
1. GİRİŞ	1
2. PCI V2.2 Yerel Veriyolu Mimarîsi.....	2
2.1 PCI Veri yolu Uygulamaları	3
2.1.2 PCI Veriyolu Özellikleri ve Avantajları	3
2.2 Sinyal Tanımı.....	5
2.2.1 Sistem Pinleri	5
2.2.2 Adres ve Veri Pinleri	6
2.2.3 Arayüz Kontrol Pinleri.....	7
2.2.4 Sorun Çözme (Arbitration) Pinleri.....	8
2.2.5 Hata Raporlama Pinleri.....	9
2.2.6 Kesme Pinleri.....	9
2.2.7 Ek Sinyaller.....	10
2.2.8 64-Bit Veri Yolu Genişleme Pinleri (Seçime Bağlı)	10
2.2.9 JTAG/Boundary Scan Pinleri (Seçime Bağlı)	12
2.3. Elektrik Şartnamesi.....	13
2.3.1 5V'dan 3.3V'a Geçiş Yol Haritası.....	13
2.3.2 5V Sinyalleşme Ortamı.....	13
2.3.2.1 DC Şartnameler.....	13
2.3.2.2 AC Şartnameleri.....	14
2.3.3 3.3V Sinyalleşme Ortamı.....	15
2.3.3.1 DC Şartnameleri.....	15
2.3.3.2 AC Şartnameleri.....	15
2.3.4 Pin Yerleşim Planı (Pinout) Önerisi	16
2.3.5 Genişleme Kartı Şartnamesi	16

2.3.5.1 Kart Pin Tahsisi.....	16
2.3.6 Güç Gereksinimleri.....	20
2.3.6.1 Dekuplajlama.....	20
2.3.6.2 Güç Tüketimi.....	20
2.3.6.3 Fiziksel Gereksinimler.....	21
2.3.6.3.1 Yol Uzunluğu Sınırları.....	21
2.4 Mekanik Şartnamesi.....	21
2.4.1 Genişleme Kartı Fiziksel Boyutları ve Toleransı.....	22
3. Spartan-3E FPGA AİLESİ.....	29
3.1 Genel Özellikleri.....	31
3.2 Mimari Tanımlama.....	33
3.2.1 IOB Yapısı.....	33
3.2.2 SelectIO Sinyal Standartları.....	34
3.2.3 Yonga-üzeri Fark Sonlandırma.....	36
3.2.4 Yukarı-Çekme Aşağı-Çekme Dirençleri.....	37
3.2.5 I/O Kümelere Ayırma (Banking).....	38
3.3 Yapılandırılabilir Lojik Bloklar (CLB) ve Dilim (Slice) Kaynakları.....	38
3.3.1 CLB Dizisi.....	38
3.3.2 Dilimler.....	39
3.4 Sayısal Saat Yöneticisi (DCMs).....	41
3.5 Yapılandırma.....	43
3.5.1 Yapılandırma Dosyası.....	43
3.6 DC Karakteristikleri.....	46
3.6.1 DC Giriş ve Çıkış Seviyeleri.....	46
3.6.2 Bypass Kapasitesi Hesapları.....	48
3.6.2.1 Çekirdek VCC Bypass Kapasitesi Hesabı.....	48
3.6.2.2 I/O Güç Kaynağı Bypass Kapasitesi Hesabı.....	50
3.7 Spartan-3E Pin Bağlantıları.....	52
4. DS92LV18 18-Bit Bus LVDS Serializer/Deserializer.....	57
4.1 Genel Tanım.....	57
4.2 İşlevsel Tanım.....	60
5. VHDL'E GİRİŞ.....	66
5.1 VHDL NEDİR?.....	66

5.2 VHDL YAPISI.....	66
5.2.1 Yapısal Tanımlama	67
5.2.2 Davranışsal Tanımlama	67
5.2.3 Özgül Bir VHDL Örneği	68
5.3 VHDL Terimleri	71
5.4 Veri Tipleri.....	72
5.4.1 Nesne Tipleri (Object Types).....	72
5.4.1.1 İşaretler (Signal).....	73
5.4.1.2 Değişkenler (Variable).....	74
5.4.1.3 Sabitler (Constant)	76
5.4.2 Veri Tipleri (Data Types).....	77
5.4.2.1 Skaler Tipler (Scaler Types)	78
5.4.2.1.1 Tamsayılar (Integer).....	79
5.4.2.1.2 Gerçek Sayılar (Real).....	80
5.4.2.1.3 Sıralı Sayılar (Enumerated).....	80
6. LVDS'E GİRİŞ	82
6.1 LVDS Modası	82
6.2 Düşük Gürültü ve Düşük Güçle Gelen Hız.....	83
6.2.1 LVDS Nasıl Çalışır?	83
6.2.2 LVDS Standardı.....	84
6.3 LVDS Yongaları	84
6.4 Bus LVDS (BLVDS)	85
6.5 LVDS Yapılandırmaları.....	86
6.6 FPGA ve ASIC İçerisine Gömülü LVDS I/O'lar	87
7. EAGLE KULLANARAK DEVRE TASARIMININ GERÇEKLENMESİ.....	89
7.1 EAGLE'da Kütüphane Oluşturma	89
7.2 Devre Şeması Tasarımı	94
8. SONUÇ.....	96
KAYNAKLAR	97
ÖZGEÇMİŞ	100

ŞEKİLLER DİZİNİ

Şekil 2.1 PCI Yerel Veriyolu Uygulamaları	2
Şekil 2.2 PCI Sistemi Blok Diyagramı	3
Şekil 2.3 PCI Pin Listesi	6
Şekil 2.4 PCI Kart Konektörleri.....	13
Şekil 2.5 Önerilen Pin Yerleşim Planı	16
Şekil 2.6 PCI Raw Kart (5V)	22
Şekil 2.7 PCI Raw Kart (3.3V ve Üniversal).....	22
Şekil 2.8 PCI Raw Değişken Yükseklikli Kısa Kart (5V 32-bit)	23
Şekil 2.9 PCI Raw Değişken Yükseklikli Kısa Kart (3.3V 32-bit)	23
Şekil 2.10 PCI Raw Değişken Yükseklikli Kısa Kart (5V 64-bit)	24
Şekil 2.11 PCI Raw Değişken Yükseklikli Kısa Kart (3.3V 64-bit)	24
Şekil 2.12 G/Ç Pencere Yüksekliği	25
Şekil 2.13 5V 32-bit Kart Konektör Kenarı Ölçüleri ve Toleransı.....	25
Şekil 2.14 5V 64-bit Kart Konektör Kenarı Ölçüleri ve Toleransı.....	26
Şekil 2.15 3.3V 32-bit Kart Konektör Kenarı Ölçüleri ve Toleransı.....	26
Şekil 2.16 3.3V 64-bit Kart Konektör Kenarı Ölçüleri ve Toleransı.....	27
Şekil 2.17 Üniversal 32-bit Kart Konektör Kenarı Ölçüleri ve Toleransı	27
Şekil 2.18 Üniversal 64-bit Kart Konektör Kenarı Ölçüleri ve Toleransı	28
Şekil 2.19 PCI Kart Konektör Ucu Bağlantıları	28
Şekil 3.2 Spartan-3E Basitleştirilmiş Giriş/Çıkış bloğu (IOB).....	34
Şekil 3.3 Differential (Fark) Giriş ve Çıkışları	37
Şekil 3.4 Spartan-3E I/O Kümeleri (üstten görünüm)	38
Şekil 3.5 CLB Yerleşimi.....	39
Şekil 3.6 CLB İçerisinde Dilimlerin Düzeni.....	40
Şekil 3.7 SLICEM Basitleştirilmiş Diyagramı	40
Şekil 3.8 DCM İşlevsel Blokları ve Birleştirilmiş Sinyaller.....	42
Şekil 3.9 DLL'in Basitleştirilmiş İşlevsel Diyagramı.....	42
Şekil 3.10 Master Serial Modda Yapılandırma.....	45
Şekil 3.11 Slave Serial Modda Yapılandırma.....	45
Şekil 3.12 Fark Giriş Gerilimleri	47

Şekil 3.12 Fark Sinyal Çifti Etiketlemesi	53
Şekil 4.1 DS92LV18 SERDES Blok Şeması.....	58
Şekil 4.2 Paralel-seri Dönüştürücü PLL Kilitlenme Süresi ve PWRDN, 3-STATE ..	59
Şekil 4.3 Seri-paralel Dönüştürücü PLL Kilitlenme Süresi ve PWRDN, 3-STATE..	59
Şekli 4.4 Paralel-seri Dönüştürücü Gecikmesi	59
Şekli 4.5 Seri-paralel Dönüştürücü Gecikmesi.....	60
Şekil 5.1 Bir yapı tanımlama örneği	67
Şekil 5.2 Count2 tasarımının yapısı	69
Şekil 5.3 VHDL Veri Tipleri Diyagramı	78
Şekil 6.1 100Ω fark empedanslı ortam aracılığıyla bağlanmış LVDS sürücü ve alıcının basitleştirilmiş şeması	83
Şekil 6.2. Noktadan-Noktaya Yapılandırma.....	86
Şekil 6.3 Çiftyönlü Half-Duplex Yapılandırma.....	87
Şekil 6.4 Çokbağlantılı Yapılandırma.....	87
Şekil 6.5 Backplane Performansının Tümüleşik ve Ayrık Çözümler Arasında Karşılaştırılması	88

TABLolar DİZİNİ

Tablo 2.1 5V DC Şartname.....	14
Tablo 2.2 5V AC Şartname.....	14
Tablo 2.3 3.3V DC Şartname.....	15
Tablo 2.4 3.3V AC Şartname.....	15
Tablo 2.5 Present Sinyali Tanımlaması	17
Tablo 2.6 PCI Genişleme Kartı Pin Yerleşim Planı	17
Tablo 2.7 32-bit Genişleme Kartı Pin Özeti	19
Tablo 2.8 64-bit Genişleme Kartı Pin Özeti	19
Tablo 3.1 Spartan-3E FPGA ailesi üyeleri	30
Tablo 3.2 Mevcut Kullanıcı ve Fark (Diff) G/Ç Sayıları.....	32
Tablo 3.3 Single-Ended IOSTANDARD Küme Uyumluluğu	35
Tablo 3.4 Differential IOSTANDARD Küme Uyumluluğu.....	36
Tablo 3.5 Spartan-3E CLB Kaynakları.....	39
Tablo 3.6 DLL Sinyalleri.....	42
Tablo 3.7 Spartan-3E Yapılandırma Dosyası Boyutları	44
Tablo 3.8 Yapılandırma Modu Seçenekleri ve Pin Ayarları.....	44
Tablo 3.9 Mutlak Maksimum Oranlar	46
Tablo 3.10 Önerilen Çalışma Şartları	46
Tablo 3.11 Single-Ended Standartları Kullanan Kullanıcı I/O'ları İçin Önerilen Çalışma Koşulları.....	47
Tablo 3.12 Fark Sinyal Standartlarını Kullanan Kullanıcı I/O'ları İçin Önerilen Çalışma Koşulları.....	48
Tablo 3.13 Spartan-3E Pin Türleri.....	52
Tablo 3.14 Her Bir Paket ve Aygıt Göre Kullanılabilir Maksimum Kullanıcı I/O Sayıları	53
Tablo 3.15 Pin Yerleşim Tablosu	54
Tablo 6.1. ANSI/TIA/EIA-644 (LVDS) standartları.....	84

1. GİRİŞ

Bu tezde anlatılan projenin gerçeklenmesindeki amaç, bilgisayarın PCI slotuna yerleştirilecek FPGA tabanlı bir PCI kartı tasarlayarak bilgisayar ile dış dünyadaki sistemler arasında yüksek hızlı seri veri alış verişi yapmaktır.

Projenin gerçeklenebilmesi için birçok aşamadan geçilmesi gerekmektedir. Öncelikle yapılması gereken PCI veri yolu mimarisinin çalışmasını kavramaktır. Daha sonra ise isteklerimizi karşılayabilecek bir FPGA seçimi ve bu FPGA'in özelliklerinin iyice kavranması ardından PCI iletişimi için kullanılacak IP Core'un belirlenmesi, seri veri iletim arayüzünün seçilmesi ve devre tasarımı için kullanılacak programın belirlenmesiyle şematik tasarımı gerçekleştirilerek PCB tasarımının yapılması ve son aşamada kartın montajının yapılması gerekmektedir. Kartın montajı tamamlandığında sıra yazılım kısmına gelecektir. Öncelikle FPGA'e yüklenecek program hazırlanıp EPROM'a aktarılmalı ve kartın testlerinin yapılması gerekmektedir. Testler olumlu sonuçlandığında en son aşama olarak kartın aygıt sürücüsü ve programının yazılması ve ardından sistemin çalışır hale getirilmesi ile tasarım sonuçlandırılacaktır.

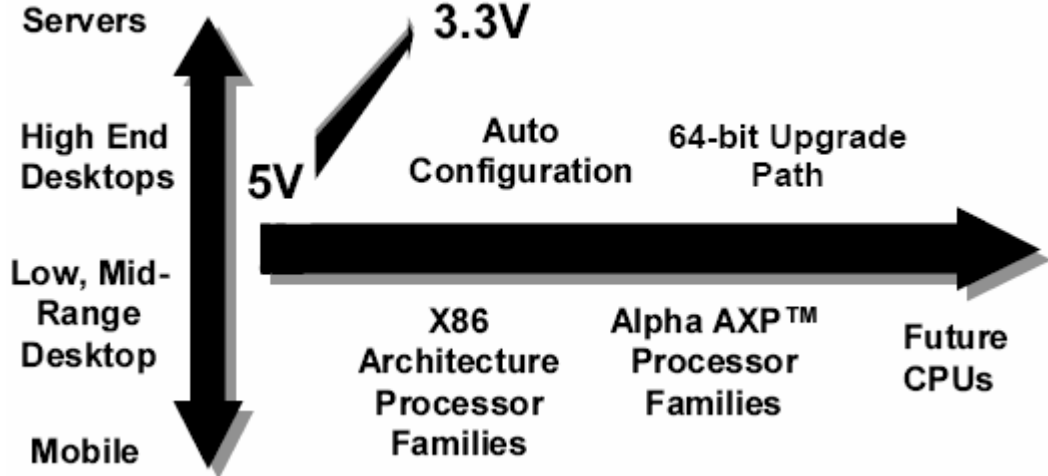
Bu tezde önce PCI veri yolu mimarisi genel hatlarıyla anlatılmaya çalışılmış, ardından devre tasarımında kullanılacak FPGA ve SERDES elemanları incelenmiş, VHDL ve LVDS standardı tanıtılmış ve devre tasarımının EAGLE programı kullanılarak nasıl yapıldığı anlatılmıştır.

2. PCI V2.2 YEREL VERİ YOLU MİMARİSİ

PCI veri yolu mimarisi adres ve veri yollarının birlikte kullanıldığı yüksek performanslı 32 yâda 64 bitlik bir veri yoludur. Bu veri yolu, tümleşik çevre birimi kontrol cihazları ile çevre birimi add-in kartları ve işlemci/hafıza sistemleri arasında bir bağlantı mekanizması olarak kullanmak için tasarlanmıştır.

2.1 PCI Veri yolu Uygulamaları

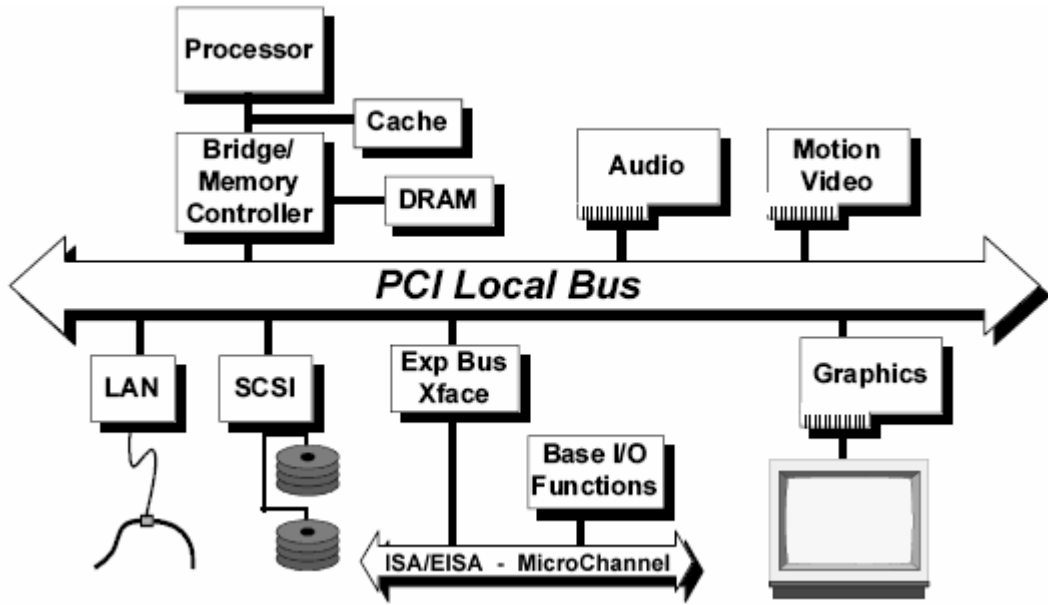
PCI veriyolu, birinci amaç olarak bir endüstri standardı oluşturarak düşük maliyetli yüksek performans sunan veri yolu mimarisi olarak sunulmuştur. Günümüz sistemlerine yeni fiyat-performans noktaları sunarken, gelecekteki sistem ihtiyaçlarını karşılamak ve çoklu platformlar ve mimarilere de uygulanabilirlik de çok önemlidir.



Şekil 2.1 PCI Yerel Veriyolu Uygulamaları

Yerel veriyolu uygulamalarının ilk odağı alt ve üst uç masaüstü sistemleri iken, PCI yerel veri yolu aynı zamanda mobil uygulamalarından büyük sunuculara kadar olan ihtiyaçları da karşılamaktadır (Şekil 2.1). PCI elemanları ve add-in kart arayüzü işlemciden bağımsızdır ve bu gelecek nesil işlemcilere verimli geçişe ve çoklu işlemci mimarileri ile birlikte kullanıma izin verir.

İşlemciden bağımsızlık PCI veri yoluna G/Ç işlevleri için optimizasyon sağlar, işlemci/hafıza altsistemi ile yerel veriyolunun uyumlu çalışmasına olanak sağlar ve grafiklere ek olarak çoklu yüksek performanslı çevrebirimlerini barındırır (hareketli video, LAN, SCSI, FDDI, sabit disk sürücüler v.s.). Gelişmiş video ve çoklu ortam displayler (HDTV ve 3B grafikler) ve diğer yüksek bantgenişlikli G/Ç'lar veri yolu bantgenişliği ihtiyacını arttırmaya devam etmektedir. PCI BUS V2.2, ileriye ve geriye uyumlu 32 bit ve 64 bit veri yoluna sahip 33MHz ve 66MHz veriyolu mimarisi desteği sağlamaktadır. Şekil 2.2'de PCI sisteminin blok yapısı görülmektedir.



Şekil 2.2 PCI Sistemi Blok Diyagramı

2.1.2 PCI Veriyolu Özellikleri ve Avantajları

PCI veriyolu çeşitli ürün kuşaklarına yüksek performanslı bir yerel veriyolu standardı oluşturmak için tanımlanmıştır. PCI özellikleri aşağıdaki avantajlarla kategorize edilmiştir.

Yüksek performans:

- 33MHz 32bit (132MB/s) veriyolundan 33MHz 64bit (264MB/s) veriyoluna ve 66MHz 32bit (264MB/s) veriyolundan 66MHz 64bit (528MB/s) veriyoluna kolay güncelleme.
- Değişken uzunluklu lineer ve yazma ve okuma için cacheline wrap mode bursting, yazmaya bağımlı grafik performansını artırır.

- İşlemci/hafıza altsistemi ile tamamen uyum kabiliyeti.
- 33MHz yâda 66MHz'e kadar senkronize veriyolu.
- Gizli merkezi sorun çözme.

Düşük maliyet:

- Doğrudan silikon bağlantısı için optimize edilmiştir. Elektriksel/sürücü ve frekans şartnameleri standart ASIC teknolojisi ve diğer proseslerle uyumlu.
- Çoklayıcı mimari PCI elemanlarının boyutunu ve pin sayısını azaltmaktadır.

Kullanım kolaylığı:

- PCI add-in kartları ve bileşenlerinin tam otomatik yapılandırma desteğine izin verir. PCI aygıtları, yapılandırma için gereken aygıt bilgilerini barındıran saklayıcıları barındırmaktadır.

Uzun ömür:

- İşlemciden bağımsızdır. Gelecek nesil işlemciler gibi çoklu işlemci ailelerini destekler (köprü yâda doğrudan bütünleşerek).
- 64bit adreslemeyi destekler.
- 5V ve 3.3V işaretleme ortamları belirtilmiştir.

Güvenilirlik:

- Küçük boyutlu add-in kartlar
- Present sinyalleri, güç kaynaklarının add-in kartları gözlemleyerek beklenen sistem kullanımı için optimize edilmesine olanak sağlar.
- 32bit ve 64bit add-in kartları ve bileşenlerinin ileriye ve geriye uyumluluğu.
- 33MHz ve 64MHz add-in kartları ve bileşenlerinin ileriye ve geriye uyumluluğu.
- MC-türü genişleme konektörleri.

Esneklik:

- Her hangi PCI master/target'dan her hangi bir PCI master'a noktadan noktaya erişime izin veren tam multi-master kabiliyeti.
- Paylaşımlı bir slot, standart ISA, EISA veya MC kart yâda bir PCI add-in karta yer sağlar.

Veri doğruluğu:

- Robust client platformlarının gerçekleşmesine olanak sağlayan veri ve adres üzerinde eşlik sağlar.

Yazılım uyumluluğu:

- PCI bileşenleri, mevcut sürücü ve uygulama yazılımları ile tam uyumlu olabilmektedir. Aygıt sürücülere çeşitli platform sınıfları için taşınabilirlik.

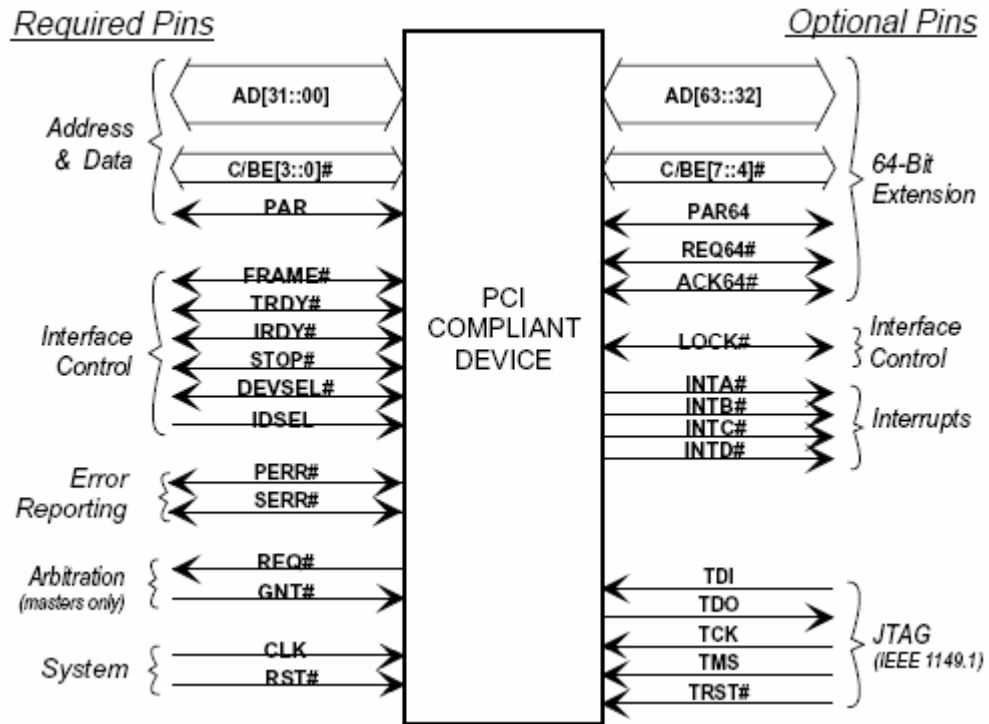
2.2 Sinyal Tanımı

PCI arayüzü target-only bir aygıt için minimum 47, veri ve adresleme, arayüz kontrol, sorun çözme (arbitration) ve sistem fonksiyonlarını idare eden bir master için en az 49 pin gerektirmektedir. Şekil 2.3'de PCI pinleri solda gerekli pinler ve sağda opsiyonel pinler olmak üzere fonksiyonel gruplar halinde gösterilmektedir. Sinyallerin sonundaki # simgesi, sinyalin aktif sıfır olduğunu göstermektedir.

2.2.1 Sistem Pinleri

CLK: *Saat (Clock)*, PCI üzerindeki tüm işlemler için zamanlamayı sağlamaktadır ve bütün PCI cihazları için giriştir. Diğer tüm PCI sinyalleri (RST#, INTA#, INTB#,INTC# ve INTD# dışında) CLK'nın yükselen kenarında örneklenir ve tüm diğer zamanlama parametreleri bu kenara uyarak tanımlanır. PCI 33MHz yâda 66MHz'e kadar çalışır ve genellikle minimum frekans DC (0Hz)'dir.

RST#: *Reset*, PCI-özümlü saklayıcıları, sıralayıcı ve sinyalleri, uygun bir duruma getirmekte kullanılmaktadır. RST # sinyali verildiğinde tüm PCI çıkış sinyalleri başlangıç durumlarına sürülmelidir. Genellikle bu çıkışların asenkron olarak tri-state konumuna geçmesi demektir. REQ# ve GNT# de tri-state konumunda olmalıdır. AD, C/BE# ve PAR sinyallerinin reset durumunda boşta kalmalarından kaçınmak için merkezi kaynak reset sırasında bu hatları (lojik 0 olarak) sürebilir (bus parking). RST#, CLK ile asenkron olabilir.



Şekil 2.3 PCI Pin Listesi

2.2.2 Adres ve Veri Pinleri

AD[31:00]: *Adres ve Veri* aynı PCI pinleri üzerinde çoklanmışlardır. Bir bus işlemi bir adres fazı ve ardından bir yâda daha fazla olan veri fazından meydana gelir. Adres fazı FRAME# sinyali aktif edildiğindeki ilk saat çevrimidir. Adres fazında, AD[31:00] fiziksel adresi (32bit) içerir. Bu G/Ç için bir byte adresi, yapılandırma ve hafıza için ise bir DWORD adresidir. Veri fazında AD[07:00] en az anlamlı byte'ı (lsb), AD[31:24] ise en anlamlı byte'ı (msb) içermektedir. IRDY# aktifken yazılan

veri sabit ve geçerli, TRDY# aktifken okunan veri sabit ve geçerlidir. Veri IRDY# ve TRDY#'nin her ikisi aktifken bu saat çevrimleri süresince transfer edilir.

C/BE[3::0]#: *Bus Komutu (Bus Command) ve Byte Yetkilendirme (Byte Enable)* aynı PCI pinlerinde çoklanmıştır. Bir işlemin adres fazı süresince C/BE[3::0]# Bus komutlarını tanımlamaktadır. Veri fazı süresince ise Byte yetkilendirme olarak kullanılır. Byte yetkilendirme bütün veri fazı için geçerlidir ve hangi byte yolunun anlamlı veri taşıdığını belirtir. C/BE[0]# byte 0 (lsb)'ı, C/BE[3]# byte 3 (msb)'ü temsil etmektedir.

PAR: *Eşlik (parity)*, AD[31::0] ve C/BE[3::0]# için çift eşliktir. Eşlik üretme tüm PCI vasıtaları tarafından istenmektedir. PAR her bir adres fazından sonraki bir çevrim boyunca sabit ve geçerlidir. Veri fazları için, yazma işlemlerinde IRDY# veya okuma işlemlerinde TRDY#'den herhangi birinin aktif edilmesinden sonraki bir çevrim boyunca sabit ve geçerlidir. PAR bir kez geçerli olduğunda geçerli veri fazının tamamlanmasından sonraki bir çevrime kadar geçerli kalır. PAR AD[31::00] ile aynı zamanlamaya sahiptir fakat bir çevrim geciktirilmiştir. Master, adres ve veri yazma fazları için PAR'ı sürerken Target, veri okuma fazlarında PAR'ı sürmektedir.

2.2.3 Arayüz Kontrol Pinleri

FRAME#: *Çerçeve çevrimi (Frame Cycle)*, bir erişim başlangıcını ve süresini belirtmek için geçerli master tarafından sürülmektedir. FRAME# bir bus işleminin başladığını belirtmek için aktif edilir. FRAME# aktif olduğu sürece veri transferi devam eder. FRAME# kaldırıldığında işlem son veri fazında yâda tamamlanmıştır.

IRDY#: *Başlatıcı Hazır (Initiator Ready)*, mevcut veri fazı işleminin tamamlanması için başlatıcı aracının (Bus Master'ın) hâkimiyetini gösterir. IRDY#, TRDY# ile birlikte kullanılır. Veri fazı IRDY# ve TRDY# nin ikisi de aktif edildiği herhangi bir çevrimde tamamlanır. Yazma süresince IRDY# AD[31::00] üzerinde geçerli bir verinin olduğunu belirtir. Okuma süresince, Master'ın veriyi kabul etmeye hazırlandığını belirtir. Bekleme çevrimleri IRDY# ve TRDY# birlikte aktif olana kadar eklenir.

TRDY#: *Target Hazır (Target Ready)*, mevcut veri fazı işleminin tamamlanması için target aracının (seçilen sürücünün) hâkimiyetini gösterir. TRDY#, IRDY# ile birlikte kullanılır. Veri fazı, TRDY# ve IRDY# nin ikisi de aktif edildiği herhangi bir çevrimde tamamlanır. Okuma süresince TRDY# AD[31::00] üzerinde geçerli bir verinin olduğunu belirtir. Yazma süresince, Target'ın veriyi kabul etmeye hazırlandığını belirtir. Bekleme çevrimleri TRDY# ve IRDY# birlikte aktif olana kadar eklenir.

STOP#: *Stop*, geçerli Target'ın, Master'a mevcut işlemi durdurması için istekte bulunduğunu gösterir.

LOCK#: *Lock*, tamamlanması için çoklu işlemler gerektirebilen bir Bridge'e yapılan atomik işlemleri göstermektedir. LOCK# aktif edildiğinde özel olmayan işlemler o anda kilitli olmayan bir Bridge'de yürütülebilir. Kilitli işlemler sadece Host Bridge, PCI-to-PCI bridge ve expansion bus bridge tarafından başlatılabilir.

IDSEL: *Sürücü başlatma seçme (Initialization Device Select)*, yapılandırma okuma ve yazma işlemleri süresince bir çip seçme (chip select) olarak kullanılır.

DEVSEL#: *Sürücü Seçme (Device Select)*, aktif olarak sürüldüğünde, sürülen aygıtın kendi adresini geçerli erişimin köprüsü olarak çözdüğünü belirtir. Giriş olarak DEVSEL#, bus üzerindeki herhangi bir aygıtın seçilip seçilmediğini belirtir.

2.2.4 Sorun Çözme (Arbitration) Pinleri

REQ#: *İstek (Request)*, bir aracının (agent) veri yolunu kullanmak istediğini hakeme bildirmektedir. Bu, noktadan noktaya bir sinyaldir. Her bir Master, RST# aktif edildiğinde tri-state konumunda olması gereken, kendi REQ# sinyaline sahiptir.

GNT#: *Onaylama (Grant)*, aracıya veri yoluna erişimin onaylandığını belirtmektedir.

2.2.5 Hata Raporlama Pinleri

PERR#: *Eşlik Hatası (Parity Error)*, özel çevrim (Special cycle) dışındaki tüm PCI işlemleri süresince veri eşlik hatalarını raporlamak içindir. PERR# pini tri-state konumunda tutulmalı ve bir veri eşlik hatası olduğunda veriyi takip eden iki çevrimde aracı tarafından aktif edilmelidir. PERR# 'in minimum süresi veri eşlik hatası tespit edilen her bir veri fazı için bir çevrimdir.

SERR#: *Sistem Hatası (System Error)*, özel çevrim komutunda bir adres yâda veri eşlik hatası olduğunda yâda kötü bir sonuca neden olabilecek diğer sistem hatalarının rapor edilmesi içindir. Eğer bir aracı bir NMI kesmesinin oluşturulmasını istemiyorsa farklı bir raporlama mekanizmasına ihtiyaç duyulur. SERR#, saf open-drain'dir ve hatayı rapor eden aracı tarafından tek bir PCI çevrimi için aktif olarak sürülür.

2.2.6 Kesme Pinleri

PCI üzerindeki kesmeler seçime bağlıdır ve seviye hassasiyetli olarak tanımlanmıştır. INTx# aktif etme ve kaldırma CLK ile asenkronudur. Bir aygıt, mesaj işaretli kesmeleri (message signaled interrupts-MSI) kullanmaya yetkili ise, kendi aygıt sürücüsünden bir istekte bulunmak için INTx# sinyalini aktif eder. INTx# sinyali aktif edildiğinde, aygıt sürücüsü askıdaki isteği silene kadar aktif olarak kalır. İstek silindiğinde, aygıt RSTx# sinyalini kaldırır. PCI tek fonksiyonlu aygıtlar için bir tane kesme hattı tanımlamaktadır. Tek fonksiyonlu aygıtlar için sadece INTA# kullanılabilirken diğer üç kesme hattının bir anlamı yoktur.

PCI'da INTA#, INTB#, INTC# ve INTD# olmak üzere dört adet kesme tanımlanmıştır.. Çok fonksiyonlu aygıttaki herhangi bir fonksiyon INTx# hatlarından herhangi birine bağlanabilmektedir. Kesme Pin saklayıcısı hangi INTx# hattının, hangi kesme isteği için kullanılacağını belirtmektedir.

2.2.7 Ek Sinyaller

PRSNT[1:2]#: *Mevcut (Present)* sinyalleri, aygıt sinyalleri değildir, fakat bir add-in kartı tarafından sağlanmaktadır. Present sinyalleri slotta fiziksel olarak bir add-in kartının bulunduğu ve kartın toplam güç ihtiyacını belirtmektedir. Bu sinyaller add-in kartları için gereklidir ancak anakartlar için seçime bağlıdır.

CLKRUN#: *Saat çalışıyor (Clock running)*, bir aygıtta CLK'nın durumunu belirtmek için giriş olarak kullanılan seçime bağlı bir sinyaldir ve open-drain bir çıkış aygıt tarafından CLK'yı başlatma yâda hızlandırma isteği için kullanılmaktadır. CLKRUN#, merkezi kaynak (central resource) tarafından CLK'yı yavaşlatma yâda durdurma izni istemek için kullanılan bir sustained tri-state (s/t/s) sinyaldir. Merkezi kaynak CLK çalışırken CLKRUN#'ın aktif olarak kalmasından ve CLK'yı yavaşlatma yâda durdurma izni isteği için CLKRUN#'ı kaldırmaktan sorumludur.

M66EN: *66MHz_Enable* pini, aygıtta veri yolunun 66MHz yâda 33MHz'de çalıştığını bildirmektedir.

PME#: *Güç Yönetimi olayı (Power Management Event)* sinyali, aygıt tarafından aygıtta yâda sistem durumunda bir değişiklik isteği için kullanılan seçime bağlı bir sinyaldir. PME#'nin aktif edilmesi ve kaldırılması CLK ile asenkronudur.

3.3Vaux: Seçime bağlı 3.3 volt yardımcı güç kaynağı, bir PCI add-in kartın ana gücü yazılım tarafından kapatıldığında, güç yönetimi olayları üretmek için PCI add-in karta güç verir. PCI veri yolu güç yönetimini desteklemeyen bir sistem yâda add-in kartı, 3.3Vaux pinini ayrılmış (reserved) olarak ele almalıdır.

2.2.8 64-Bit Veri Yolu Genişleme Pinleri (Seçime Bağlı)

64-Bit genişleme pinleri toplu olarak seçime bağlıdır. Eğer 64-Bit genişleme kullanılacak ise bu bölümdeki bütün pinlerin kullanılması gerekmektedir.

AD[63::32]: *Adres ve Veri* aynı PCI pinleri üzerinde çoklanmışlardır ve 32 adet ek biti sağlamaktadır. Adres fazında (DAC komutu kullanırken ve REQ64# aktifken), 64-Bitlik adresin üst 32-biti transfer edilmektedir. Diğer durumlarda bu pinler ayrılmıştır fakat sabit ve belirsizdir. Veri fazında REQ64# ve ACK64# aktif edilerek 64-bit işlemi akdedildiğinde ek 32-bitlik veri transfer edilmektedir.

C/BE[7::4]#: *Bus Komutu (Bus Command) ve Byte Yetkilendirme (Byte Enable)* aynı PCI pinlerinde çoklanmışlardır. Adres fazı süresince (DAC komutu kullanırken ve REQ64# aktifken), asıl veri yolu komutu C/BE[7::4]# üzerinden transfer edilir. Veri fazı süresince ise Byte yetkilendirme olarak kullanılır. Byte yetkilendirme REQ64# ve ACK64# aktif edilerek 64-bit işlemi akdedildiğinde hangi byte yolunun anlamlı veri taşıdığını belirtir. C/BE[4]# byte 4 'ü, C/BE[7]# byte 7 'yi temsil etmektedir.

REQ64#: *64-Bit transfer isteği (Request 64-Bit Transfer)*, geçerli Bus Master tarafından aktif edildiğinde verinin 64-bit olarak transfer edilmek istendiğini belirtir. REQ64# aynı zamanda FRAME# ile aynı zamanlamaya sahiptir.

ACK64#: *64-bit transfer kabul (Acknowledge 64-bit Transfer)*, adresini gerçekten geçerli erişimin hedefi olarak çözen aygıt tarafından aktif olarak sürüldüğünde target'ın 64-bitlik veri transferine hazır olduğunu belirtmektedir.

PAR64: *Eşlik yüksek DWORD (Parity Upper DWORD)*, AD[63::32] ve C/BE[7::4]'ü koruyan çift eşlik bitidir. PAR64, REQ64# aktifken her hangi bir işlemde her bir veri fazından sonraki bir çevrim geçerli kalmalıdır. 64-bit veri fazları süresince, yazma işlemlerinde IRDY# veya okuma işlemlerinde TRDY#'den herhangi birinin aktif edilmesinden sonraki bir çevrim boyunca sabit ve geçerlidir. PAR64 AD[63::32] ile aynı zamanlamaya sahiptir fakat bir çevrim geciktirilmiştir. Master, adres ve veri yazma fazları için PAR64'ü sürerken Target, veri okuma fazlarında PAR64'ü sürmektedir.

2.2.9 JTAG/Boundary Scan Pinleri (Seçime Bağlı)

IEEE 1149.1 standardı, *Test Access Port ve Boundary Scan Mimarisi* PCI aygıtları için seçime bağlı olarak dâhil edilmiştir. IEEE 1149.1 standardı, 1149.1-uyumlu çiplerin dizaynı için kulları ve izinleri belirtmektedir. Bir aygıtta Test Erişim Portu (TAP) eklemek, aygıtı ve takılan kartı test etmede kullanılan boundary scan işlemine olanak sağlar. TAP, PCI aygıtı içerisinde TAP denetleyicisi ile seri olarak bağlamakta kullanılan dört pinden oluşmaktadır (opsiyonel olarak beş).

TCK (in): *Test Clock*, TAP işlemi süresince durum bilgisini ve aygıtta giren ve çıkan test verilerini ölçmek için kullanılmaktadır.

TDI (in): *Test Verisi Girişi (Test Data Input)*, TAP işlemi süresince seri olarak test verilerini ve test komutlarını aygıtın içine kaydırmakta kullanılmaktadır.

TDO (out): *Test Çıkışı (Test Data Output)*, TAP işlemi süresince seri olarak test verilerini ve test komutlarını aygıtın dışına kaydırmakta kullanılmaktadır.

TMS (in): *Test Modu Seçimi (Test Mode Select)*, aygıt içerisindeki TAP denetleyicisinin durumunu kontrol etmede kullanılmaktadır.

TRST#: *Test Reset*, TAP denetleyicisinin asenkron başlatılmasını sağlamaktadır. Bu sinyal, IEEE 1149.1’de seçime bağlı bir sinyaldir.

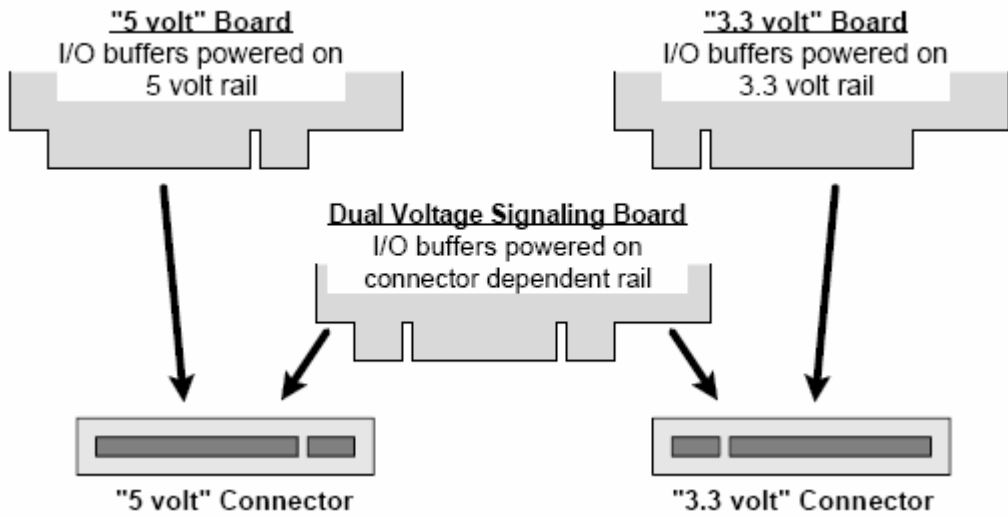
Sistem satıcıları sistem içerisinde gereken 1149.1 seri zinciri (halka – “Ring”) işlemi ve dizaynından sorumludur. Tipik olarak bir 1149.1 halkası, aygıtların seri zincirini oluşturmak için bir aygıtın TDO pininin diğer aygıtın TDI pinine bağlanması ile oluşturulur. Bu uygulamada aygıtlar aynı TCK, TMS ve TRST# (opsiyonel) sinyallerini alırlar.

IEEE 1149.1 standardını desteklemeyen genişlemek artları TDI ve TDO pinlerini donanımsal olarak kabloyla birbirine bağlamalıdır.

2.3. Elektrik Şartnamesi

2.3.1 5V'dan 3.3V'a Geçiş Yol Haritası

PCI elektrik şartnamesinin bir amacı 5V'dan 3.3V'luk bileşen teknolojisine kolay ve hızlı bir geçiş sağlamaktır. Bu geçişi kolaylaştırmak için PCI biri 5V diğeri 3.3V sinyalleşme ortamı için iki adet genişleme kartı konektörü tanımlamıştır. Konektörlerde bulunan çentikler sayesinde kartların yanlış konektöre takılması önlenmiştir. Buna karşın bu iki sinyalleşme ortamını destekleyen genel (Üniversal board) kartlar geliştirilmiştir. Bu kartlar sinyalleşme ortamını algılayabilme kabiliyetine sahiptirler. Bu sayede her iki konektöre de takılabilmektedirler. Şekil 2.4'de üç çeşit genişleme kartı ve sinyalleşme ortamı konektörleri görülmektedir.



Şekil 2.4 PCI Kart Konektörleri

2.3.2 5V Sinyalleşme Ortamı

2.3.2.1 DC Şartnameler

Aşağıda tablo 2.1'de 5V sinyalleşme için DC şartnameleri özetlemektedir.

Symbol	Parameter	Condition	Min	Max	Units	Notes
V_{cc}	Supply Voltage		4.75	5.25	V	
V_{ih}	Input High Voltage		2.0	$V_{cc}+0.5$	V	
V_{il}	Input Low Voltage		-0.5	0.8	V	
I_{ih}	Input High Leakage Current	$V_{in} = 2.7$		70	μA	1
I_{il}	Input Low Leakage Current	$V_{in} = 0.5$		-70	μA	1
V_{oh}	Output High Voltage	$I_{out} = -2 \text{ mA}$	2.4		V	
V_{ol}	Output Low Voltage	$I_{out} = 3 \text{ mA}, 6 \text{ mA}$		0.55	V	2
C_{in}	Input Pin Capacitance			10	pF	3
C_{clk}	CLK Pin Capacitance		5	12	pF	
C_{IDSEL}	IDSEL Pin Capacitance			8	pF	4
L_{pin}	Pin Inductance			20	nH	5
I_{Off}	PME# input leakage	$V_o \leq 5.25 \text{ V}$ V_{cc} off or floating	-	1	μA	6

Tablo 2.1 5V DC Şartname

2.3.2.2 AC Şartnameleri

Aşağıda tablo 2.2’de 5V sinyalleşme için AC şartnameleri özetlemektedir.

Symbol	Parameter	Condition	Min	Max	Units	Notes
$I_{oh(AC)}$	Switching	$0 < V_{out} \leq 1.4$	-44		mA	1
	Current High	$1.4 < V_{out} < 2.4$	$-44 + (V_{out} - 1.4) / 0.024$		mA	1, 2
		$3.1 < V_{out} < V_{cc}$			Eq't'n A	1, 3
	(Test Point)	$V_{out} = 3.1$		-142	mA	3
$I_{ol(AC)}$	Switching	$V_{out} \geq 2.2$	95		mA	1
	Current Low	$2.2 > V_{out} > 0.55$	$V_{out} / 0.023$		mA	1
		$0.71 > V_{out} > 0$			Eq't'n B	1, 3
	(Test Point)	$V_{out} = 0.71$		206	mA	3
I_{cl}	Low Clamp Current	$-5 < V_{in} \leq -1$	$-25 + (V_{in} + 1) / 0.015$		mA	
$slew_r$	Output Rise Slew Rate	0.4V to 2.4V load	1	5	V / ns	4
$slew_f$	Output Fall Slew Rate	2.4V to 0.4V load	1	5	V / ns	4

Tablo 2.2 5V AC Şartname

2.3.3 3.3V Sinyalleşme Ortamı

2.3.3.1 DC Şartnameleri

Aşağıda tablo 2.3’de 3.3V sinyalleşme için DC şartnameleri özetlemektedir.

Symbol	Parameter	Condition	Min	Max	Units	Notes
V_{CC}	Supply Voltage		3.0	3.6	V	
V_{ih}	Input High Voltage		$0.5V_{CC}$	$V_{CC} + 0.5$	V	
V_{il}	Input Low Voltage		-0.5	$0.3V_{CC}$	V	
V_{ipu}	Input Pull-up Voltage		$0.7V_{CC}$		V	1
I_{il}	Input Leakage Current	$0 < V_{in} < V_{CC}$		± 10	μA	2
V_{oh}	Output High Voltage	$I_{out} = -500 \mu A$	$0.9V_{CC}$		V	
V_{ol}	Output Low Voltage	$I_{out} = 1500 \mu A$		$0.1V_{CC}$	V	
C_{in}	Input Pin Capacitance			10	pF	3
C_{clk}	CLK Pin Capacitance		5	12	pF	
C_{IDSEL}	IDSEL Pin Capacitance			8	pF	4
L_{pin}	Pin Inductance			20	nH	5
I_{Off}	PME# input leakage	$V_O \leq 3.6 V$ V_{CC} off or floating	-	1	μA	6

Tablo 2.3 3.3V DC Şartname

2.3.3.2 AC Şartnameleri

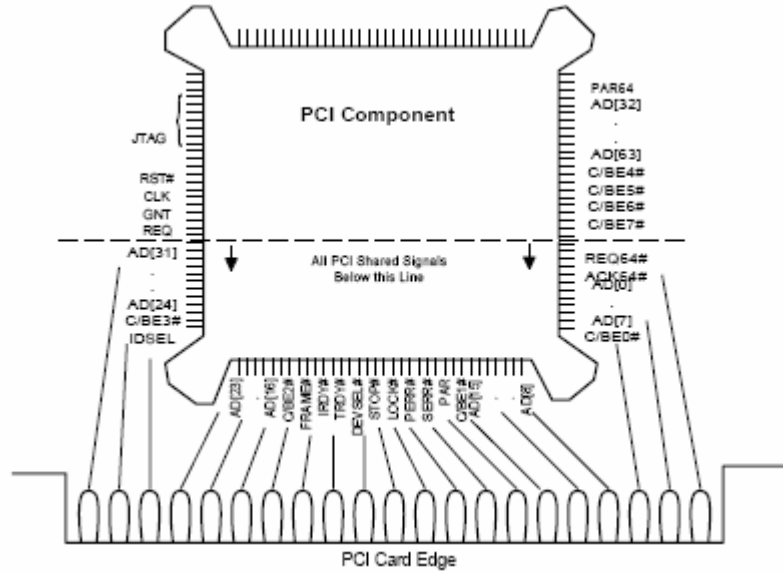
Aşağıda tablo 2.4’de 3.3V sinyalleşme için AC şartnameleri özetlemektedir.

Symbol	Parameter	Condition	Min	Max	Units	Notes
$I_{oh(AC)}$	Switching	$0 < V_{out} \leq 0.3V_{CC}$	$-12V_{CC}$		mA	1
	Current High	$0.3V_{CC} < V_{out} < 0.8V_{CC}$	$-17.1(V_{CC} - V_{out})$		mA	1
		$0.7V_{CC} < V_{out} < V_{CC}$		Eq'n C		1, 2
	(Test Point)	$V_{out} = 0.7V_{CC}$		$-32V_{CC}$	mA	2
$I_{ol(AC)}$	Switching	$V_{CC} > V_{out} \geq 0.6V_{CC}$	$16V_{CC}$		mA	1
	Current Low	$0.6V_{CC} > V_{out} > 0.1V_{CC}$	$26.7V_{out}$		mA	1
		$0.18V_{CC} > V_{out} > 0$		Eq'n D		1, 2
	(Test Point)	$V_{out} = 0.18V_{CC}$		$38V_{CC}$	mA	2
I_{cl}	Low Clamp Current	$-3 < V_{in} \leq -1$	$-25 + (V_{in} + 1) \cdot 0.015$		mA	
I_{ch}	High Clamp Current	$V_{CC} + 4 > V_{in} \geq V_{CC} + 1$	$25 + (V_{in} - V_{CC} - 1) \cdot 0.015$		mA	
$slew_r$	Output Rise Slew Rate	$0.2V_{CC} - 0.6V_{CC}$ load	1	4	V/ns	3
$slew_f$	Output Fall Slew Rate	$0.6V_{CC} - 0.2V_{CC}$ load	1	4	V/ns	3

Tablo 2.4 3.3V AC Şartname

2.3.4 Pin Yerleşim Planı (Pinout) Önerisi

Bu bölümde önerilen PCI bileşenlerinin pin bağlantısı planını gösterilmektedir. Aşağıda şekil 2.5’de bir PQFP PCI bileşeninin önerilen pin bağlantı şekli görülmektedir. Burada kart ile konektörün sinyal sıralamasının tam olarak birbirine eşlendiğine dikkat ediniz. Yerleştirme ile güç ve toprak pinleri sayısı aygıtı bağlıdır. Ek sinyaller, 64-bit veri yolu versiyonunda bileşen üzerinde saat yönünün tersine şekilde sıralanacak şekilde olması gerekmektedir.



Şekil 2.5 Önerilen Pin Yerleşim Planı

2.3.5 Genişleme Kartı Şartnamesi

2.3.5.1 Kart Pin Tahsisi

PCI konektörü, PCI bileşenleri için tanımlanmış tüm sinyalleri ve ayrıca PRSNT1# ve PRSNT2# olmak üzere konektöre bağlı olarak iki adet ek pini barındırmaktadır. Bu ek pinler iki amaç için kullanılmaktadır; slota fiziksel olarak bir genişleme kartının bağlı olduğunu belirtmek ve kartın toplam güç gereksinimi hakkında bilgi sağlamak. Aşağıda tablo 2.5’de genişleme kartları için istenen PRSNT# pin ayarını belirtmektedir. PCI genişleme kartı pin yerleşimi tablo 2.6’da görülmektedir.

PRSENT1#	PRSENT2#	Genişleme Şekli
Açık	Açık	Genişleme kartı mevcut değil
Toprak	Açık	Genişleme kartı mevcut. En fazla 25W
Açık	Toprak	Genişleme kartı mevcut. En fazla 15W
Toprak	Toprak	Genişleme kartı mevcut. En fazla 7.5W

Tablo 2.5 Present Sinyali Tanımlaması

Pin	5V System Environment		3.3V System Environment		Comments
	Side B	Side A	Side B	Side A	
1	-12V	TRST#	-12V	TRST#	32-bit connector start
2	TCK	+12V	TCK	+12V	
3	Ground	TMS	Ground	TMS	
4	TDO	TDI	TDO	TDI	
5	+5V	+5V	+5V	+5V	
6	+5V	INTA#	+5V	INTA#	
7	INTB#	INTC#	INTB#	INTC#	
8	INTD#	+5V	INTD#	+5V	
9	PRSENT1#	Reserved	PRSENT1#	Reserved	
10	Reserved	+5V (I/O)	Reserved	+3.3V (I/O)	
11	PRSENT2#	Reserved	PRSENT2#	Reserved	
12	Ground	Ground	CONNECTOR KEY		
13	Ground	Ground	CONNECTOR KEY		
14	Reserved	3.3Vaux	Reserved	3.3Vaux	3.3 volt key
15	Ground	RST#	Ground	RST#	
16	CLK	+5V (I/O)	CLK	+3.3V (I/O)	
17	Ground	GNT#	Ground	GNT#	
18	REQ#	Ground	REQ#	Ground	
19	+5V (I/O)	PME#	+3.3V (I/O)	PME#	
20	AD[31]	AD[30]	AD[31]	AD[30]	
21	AD[29]	+3.3V	AD[29]	+3.3V	
22	Ground	AD[28]	Ground	AD[28]	
23	AD[27]	AD[26]	AD[27]	AD[26]	
24	AD[25]	Ground	AD[25]	Ground	
25	+3.3V	AD[24]	+3.3V	AD[24]	
26	C/BE[3]#	IDSEL	C/BE[3]#	IDSEL	
27	AD[23]	+3.3V	AD[23]	+3.3V	
28	Ground	AD[22]	Ground	AD[22]	
29	AD[21]	AD[20]	AD[21]	AD[20]	
30	AD[19]	Ground	AD[19]	Ground	
31	+3.3V	AD[18]	+3.3V	AD[18]	
32	AD[17]	AD[16]	AD[17]	AD[16]	
33	C/BE[2]#	+3.3V	C/BE[2]#	+3.3V	
34	Ground	FRAME#	Ground	FRAME#	
35	IRDY#	Ground	IRDY#	Ground	
36	+3.3V	TRDY#	+3.3V	TRDY#	
37	DEVSEL#	Ground	DEVSEL#	Ground	
38	Ground	STOP#	Ground	STOP#	
39	LOCK#	+3.3V	LOCK#	+3.3V	
40	PERR#	Reserved*	PERR#	Reserved*	
41	+3.3V	Reserved*	+3.3V	Reserved*	
42	SERR#	Ground	SERR#	Ground	

Tablo 2.6 PCI Genişleme Kartı Pin Yerleşim Planı

Pin	5V System Environment		3.3V System Environment		Comments
	Side B	Side A	Side B	Side A	
43	+3.3V	PAR	+3.3V	PAR	66 MHz / gnd 5 volt key 5 volt key 32-bit connector end
44	C/BE[1]#	AD[15]	C/BE[1]#	AD[15]	
45	AD[14]	+3.3V	AD[14]	+3.3V	
46	Ground	AD[13]	Ground	AD[13]	
47	AD[12]	AD[11]	AD[12]	AD[11]	
48	AD[10]	Ground	AD[10]	Ground	
49	Ground	AD[09]	M66EN	AD[09]	
50	CONNECTOR KEY		Ground	Ground	
51	CONNECTOR KEY		Ground	Ground	
52	AD[08]	C/BE[0]#	AD[08]	C/BE[0]#	
53	AD[07]	+3.3V	AD[07]	+3.3V	
54	+3.3V	AD[06]	+3.3V	AD[06]	
55	AD[05]	AD[04]	AD[05]	AD[04]	
56	AD[03]	Ground	AD[03]	Ground	
57	Ground	AD[02]	Ground	AD[02]	
58	AD[01]	AD[00]	AD[01]	AD[00]	
59	+5V (I/O)	+5V (I/O)	+3.3V (I/O)	+3.3V (I/O)	
60	ACK64#	REQ64#	ACK64#	REQ64#	
61	+5V	+5V	+5V	+5V	
62	+5V	+5V	+5V	+5V	
	CONNECTOR KEY		CONNECTOR KEY		64-bit spacer
	CONNECTOR KEY		CONNECTOR KEY		64-bit spacer
63	Reserved	Ground	Reserved	Ground	64-bit connector start
64	Ground	C/BE[7]#	Ground	C/BE[7]#	
65	C/BE[6]#	C/BE[5]#	C/BE[6]#	C/BE[5]#	
66	C/BE[4]#	+5V (I/O)	C/BE[4]#	+3.3V (I/O)	
67	Ground	PAR64	Ground	PAR64	
68	AD[63]	AD[62]	AD[63]	AD[62]	
69	AD[61]	Ground	AD[61]	Ground	
70	+5V (I/O)	AD[60]	+3.3V (I/O)	AD[60]	
71	AD[59]	AD[58]	AD[59]	AD[58]	
72	AD[57]	Ground	AD[57]	Ground	
73	Ground	AD[56]	Ground	AD[56]	
74	AD[55]	AD[54]	AD[55]	AD[54]	
75	AD[53]	+5V (I/O)	AD[53]	+3.3V (I/O)	
76	Ground	AD[52]	Ground	AD[52]	
77	AD[51]	AD[50]	AD[51]	AD[50]	
78	AD[49]	Ground	AD[49]	Ground	
79	+5V (I/O)	AD[48]	+3.3V (I/O)	AD[48]	
80	AD[47]	AD[46]	AD[47]	AD[46]	
81	AD[45]	Ground	AD[45]	Ground	
82	Ground	AD[44]	Ground	AD[44]	

Tablo 2.6 PCI Genişleme Kartı Pin Yerleşim Planı (devam)

Pin	5V System Environment		3.3V System Environment		Comments
	Side B	Side A	Side B	Side A	
83	AD[43]	AD[42]	AD[43]	AD[42]	
84	AD[41]	+5V (I/O)	AD[41]	+3.3V (I/O)	
85	Ground	AD[40]	Ground	AD[40]	
86	AD[39]	AD[38]	AD[39]	AD[38]	
87	AD[37]	Ground	AD[37]	Ground	
88	+5V (I/O)	AD[36]	+3.3V (I/O)	AD[36]	
89	AD[35]	AD[34]	AD[35]	AD[34]	
90	AD[33]	Ground	AD[33]	Ground	
91	Ground	AD[32]	Ground	AD[32]	
92	Reserved	Reserved	Reserved	Reserved	
93	Reserved	Ground	Reserved	Ground	
94	Ground	Reserved	Ground	Reserved	

Tablo 2.6 PCI Genişleme Kartı Pin Yerleşim Planı (devam)

Pin Type	5V Board	Universal Board	3.3V Board
Ground	22	18 (Note)	22 (Note)
+5 V	13	8	8
+3.3 V	12	12	17
I/O pwr	0	5	0
Reserv'd	4	4	4

Tablo 2.732-bit Genişleme Kartı Pin Özeti

Not: M66EN pini gerçekleştirildiğinde toplam toprak pini universal kart için 17, 3.3V kartlar için 21'dir.

Pin Type	5V Board	Universal Board	3.3V Board
Ground	16	16	16
+5 V	6	0	0
+3.3 V	0	0	6
I/O pwr	0	6	0
Reserv'd	5	5	5

Tablo 2.8 64-bit Genişleme Kartı Pin Özeti

2.3.6 Güç Gereksinimleri

2.3.6.1 Dekuplajlama

Tipik koşullar altında Vcc düzeyinden toprak düzeyine olan kapasitans Vcc konektör pinleri için yeterli dekuplajlama sağlamaktadır. Bağlantı noktasından Vcc/toprak düzlemine olan maksimum yol uzunluğu 0.25 inç (yol genişliğini 20 mil varsayarsak) olmalıdır.

Bununla beraber, üniversal kartlarda G/Ç tampon güç rayı gerekli dekuplajlamayı sağlamak için toprak düzlemine uygun kapasitansa sahip olması olasıdır. +V_{IO} olarak etiketlenmiş pinler, pin başına ortalama 0.0047uF'lık kapasite ile toprağa dekupe edilmelidir.

Ek olarak, PCI konektör ucundaki tüm +3.3V pinleri ve kullanılmayan +5V ve +V_{IO} pinleri AC dönüş yolu barındırırlar ve aşağıda belirtilen şekilde verimli AC referans noktaları olarak fonksiyona devam ettiğinden emin olmak için genişlemek kartı üzerinde toprak düzlemine kuplajlanmalıdır.

1. Dekuplajlama, her Vcc pini başına ortalama en az 0.01uF (yüksek hızlı) olmalıdır.
2. Pin noktasından kapasite bağlantı noktasına olan yol uzunluğu en az 0.02 inç genişliğinde ve en fazla 0.25 inç uzunluğunda olmalıdır.
3. 1 ve 2'deki gereksinimleri karşılayan, aynı kapasiteyi paylaşan pin sayısında herhangi bir sınırlama yoktur.

2.3.6.2 Güç Tüketimi

Her hangi bir PCI kartı için izin verilen maksimum güç 25 Watt'dır ve konektörden sağlanan tüm güç yollarından (+5V, +3.3V, +V_{IO}, +12V, -12V, +3.3V_{AUX}) çekilen toplam gücü temsil etmektedir.

2.3.6.3 Fiziksel Gereksinimler

2.3.6.3.1 Yol Uzunluęu Sınırları

Geniřleme kartının konektör ucunun üstünden PCI aygıtına olan yol uzunlukları ařaęıdaki gibidir:

- 32-bit ve 64-bit geniřleme kartları için tüm 32-bit arayüz sinyallerinin maksimum yol uzunluęu 1.5 inç ile sınırlandırılmıřtır. Bu sistem pinleri, kesme pinleri ve JTAG pinleri dıřındaki tüm sinyal gruplarını kapsamaktadır.
- 64-bit sistemlerde kullanılan ek sinyallerin yol uzunlukları tüm 64-bit geniřleme kartlarında 2 inç ile sınırlandırılmıřtır.
- 32-bit ve 64-bit geniřleme kartları için PCI CLK sinyalinin yol uzunluęu 2.5 inç (± 0.1 inç)'dir ve sadece bir yüke yönlendirilmelidir.

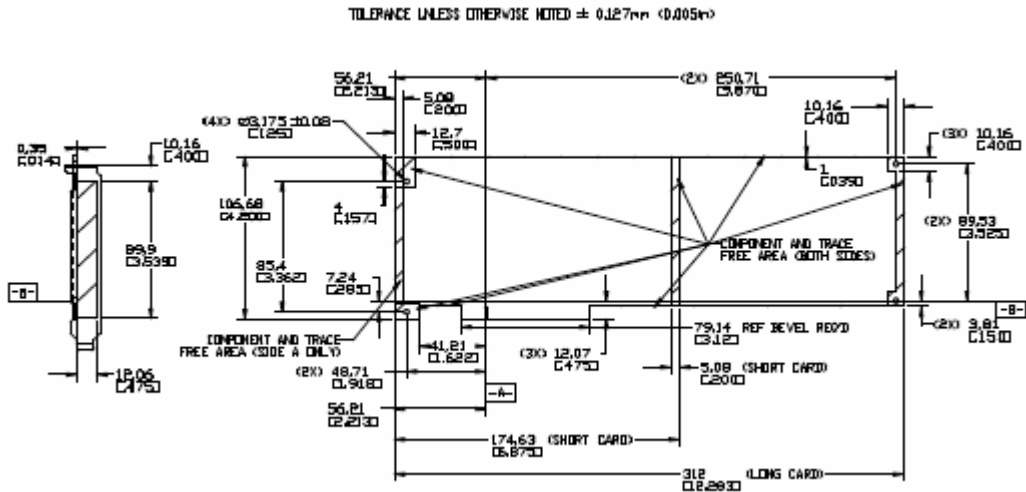
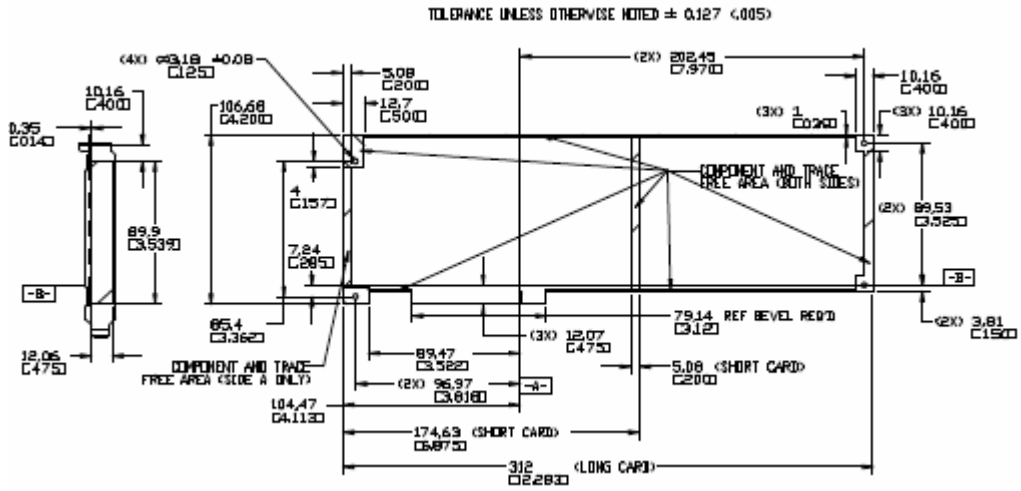
2.4 Mekanik řartnamesi

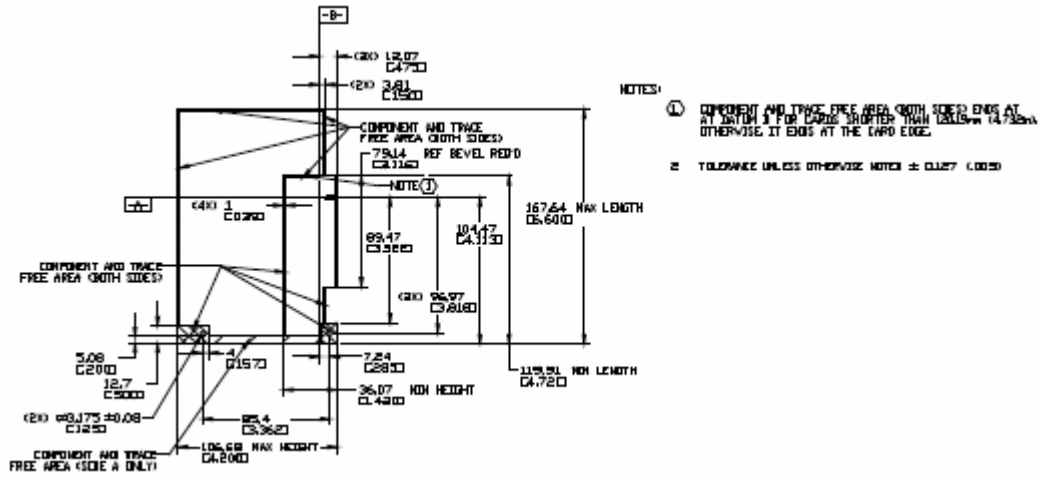
PCI geniřleme kartları mevcut řasi tasarımları içinde kolayca gerçeklenen raw kart tasarımını esas almaktadır. PCI geniřleme kartları iki temel biçime sahiptir: standart ve kısa uzunluklu. Standart uzunluktaki kart 49 inç kare boyutlarındadır. Sabit ve deęiřken yükseklikli kısa kartlar düşük maliyet saęlayan panel optimizasyonu için seçilmiřtir ve aynı zamanda gerçekleřtirilmesi en ucuz olan en az enerji harcayan küçük sistem tasarımlarına olanak saęlamaktadır. PCI geniřleme kartı ara baęlantısı 43-bit ve 64-bit arayüzlerin her ikisi için tanımlanmıřtır.

PCI kartları ve konektörleri 5V ve 3.3V iřlemlerini yönetmek için ayarlanmıřtır. Temel 32-bit konektör 120 pinden oluřmaktadır. Mantıksal pin sayısı 124 pin ile gösterilmektedir. Fakat 4 pin anahtarlama için kullanılmaktadır. 64-bit geniřleme konektörü 32 bit konektöre 64 pin eklenerek toplam 184 pinden oluřmaktadır. 32-bitlik bir kart 64-bitlik bir konektör üzerinde kendini 32-bit transfer için tanımlamaktadır. 32-bitlik bir konektör üzerindeki 64-bitlik bir kart 32-bit transfer sistemi için yapılandırılmalıdır.

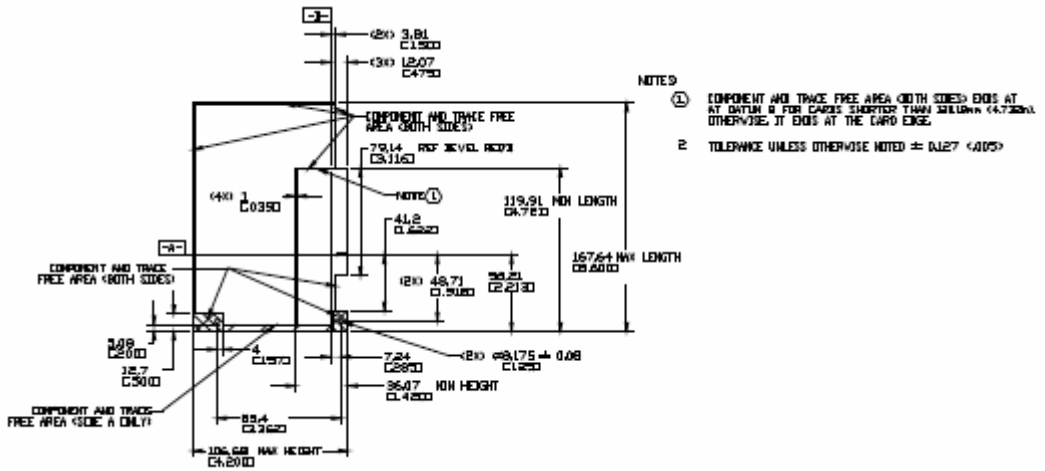
2.4.1 Genişleme Kartı Fiziksel Boyutları ve Toleransı

PCI genişleme kartının birincil eleman yüzündeki maksimum eleman yüksekliği 0.57 inç (14.48 mm)'dir. Kartın arka yüzündeki maksimum eleman yüksekliği ise 0.105 inç (2.67 mm)'yi geçmemelidir. Aşağıda şekil 2.6'dan şekil 2.19'e kadar çeşitli genişleme kartlarının fiziksel boyutları görülmektedir.

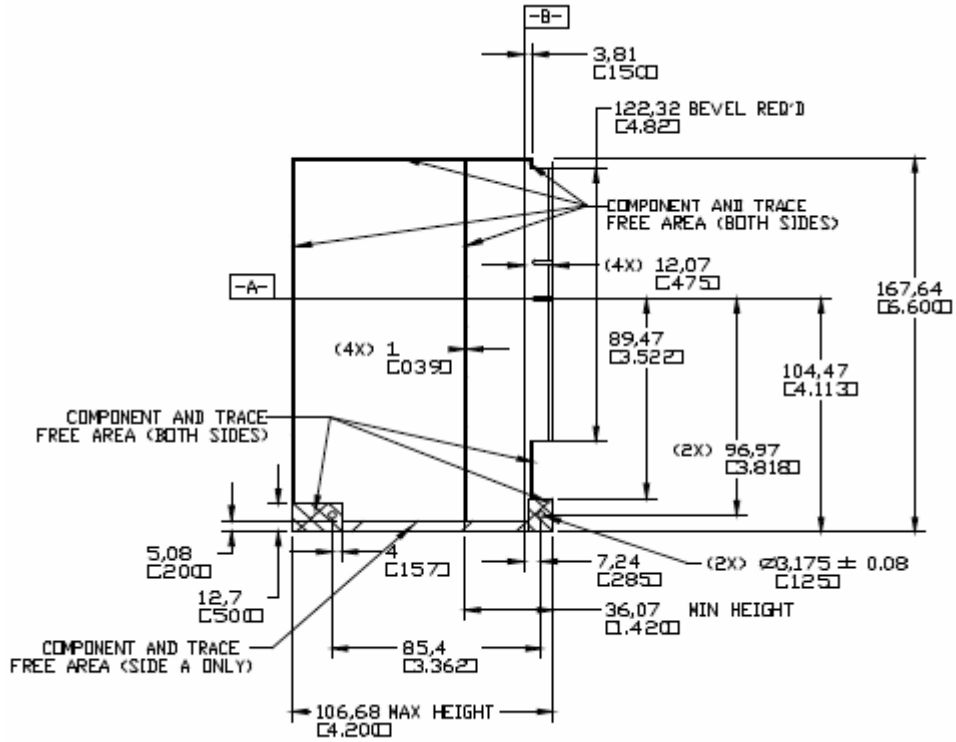




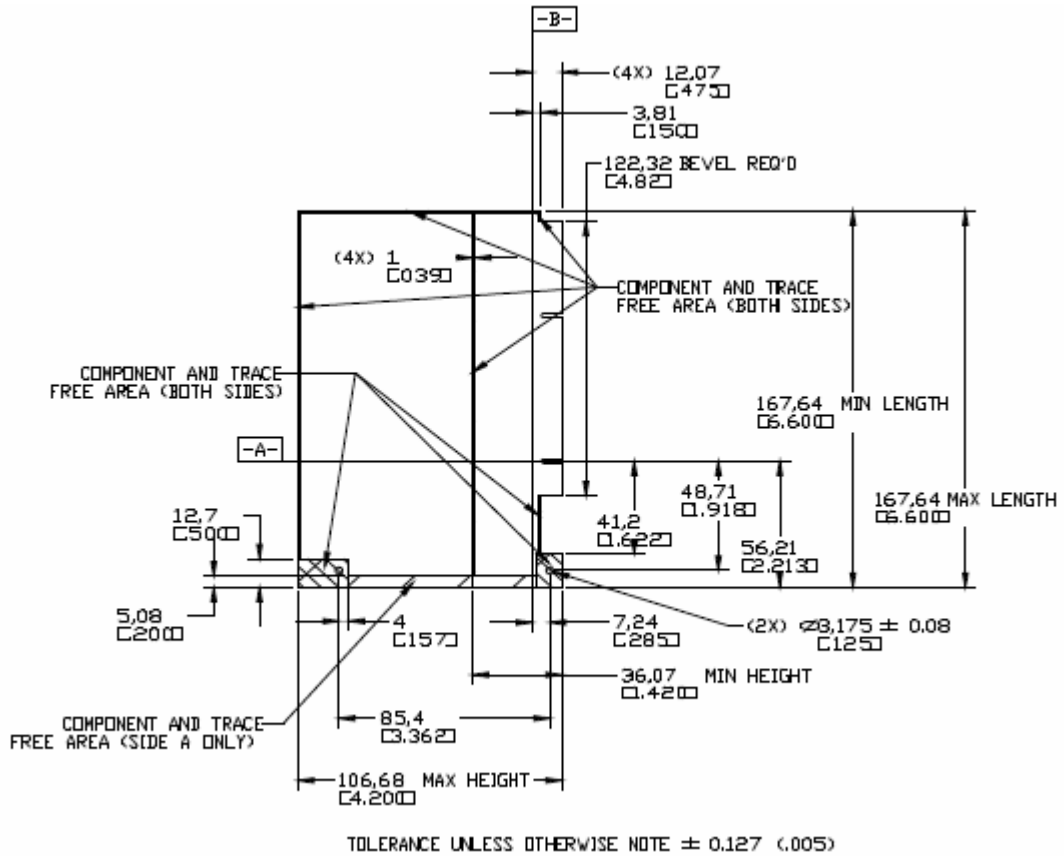
Şekil 2.8 PCI Raw Değişken Yükseklikli Kısa Kart (5V 32-bit)



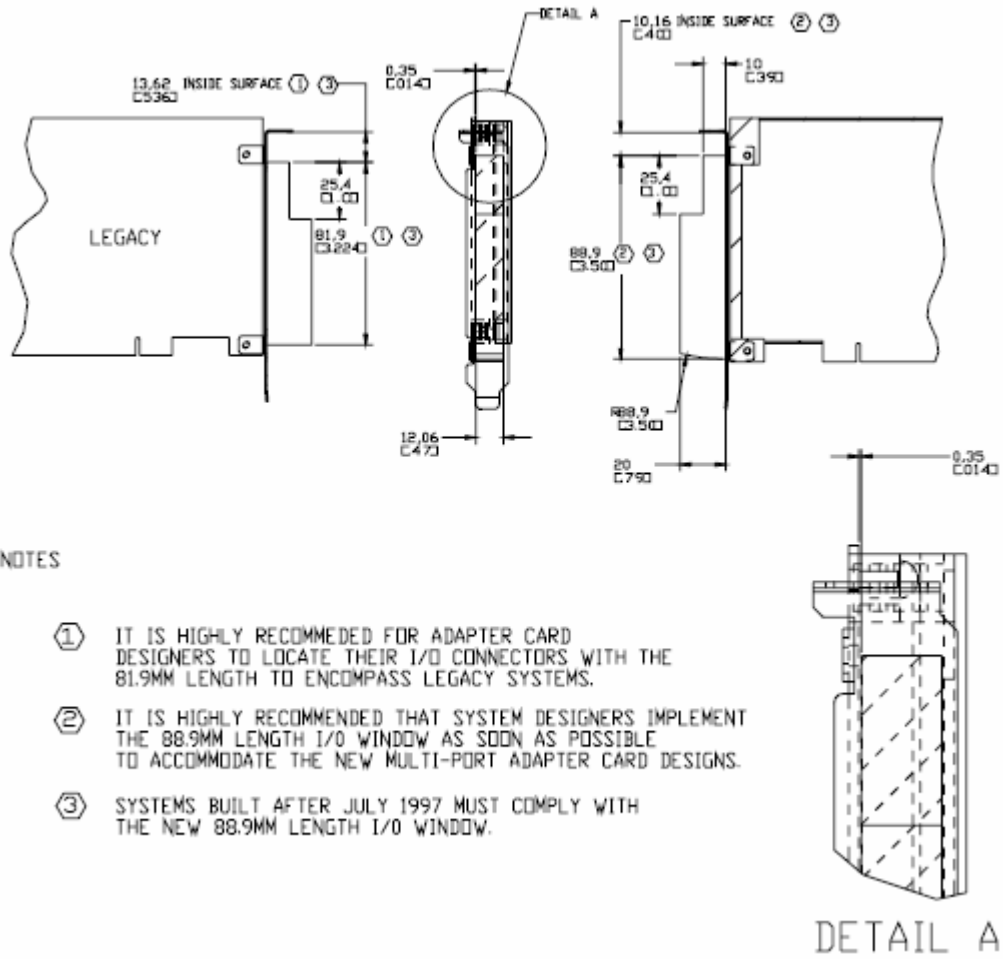
Şekil 2.9 PCI Raw Değişken Yükseklikli Kısa Kart (3.3V 32-bit)



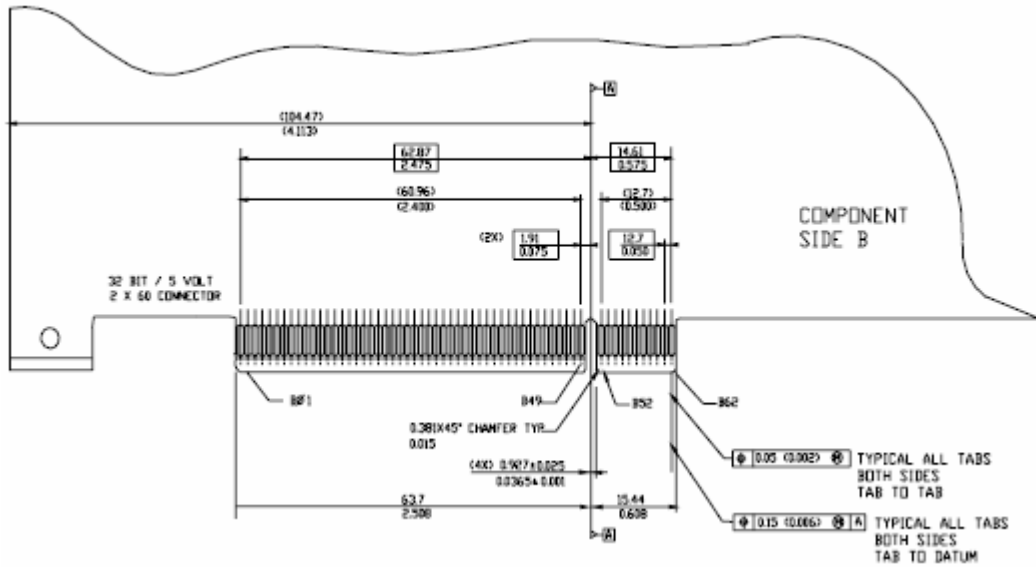
TOLERANCE UNLESS OTHERWISE NOTED $\pm 0.127 (0.005)$
 Şekil 2.10 PCI Raw Değişken Yükseklikli Kısa Kart (5V 64-bit)



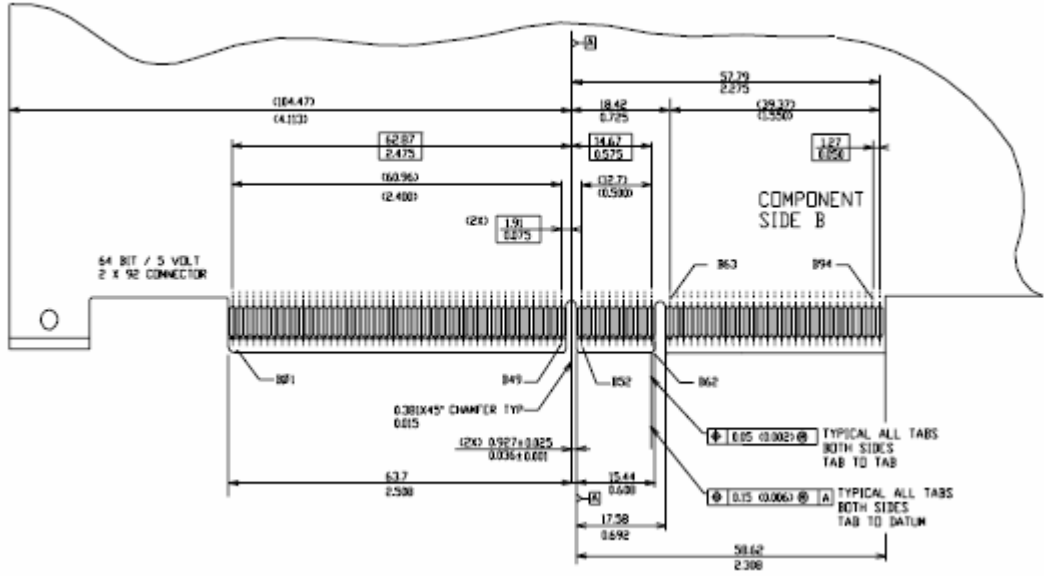
TOLERANCE UNLESS OTHERWISE NOTE $\pm 0.127 (0.005)$
 Şekil 2.11 PCI Raw Değişken Yükseklikli Kısa Kart (3.3V 64-bit)



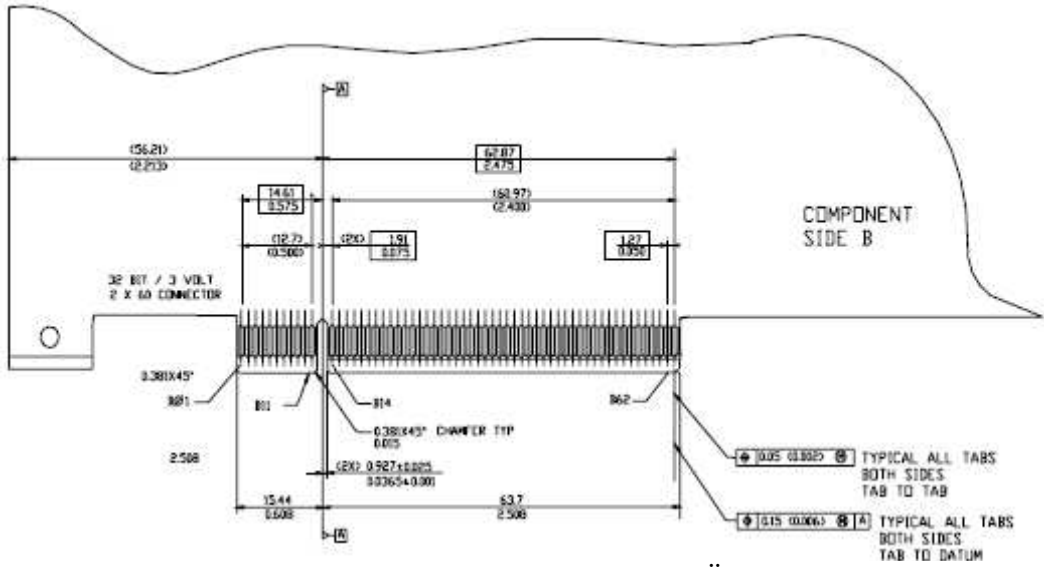
Şekil 2.12 G/Ç Pencere Yüksekliği



Şekil 2.13 5V 32-bit Kart Konektör Kenarı Ölçüleri ve Toleransı



Şekil 2.14 5V 64-bit Kart Konektör Kenarı Ölçüleri ve Toleransı



Şekil 2.15 3.3V 32-bit Kart Konektör Kenarı Ölçüleri ve Toleransı

3. Spartan-3E FPGA AİLESİ

Spartan-3E Alan Programlamalı Kapı Dizisi ailesi 5 üyeden oluşmaktadır (Bkz. Tablo 3.1). Bu beş üye toplam 100,000 'den 1.6 milyon'a kadar sistem kapı yoğunluğuna sahiptir.

Spartan-3E ailesi I/O başına lojik kapı miktarının arttırılması ve lojik hücre başına olan maliyetin önemli ölçüde düşürülmesi ile daha önceki Spartan-3 ailesinin başarısı üzerine geliştirilmiştir. Yeni özellikler sistem performansını arttırmakta ve yapılandırma maliyetini azaltmaktadır. Spartan-3E'deki bu gelişimler 90nm işlem teknolojisi ile birleştirilerek daha fazla işlevsellik ve bant-genişliği getirmekte ve programlanabilir lojik endüstrisinde yeni standartlar sunmaktadır.

Fevkalade düşük maliyeti yüzünden Spartan-3E ailesi geniş-bant erişim, ev ağı, display/projeksiyon ve sayısal televizyon donanımları dâhil geniş bir tüketici elektroniği uygulamalarına uygundur.

Özellikleri:

- Yüksek seviyeli tüketici-yönelimli uygulamalar için çok düşük maliyetli yüksek performanslı lojik çözümü
- Kanıtlanmış gelişmiş 90nm işlem teknolojisi
- Çoklu-Gerilim, Çoklu-Standart SelectIO™ arayüz pinleri
 - 376'ya kadar I/O pini yâda 156 fark sinyali çifti
 - LVCMOS, LVTTTL, HSTL ve SSTL single-ended sinyal standartları
 - 3.3V, 2.5V, 1.8V, 1.5V ve 1.2V işaretleşme
 - I/O başına 622+ Mb/s veri transfer oranı
 - True LVDS, RSQS, mini-LVDS, differential HSTL/SSTL differential I/O
 - Gelişmiş DDR (double data rate) desteği
 - 333 Mb/s'a kadar DDR SDRAM desteği
- Bol ve esnek lojik kaynaklar

- Opsiyonel kaymalı kaydedici yâda dağıtılmış RAM desteği dâhil 33,192'e kadar lojik hücre yoğunluğu
- Verimli geniş çarpıcılar, geniş lojik
- Opsiyonel işhatlı gelişmiş 18x18 çarpıcılar
- Hiyerarşik SelectRAM™ bellek mimarisi
 - 648Kbit'e kadar hızlı blok RAM
 - 231 Kbit'e kadar verimli dağıtılmış RAM
- 8 adede kadar Sayısal Saat Yöneticisi (DCM)
 - Saat eğim eleme (gecikme kilitli çevrim DLL)
 - Frekans sentezi, çarpma, bölme
 - Yüksek çözünürlüklü faz kaydırıcı
 - Geniş frekans aralığı (5 MHz'den 300 MHz'in üzerine kadar)
- Sekiz global saat ile aygıtın her yarım bölümü başına sekiz ek saat artı bol düşük-eğimli yönlendirme
- Endüstri-standardı PROM'lar için yapılandırma arayüzü
 - Düşük-maliyetli, yerden tasarruf sağlayan SPI seri Flash PROM
 - X8 yâda x8/x16 paralel NOR Flash PROM
 - Düşük maliyetli JTAG destekli Xilinx Platform Flash
- MicroBlaze ve PicoBlaze gömülü işlemce çekirdekleri
- Tamamen uyumlu 32bit/64bit 33MHz PCI desteği (bazı aygıtlarda 66MHz)

Device	System Gates	Equivalent Logic Cells	CLB Array (One CLB = Four Slices)				Distributed RAM bits ⁽¹⁾	Block RAM bits ⁽¹⁾	Dedicated Multipliers	DCMs	Maximum User I/O	Maximum Differential I/O Pairs
			Rows	Columns	Total CLBs	Total Slices						
XC3S100E	100K	2,160	22	16	240	960	15K	72K	4	2	108	40
XC3S250E	250K	5,508	34	26	612	2,448	38K	216K	12	4	172	68
XC3S500E	500K	10,476	46	34	1,164	4,656	73K	360K	20	4	232	92
XC3S1200E	1200K	19,512	60	46	2,168	8,672	136K	504K	28	8	304	124
XC3S1600E	1600K	33,192	76	58	3,688	14,752	231K	648K	36	8	376	156

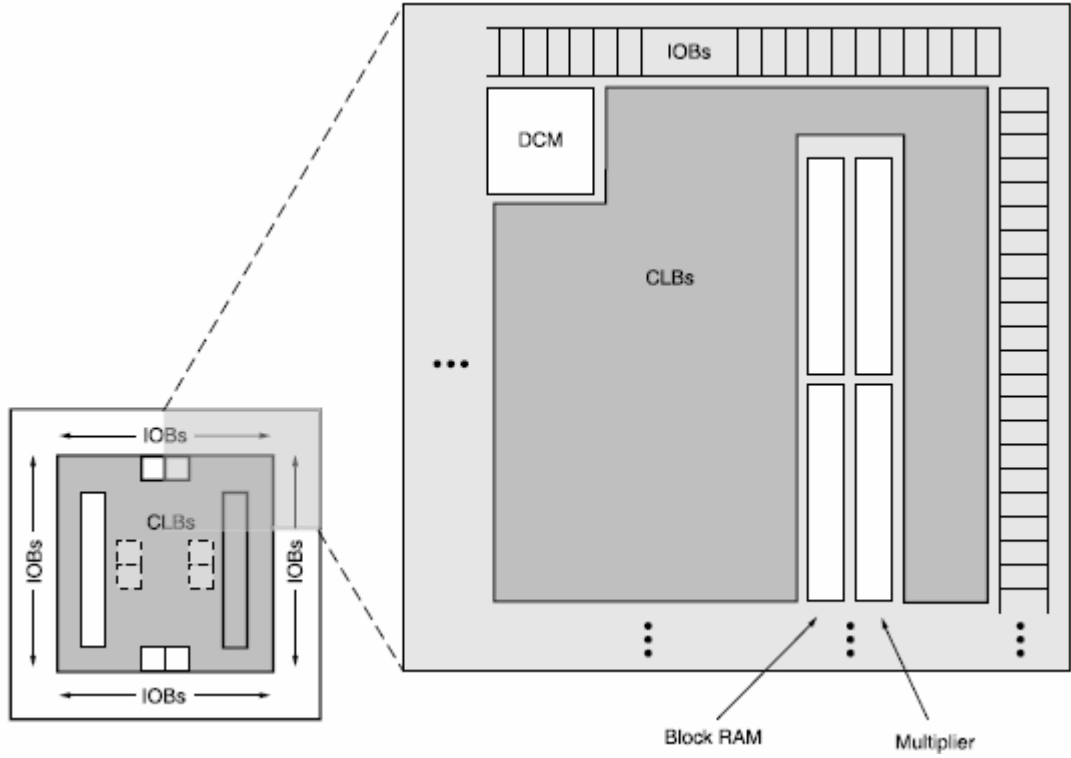
Tablo 3.1 Spartan-3E FPGA ailesi üyeleri

3.1 Genel Özellikleri

Spartan-3E ailesinin mimarisi beş temel programlanabilir fonksiyonel elemandan meydana gelmiştir:

- **Düzenlenebilir Lojik Bloklar (CLBs)**, flip-flop yâda mandal olarak kullanılan ek lojik depolama elemanların gerçekleştiren esnek Look-Up Tablolarını (LUTs) içermektedir.
- **Giriş/Çıkış Blokları (IOBs)** dâhili lojik aygıtlar ile I/O pinleri arasındaki veri akışını yönetmektedir. Her bir IOB 3-state işlemi dâhil çift yönlü veri akışını desteklemektedir. Dört yüksek performanslı fark sinyali standardı dâhil, birçok sinyal standardını desteklemektedir. DDR saklayıcıları da eklenmiştir.
- **Blok RAM**, 18-Kbit çift-portlu blok biçiminde veri depolama sağlamaktadır.
- **Çarpıcı Blokları**, girişindeki iki adet 18-bitlik sayıyı çarpıp sonucu vermektedir.
- **Sayısal Saat Yöneticisi (DCM)**, yayma, gecikme, çarpma, bölme ve faz-kaymalı saat sinyalleri için kendinden ayarlanabilir tamamen sayısal çözümler barındırmaktadır.

Bu elemanlar şekil 3.1’de görüldüğü gibi düzenlenmektedir. IOB etrafındaki halka düzgün bir CLB dizisidir. XCS100E dışındaki her bir aygıt iki sütun blok RAM’a sahiptir. Her bir blok RAM, bir ayrılmış çarpanla birleştirilmiştir. DCM, aygıtın altında iki ve üstünde iki olmak üzere merkeze yerleştirilmiştir. XC3S1200E ve XC3S1600E orta sol ve sağ tarafta olmak üzere ek iki adet DCM’e sahiptir.



Şekil 3.1 Spartan-3E Ailesi Mimarisi

Spartan-3E ailesi çeşitli kılıflarda üretilmektedir. Aşağıda tablo 3.2’de aygıt ve kullanılan kılıf tipine bağlı olarak mevcut kullanıcı G/Ç sayılarını göstermektedir:

Device	VQ100 VQG100		CP132 CPG132		TQ144 TQG144		PQ208 PQG208		FT256 FTG256		FG320 FGG320		FG400 FGG400		FG484 FGG484	
	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff	User	Diff
XC3S100E	66 (7)	30 (2)	83 (11)	35 (2)	108 (28)	40 (4)	-	-	-	-	-	-	-	-	-	-
XC3S250E	66 (7)	30 (2)	92 (7)	41 (2)	108 (28)	40 (4)	158 (32)	65 (5)	172 (40)	68 (8)	-	-	-	-	-	-
XC3S500E	-	-	92 (7)	41 (2)	-	-	158 (32)	65 (5)	190 (41)	77 (8)	232 (56)	92 (12)	-	-	-	-
XC3S1200E	-	-	-	-	-	-	-	-	190 (40)	77 (8)	250 (56)	99 (12)	304 (72)	124 (20)	-	-
XC3S1600E	-	-	-	-	-	-	-	-	-	-	250 (56)	99 (12)	304 (72)	124 (20)	376 (82)	156 (21)

Tablo 3.2 Mevcut Kullanıcı ve Fark (Diff) G/Ç Sayıları

3.2 Mimari Tanımlama

3.2.1 IOB Yapısı

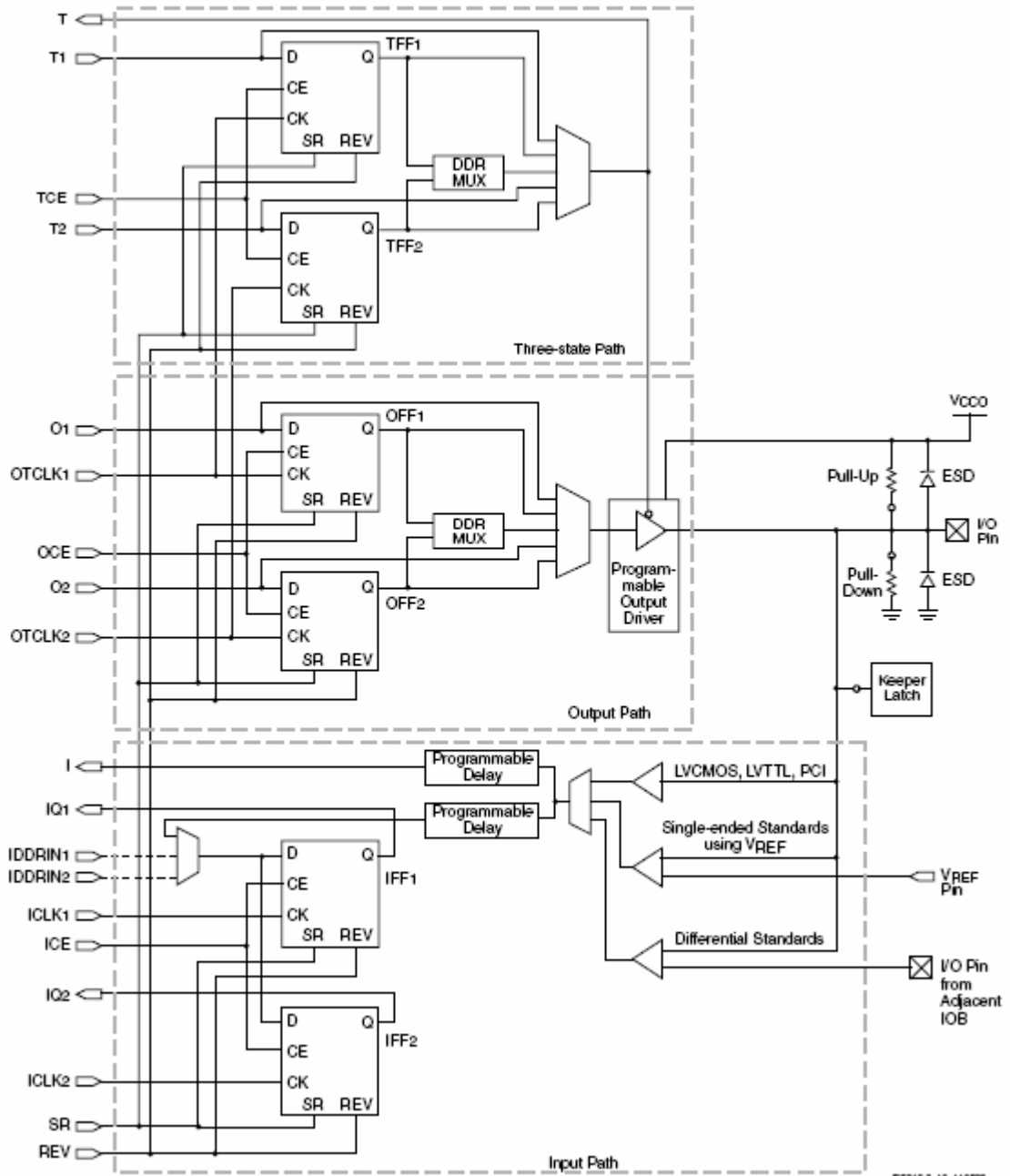
Giriş/Çıkış Bloğu (IOB), FPGA'in dâhili lojiği ile bir kılıf pini arasında programlanabilir tekyönlü yâda çiftyönlü bir arayüz sağlamaktadır.

Tekyönlü sadece-giriş bloğu tüm IOB kapasitesinin bir altkümesine sahiptir. Böylece, çıkış yolu için herhangi bir lojik yâda bağlantı bulunmamaktadır. Sadece-giriş bloklarının sayısı aygıtın boyutuna göre değişmekle beraber asla toplam IOB sayısının % 25'ini aşmamaktadır.

Şekil 3.2 IOB'un içyapısının basitleştirilmiş bir diyagramını göstermektedir. IOB'u içerisinde üç ana sinyal yolu bulunmaktadır: çıkış yolu, giriş yolu ve 3-state yolu. Her bir yol, saklayıcı yâda mandal rolü gören, kendi depolama elemanları çiftlerine sahiptir. Bu üç sinyal yolu aşağıdaki gibidir:

- Giriş yolu, bir kılıf pinine bağlanmış pad'den opsiyonel programlanabilir bir gecikme elemanından geçerek direk olarak I hattına giden veriyi taşımaktadır. Gecikme elemanından sonra IQ1 ve IQ2 hatlarına bir depolama elemanları çiftinden alternatif yollar vardır. IOB çıkışları, I, IQ1 ve IQ2 FPGA'in dâhili lojiğine gitmektedir. Gecikme elemanı, bir sıfır tutma zamanı sağlamak için aktif edilebilir.
- Çıkış yolunda veri, FPGA'in dâhili lojiğinden gelen O1 ve O2 hatlarından başlar ardından bir çarpandan geçer ve sonra bir 3-state sürücüsünden IOB pad'ine ulaşır. Bu direk yola ek olarak çarpıcı, bir depolama elemanları çifti eklemek seçeneği de barındırmaktadır.
- 3-state yolu, çıkış sürücüsünün yüksek empedansta olduğunu belirlemektedir. T1 ve T2 hatları FPGA'in dâhili lojiğinden bir çarpıcı aracılığıyla çıkış sürücüsüne veri taşımaktadır. Bu direk yola ek olarak çarpıcı, bir depolama elemanları çifti eklemek seçeneği de barındırmaktadır.

- Depolama elemanları ile birleştirilmiş olanlarda dâhil IOB'a giren tüm sinyal yolları, bir evirici seçeneğine sahiptir. Bu yola yerleştirilmiş herhangi bir evirici IOB içerisine otomatik olarak çekilecektir.



Şekil 3.2 Spartan-3E Basitleştirilmiş Giriş/Çıkış bloğu (IOB)

3.2.2 SelectIO Sinyal Standartları

Spartan-3E I/O'ları geniş çapta I/O işaretleme standartlarını destekleyen giriş ve çıkışlardan oluşmaktadır (Tablo 3.3 ve Tablo 3.4). I/O'ların çokluğu aynı zamanda

fark işaretleme standartlarından herhangi birini destekleyen fark çiftleri oluşturmak için de kullanılabilir (Tablo 3.4).

Bir tasarım içerisinde I/O işaretleme standardını tanımlamak için IOSTANDARD özelliği uygun değere ayarlanır. Xilinx, maksimum esneklikle IOSTANDARD kullanmak için birçok farklı metot sağlamaktadır.

Spartan-3E FPGA'leri I/O standartlarını farklı kümelerde birleştirerek ek giriş esnekliği sağlamaktadır. Belirli VCCO gerilimleri için Tablo 3.3 ve Tablo 3.4'de bütün IOSTANDARD'larının sadece giriş olarak mı yada giriş ve çıkış olarak mı kullanılabilecekleri listelenmiştir.

Single-Ended IOSTANDARD	V _{CCO} Supply/Compatibility					Input Requirements	
	1.2V	1.5V	1.8V	2.5V	3.3V	V _{REF}	Board Termination Voltage (V _{TT})
LVTTTL	-	-	-	-	Input/Output	N/R ⁽¹⁾	N/R
LVC MOS33	-	-	-	-	Input/Output	N/R	N/R
LVC MOS25	-	-	-	Input/Output	Input	N/R	N/R
LVC MOS18	-	-	Input/Output	Input	Input	N/R	N/R
LVC MOS15	-	Input/Output	Input	Input	Input	N/R	N/R
LVC MOS12	Input/Output	Input	Input	Input	Input	N/R ⁽¹⁾	N/R
PCI33_3	-	-	-	-	Input/Output	N/R	N/R
PCI66_3	-	-	-	-	Input/Output	N/R	N/R
PCIX					Input/Output	N/R	N/R
HSTL_I_18	-	-	Input/Output	Input	Input	0.9	0.9
HSTL_III_18	-	-	Input/Output	Input	Input	1.1	1.8
SSTL18_I	-	-	Input/Output	Input	Input	0.9	0.9
SSTL2_I	-	-	-	Input/Output	Input	1.25	1.25

Not:

1. N/R – Giriş işlemleri için gerekmemektedir.

Tablo 3.3 Single-Ended IOSTANDARD Küme Uyumluluğu

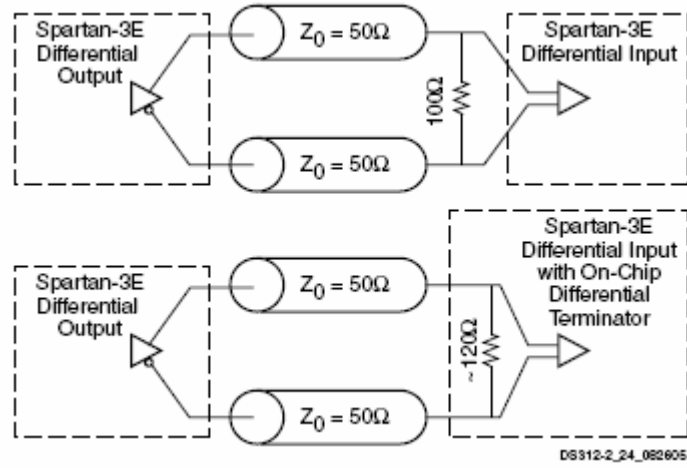
Differential IOSTANDARD	V _{CCO} Supply			Input Requirements: V _{REF}	Differential Bank Restriction ⁽¹⁾
	1.8V	2.5V	3.3V		
LVDS_25	Input	Input, On-chip Differential Termination, Output	Input	V _{REF} is not used for these I/O standards	Applies to Outputs Only
RSDS_25	Input	Input, On-chip Differential Termination, Output	Input		Applies to Outputs Only
MINI_LVDS_25	Input	Input, On-chip Differential Termination, Output	Input		Applies to Outputs Only
LVPECL_25	Input	Input	Input		No Differential Bank Restriction (other I/O bank restrictions might apply)
BLVDS_25	Input	Input, Output	Input		
DIFF_HSTL_I_18	Input, Output	Input	Input		
DIFF_HSTL_III_18	Input, Output	Input	Input		
DIFF_SSTL18_I	Input, Output	Input	Input		
DIFF_SSTL2_I	Input	Input, Output	Input		

Tablo 3.4 Differential IOSTANDARD Küme Uyumluluğu

3.2.3 Yonga-üzeri Fark Sonlandırma

Spartan-3E aygıtları giriş fark alıcı terminallerinde yonga-üzerinde ~120Ω'lık bir fark sonlandırma direnci barındırmaktadır. Spartan-3E aygıtının içerisindeki yonga-üzeri giriş fark sonlandırma direnci genel olarak fark devresi alıcısında bulunan 100Ω sonlandırma direncini ortadan kaldırmaktadır (Şekil 3.3). Fark sonlandırma uygulama elverdiğince LVDS, mini-LVDS ve RSDS standartları için kullanılmaktadır. Yonga-üzeri fark sonlandırma, 2.5V VCCO gerilimi ile beslenen kümelerde (Bankta) kullanılabilir. Bir fark I/O pin çiftinde fark sonlandırma direncini aktif etmek için DIFF_TERM özelliği TRUE olarak ayarlanmalıdır. DIFF_TERM değişkeni UCF dosyası içerisinde aşağıdaki formatta kullanılır:

```
INST <I/O_BUFFER_INSTANTIATION_NAME>
DIFF_TERM = "<TRUE/FALSE>" ;
```



Şekil 3.3 Differential (Fark) Giriş ve Çıktıları

3.2.4 Yukarı-Çekme Aşağı-Çekme Dirençleri

Her bir IOB içerisindeki yukarı-çekme ve aşağı-çekme dirençleri isteğe bağlı olarak, boşta olan bir I/O yâda sadece-giriş pinini belirli bir duruma sokmaya zorlar.

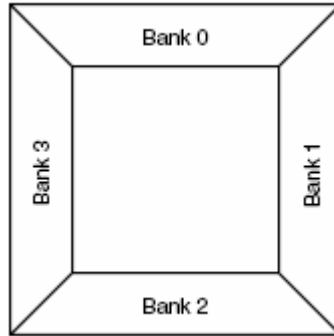
Yukarı-çekme ve aşağı-çekme dirençleri genellikle kullanılmayan I/O'lar, girişler ve 3-state çıkışlara uygulanmaktadır, fakat herhangi I/O yâda sadece-giriş pininde kullanılabilir. Yukarı-çekme direnci IOB'yi bir direnç vasıtasıyla VCCO'ya bağlamaktadır. Direnç değeri, VCCO geriliminin değerine bağlıdır. Aşağı-çekme direnci benzer şekilde IOB'yi bir direnç vasıtasıyla toprağa bağlamaktadır. PULLUP ve PULLDOWN özellikleri ve kütüphane örnekleri bu seçime bağlı dirençleri aktif etmektedir.

Varsayılan olarak PULLDOWN dirençleri tüm kullanılmayan I/O ve sadece-giriş pinlerini sonlandırmaktadır. Kullanılmayan I/O ve sadece-giriş pinleri alternatif olarak PULLUP yâda FLOAT olarak ayarlanabilmektedir. Yapılandırma işlemi süresince HSWAP pinindeki bir düşük lojik seviyesi, seçilen yapılandırma modunda aktif olarak kullanılmayan tüm I/O ve sadece-giriş pinlerindeki yukarı-çekme dirençlerini aktif etmektedir.

3.2.5 I/O Kümelere Ayırma (Banking)

Şekil 3.4’de gösterildiği gibi Spartan-3E mimarisinde IOB’lar dört I/O kümesine bölünmüştür. Her bir küme ayrı VCCO ve VREF kaynakları ile sürülmektedir. Ayrı kaynaklar her bir kümenin bağımsız VCCO ile beslenmesine olanak sağlamaktadır. Benzer şekilde, VREF gerilimi her bir küme için ayarlanabilir. VCCO ve VREF gereksinimleri için Tablo 3.3 ve Tablo 3.4’e bakınız.

Spartan-3E aygıtlarıyla çalışırken birçok farklı I/O standardı uyumludur ve verilen herhangi bir küme içerisinde birleştirilebilmektedir. Her bir küme LVDS_25, MINI_LVDS_25 ve RSDS_25 çıkış farklı standartlarından ikisini destekleyebilmektedir. Örnek olarak yonga-üzeri farklı sonlandırma kullanılırken, LVDS_25 çıkışları RSDS_25 çıkışları ve herhangi bir diğer farklı girişi geçerli bir kombinasyonu oluşturmaktadır.



Şekil 3.4 Spartan-3E I/O Kümeleri (üstten görünüm)

3.3 Yapılandırılabilir Lojik Bloklar (CLB) ve Dilim (Slice) Kaynakları

3.3.1 CLB Dizisi

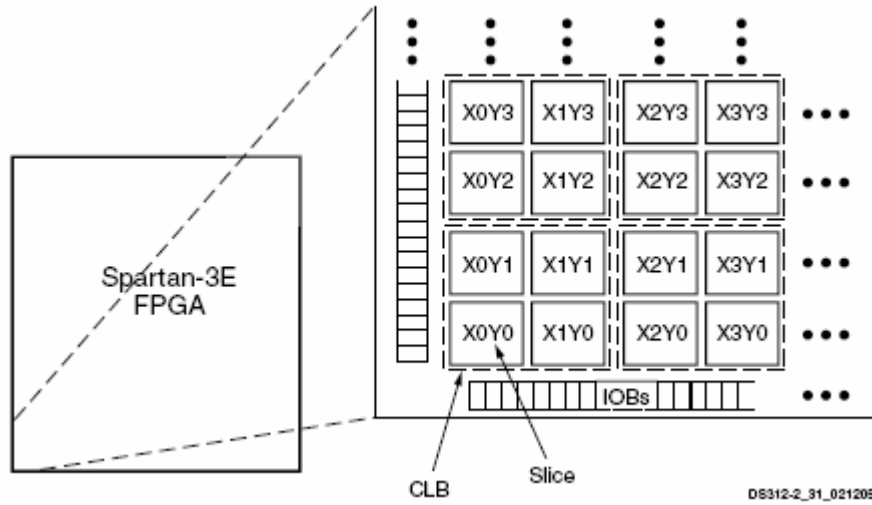
Yapılandırılabilir Lojik Bloklar (CLBs) birleşik devreler gibi senkron gerçeklemler için ana lojik kaynakları oluşturmaktadır. Her bir CLB dört dilimi barındırır ve her bir dilim de flip-flop ya da mandal olarak kullanılabilen iki özel depolama elemanı ve lojik gerçeklemler için iki adet Look-Up Tablosunu (LUT) barındırmaktadır.

LUT'lar 16x1 hafıza (RAM16) yâda 16-bit kaymalı kaydedici (SRL16) olarak kullanılabilirler. Bir tasarım içerisindeki birçok genel-amaçlı lojik CLB içerisinde otomatik olarak dilim kaynaklarına bağlanmaktadır. Her bir CLB birbiri ile aynıdır.

CLB'ler şekil 3.5'de görüldüğü gibi muntazam bir dizide satır ve sütun şeklinde yerleştirilmiştir. Her bir satır ve sütundaki CLB miktarları tablo 3.5'de listelenmiştir.

Device	CLB Rows	CLB Columns	CLB Total ⁽¹⁾	Slices	LUTs / Flip-Flops	Equivalent Logic Cells	RAM16 / SRL16	Distributed RAM Bits
XC3S100E	22	16	240	960	1,920	2,160	960	15,360
XC3S250E	34	26	612	2,448	4,896	5,508	2,448	39,168
XC3S500E	46	34	1,164	4,656	9,312	10,476	4,656	74,496
XC3S1200E	60	46	2,168	8,672	17,344	19,512	8,672	138,752
XC3S1600E	76	58	3,688	14,752	29,504	33,192	14,752	236,032

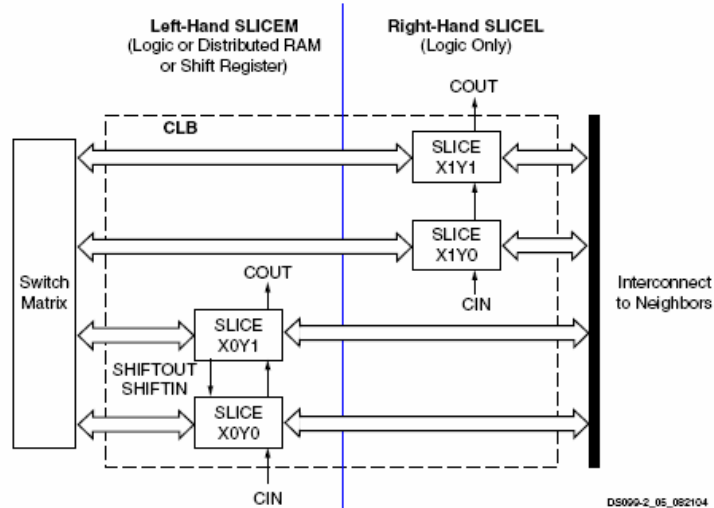
Tablo 3.5 Spartan-3E CLB Kaynakları



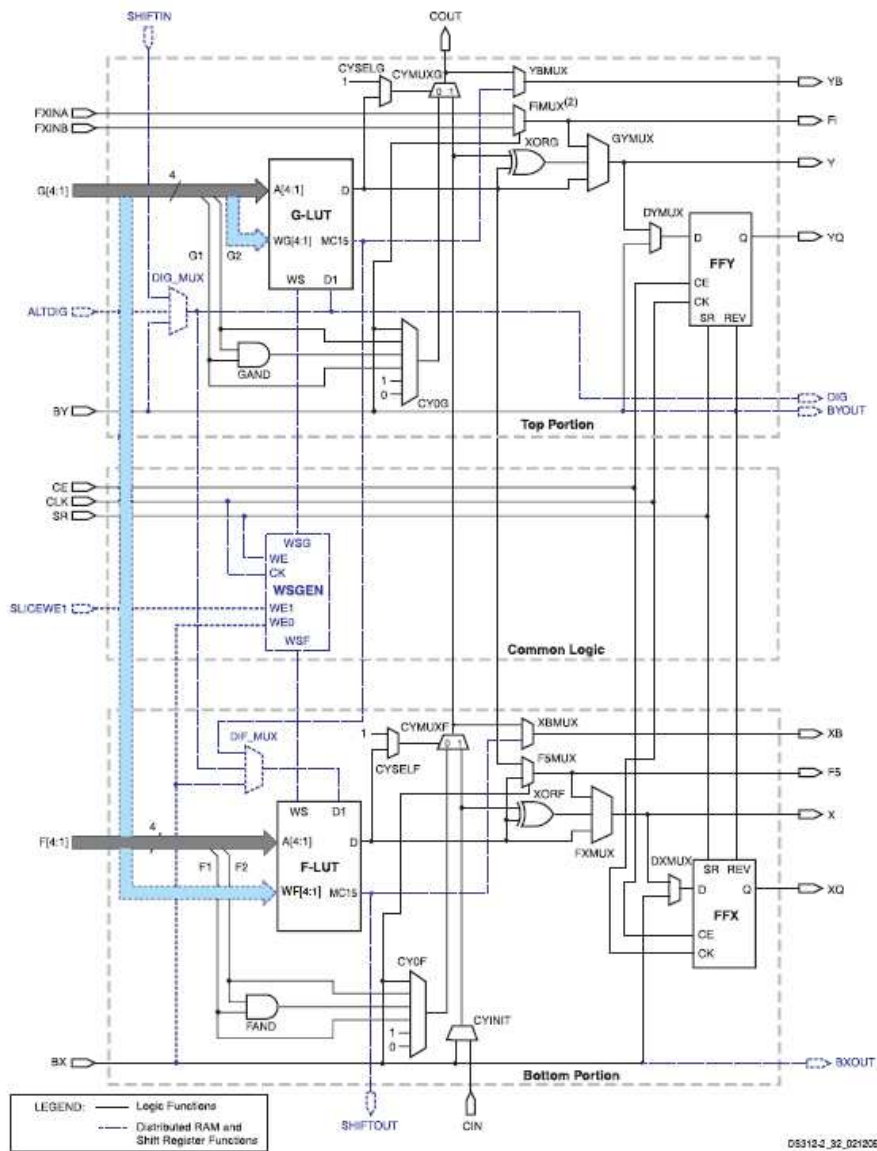
Şekil 3.5 CLB Yerleşimi

3.3.2 Dilimler

Her bir CLB şekil 3.5'de gösterildiği gibi dört adet arabağlantılı dilimden oluşmaktadır. Bu dilimler çiftler halinde gruplanmışlardır. Her bir çift bağımsız bir destek zinciri ile bir sütun olarak kurulmuştur. Soldaki çift lojik ve hafıza işlevlerinin ikisini de desteklemektedir ve dilimleri SLICEM olarak adlandırılmaktadır. Sağdaki çift sadece lojik işlevleri desteklemektedir ve SLICEL olarak adlandırılmaktadır.



Şekil 3.6 CLB İçerisinde Dilimlerin Düzeni



Şekil 3.7 SLICEM Basitleştirilmiş Diyagramı

3.4 Sayısal Saat Yöneticisi (DCMs)

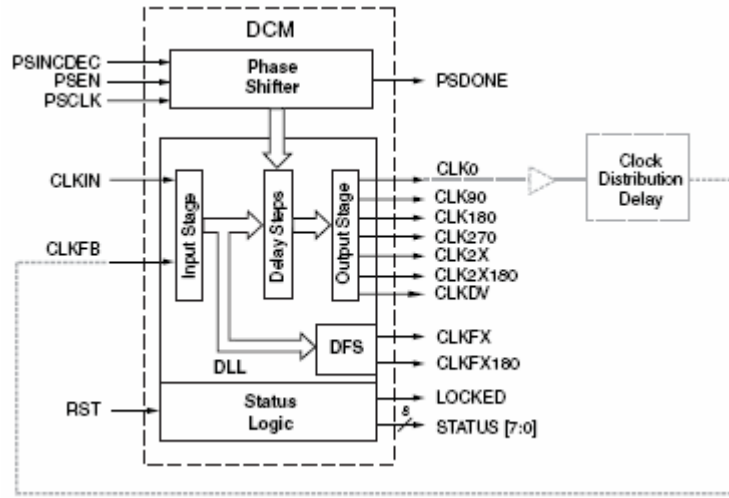
Spartan-3E Sayısal saat yöneticisi (DCM) saat frekansı, faz kayması ve eğri üzerinde tamamen esnek bir kontrol sağlamaktadır. Bunu başarmak için DCM, farklı sıcaklık ve gerilim şartları altında dahi saat sinyali karakteristiğini yüksek oranda kesinlikle çalışmasını sürdürmek için geribesleme kullanan tamamen sayısal bir kontrol sistemi olan Gecikme-Kilitlemeli Çevrim (DLL) kullanmaktadır.

Spartan-3E içerisindeki DCM lojik diziler içerisindeki CLB'ler tarafından çevrilmiştir. Sayısal saat yöneticisi bir tasarım içerisinde "DCM" örneği (primitive) kullanılarak oluşturulur.

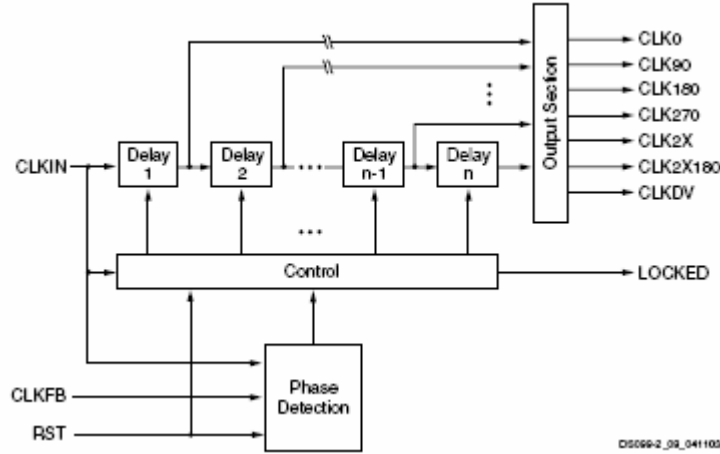
DCM, üç önemli işlevi desteklemektedir:

- **Saat-eğri (Clock-skew) Yok etme:** Bir sistem içerisindeki saat eğrisi bir saat sinyalinin yonga üzerindeki farklı noktalara farklı zamanlarda ulaşmasından meydana gelmektedir. Saat eğrisi, yüksek frekanslı uygulamalarda istenmeyen ayarlama (setup) ve tutma zamanı gereksinimlerini ve saat-çıkış süresini arttırmaktadır. DCM, gelen saat sinyali ile oluşturulan çıkış sinyalinin fazını düzenleyerek saat eğrisini yok etmektedir. Bu mekanizma etkin olarak saat yayılım gecikmesini ortadan kaldırmaktadır.
- **Frekans Sentezleyici:** DCM, gelen saat sinyali tarafından sürülen geniş çapta farklı çıkış saat frekansları üretebilmektedir. Bu, birçok farklı etmen vasıtasıyla giriş saat sinyalinin frekansının çarpılması ve/veya bölünmesi ile başarılmaktadır.
- **Faz Kaydırma:** DCM, giriş saat sinyaline bağlı olarak çıkış saat sinyalinin fazını kaydırabilme yeteneğine sahiptir.

Tek bir tasarım örneği olmasına karşın DCM, dâhili olarak ilişkili dört işlevsel birimden oluşmaktadır: Gecikme-Kilitlemeli Çevrim (DLL), Sayısal Frekans Sentezleyici (DFS), Faz Kaydırıcı (PS) ve Durum Lojiği (Status Logic). Her bir bileşen şekil 3.8'de görüldüğü gibi kendi birleştirilmiş sinyaline sahiptir.



Şekil 3.8 DCM İşlevsel Blokları ve Birleştirilmiş Sinyaller



Şekil 3.9 DLL'in Basitleştirilmiş İşlevsel Diyagramı

Sinyal	Yön	Tanım
CLKIN	Giriş	Gelen saat sinyalini alır.
CLKFB	Giriş	CLK0 yâda CLK2X'i geribesleme sinyali olarak kabul eder. (CLK_FEEDBACK değişkenine göre ayarlanır)
CLK0	Çıkış	CLKIN ile aynı frekansta ve frekansta bir saat sinyali üretir.
CLK90	Çıkış	CLKIN ile aynı frekansta ve 90 ⁰ fazı kaydırılmış bir saat sinyali üretir.
CLK180	Çıkış	CLKIN ile aynı frekansta ve 180 ⁰ fazı kaydırılmış bir saat sinyali üretir.
CLK270	Çıkış	CLKIN ile aynı frekansta ve 270 ⁰ fazı kaydırılmış bir saat sinyali üretir.
CLK2X	Çıkış	CLKIN'in iki katı frekansında ve aynı fazda bir saat sinyali üretir.
CLK2X180	Çıkış	CLKIN'in iki katı frekansında ve 180 ⁰ fazı kaydırılmış bir saat sinyali üretir.
CLKDV	Çıkış	CLKIN ile fazı düzenlenmiş düşük frekanslı bir sinyal üretmek için CLKIN frekansını CLKDV_DIVIDE değeri ile böler.

Tablo 3.6 DLL Sinyalleri

3.5 Yapılandırma

Yapılandırma, Xilinx geliştirme yazılımı tarafından üretilen bir tasarımın bit dizisinin FPGA'in dâhili yapılandırma hafızasına yükleme işlemidir. Spartan-3E FPGA'leri yapılandırma verilerinin tüm işlevsel elemanlarını ve yönlendirme kaynaklarını toplu olarak kontrol eden sağlam, yeniden programlanabilir, statik CMOS yapılandırma mandalları (CCLs) içerisine yüklenerek programlanmaktadır. FPGA yapılandırma verisi harici olarak bir EPROM yâda bazı diğer değişken olmayan ortamda kart üzerinde yâda dışındaki bir ortamda depolanmaktadır. Güç uygulandıktan sonra yapılandırma verisi aşağıdaki yedi farklı moddan biri kullanılarak FPGA içerisine yazılır:

- Master Serial, bir Xilinx Platform Flash PROM'dan
- SPI (Serial Peripheral Interface), endüstri-standardı bir SPI serial Flash'dan
- Byte Peripheral Interface (BPI), endüstri-standardı x8 yâda x8/x16 paralel NOR Flash
- Slave Serial, tipik olarak bir işlemciden yüklenir
- Slave Paralel, tipik olarak bir işlemciden yüklenir
- Boundary Scan (JTAG), tipik olarak sistem kontrolcüsü yâda bir işlemciden yüklenir

3.5.1 Yapılandırma Dosyası

Spartan-3E ailesinin üyeleri kapasitelerine yani içerisindeki kapı sayılarına bağlı olarak çeşitli boyutlarda yapılandırma bitlerine ihtiyaç duyarlar. Tablo 3.7'da Spartan-3E ailesinin her bir üyesinin gerekli yapılandırma bit dizisi boyutu görülmektedir.

Spartan-3E FPGA	Number of Configuration Bits
XC3S100E	581,344
XC3S250E	1,353,728
XC3S500E	2,270,208
XC3S1200E	3,841,184
XC3S1600E	5,969,696

Tablo 3.7 Spartan-3E Yapılandırma Dosyası Boyutları

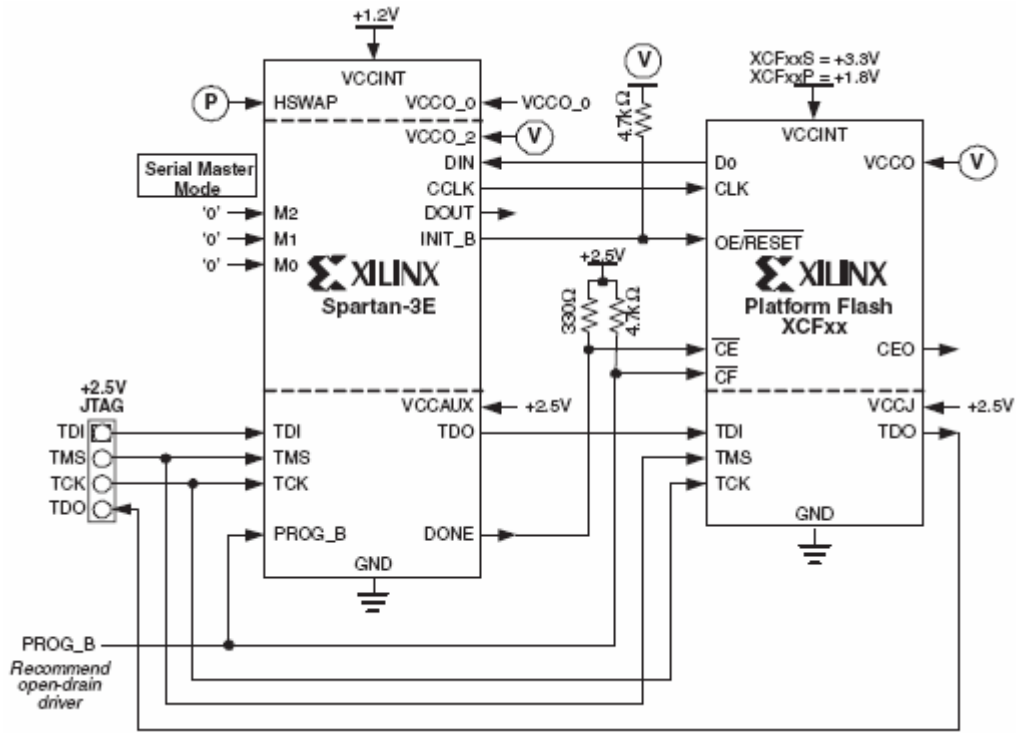
Yapılandırma modu pinleri M2,M1 ve M0, tablo 3.8’de görülen şekilde istenen moda ayarlanarak yapılandırma modu seçilebilmektedir.

	Master Serial	SPI	BPI	Slave Parallel	Slave Serial	JTAG
M[2:0] mode pin settings	<0:0:0>	<0:0:1>	<0:1:0>=Up <0:1:1>=Down	<1:1:0>	<1:1:1>	<1:0:1>
Data width	Serial	Serial	Byte-wide	Byte-wide	Serial	Serial
Configuration memory source	Xilinx Platform Flash	Industry-standard SPI serial Flash	Industry-standard parallel NOR Flash or Xilinx parallel Platform Flash	Any source via microcontroller, CPU, Xilinx parallel Platform Flash, etc.	Any source via microcontroller, CPU, Xilinx Platform Flash, etc.	Any source via microcontroller, CPU, System ACE™ CF, etc.
Clock source	Internal oscillator	Internal oscillator	Internal oscillator	External clock on CCLK pin	External clock on CCLK pin	External clock on TCK pin
Total I/O pins borrowed during configuration	8	13	46	21	8	0
Configuration mode for downstream daisy-chained FPGAs	Slave Serial	Slave Serial	Slave Parallel	Slave Parallel or Memory Mapped	Slave Serial	JTAG
Stand-alone FPGA applications (no external download host)	✓	✓	✓	Possible using XCFxxP Platform Flash, which optionally generates CCLK	Possible using XCFxxP Platform Flash, which optionally generates CCLK	
Uses low-cost, industry-standard Flash		✓	✓			
Supports optional MultiBoot, multi-configuration mode			✓			

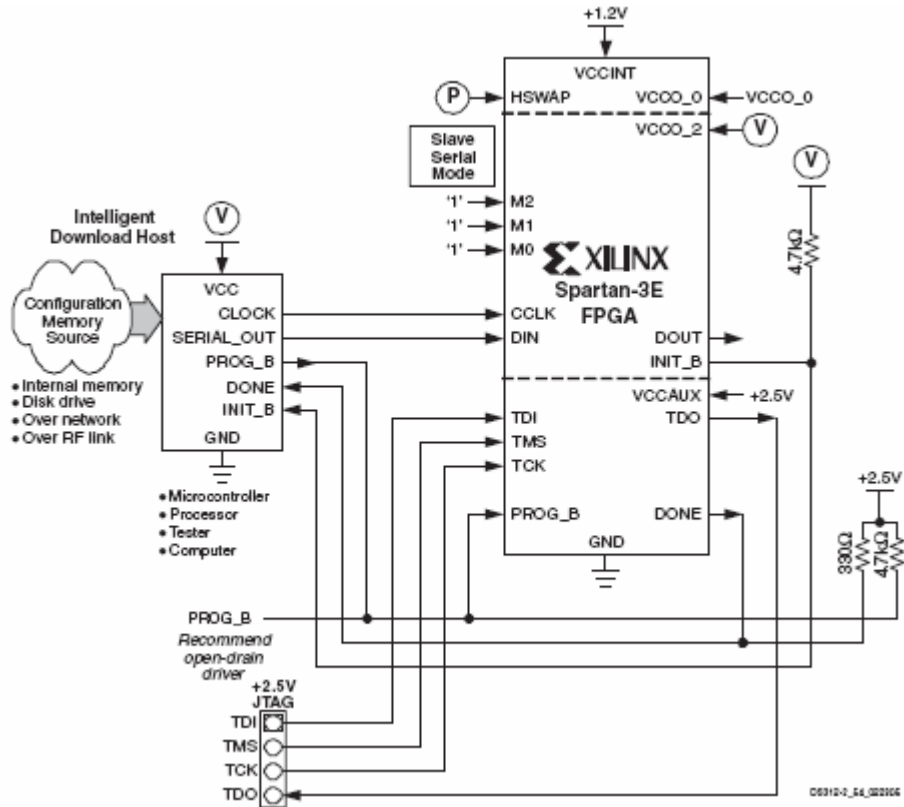
Tablo 3.8 Yapılandırma Modu Seçenekleri ve Pin Ayarları

Seri yapılandırma modunda FPGA’ye her bir CCLK çevriminde bir bit iletilmektedir. Master Serial modunda FPGA CCLK pinini sürmektedir. Slave Serial modunda ise CCLK pini harici bir kaynak tarafından sürülmektedir. Her iki seri yapılandırma modunda da her bir veri byte’ının yüksek anlamlı biti (MSB) DIN pinine ilk olarak

yazılmaktadır. Aşağıda şekil 3.10 ve şekil 3.11’de Master Serial ve Slave Serial yapılandırma için PROM ve FPGA sinyal bağlantıları görülmektedir.



Şekil 3.10 Master Serial Modda Yapılandırma



Şekil 3.11 Slave Serial Modda Yapılandırma

3.6 DC Karakteristikleri

Spartan-3E ailesinin çekirdeği $1.2V_{CCINT}$ gerilimi ile beslenmektedir. I/O bloklarını besleyen V_{CCO} ise kullanılan I/O standardına bağlı olarak $1.2V - 3.3V$ arasında olabilmektedir. Aşağıda tablo 3.9’da maksimum ve minimum gerilim değerleri listelenmiştir.

Symbol	Description	Conditions	Min	Max	Units	
V_{CCINT}	Internal supply voltage		-0.5	1.32	V	
V_{CCAUX}	Auxiliary supply voltage		-0.5	3.00	V	
V_{CCO}	Output driver supply voltage		-0.5	3.75	V	
V_{REF}	Input reference voltage		-0.5	$V_{CCO} + 0.5^{(1)}$	V	
$V_{IN}^{(1,2,3)}$	Voltage applied to all User I/O pins and Dual-Purpose pins	Driver in a high-impedance state	Commercial	-0.95	4.4	V
			Industrial	-0.85	4.3	V
	Voltage applied to all Dedicated pins	All temp. ranges	-0.5	$V_{CCAUX} + 0.5^{(3)}$	V	
I_{IK}	Input clamp current per I/O pin	$-0.5 V < V_{IN} < (V_{CCO} + 0.5 V)$	-	± 100	mA	
V_{ESD}	Electrostatic Discharge Voltage	Human body model	-	± 2000	V	
		Charged device model	-	± 500	V	
		Machine model	-	± 200	V	
T_J	Junction temperature		-	125	$^{\circ}C$	
T_{STG}	Storage temperature		-65	150	$^{\circ}C$	

Tablo 3.9 Mutlak Maksimum Oranlar

Symbol	Description	Min	Nominal	Max	Units	
T_J	Junction temperature	Commercial	0	-	85	$^{\circ}C$
		Industrial	-40	-	100	$^{\circ}C$
V_{CCINT}	Internal supply voltage	1.140	1.200	1.260	V	
$V_{CCO}^{(1)}$	Output driver supply voltage	1.100	-	3.450	V	
V_{CCAUX}	Auxiliary supply voltage	2.375	2.500	2.625	V	
$V_{IN}^{(2,3,4)}$	Input voltage extremes to avoid turning on I/O protection diodes.	I/O, Input-only, and Dual-Purpose pins ⁽²⁾	-0.5	-	$V_{CCO} + 0.5$	V
		Dedicated pins ⁽³⁾	-0.5	-	$V_{CCAUX} + 0.5$	
T_{IN}	Input signal transition time ⁽³⁾	-	-	500	ns	

Tablo 3.10 Önerilen Çalışma Şartları

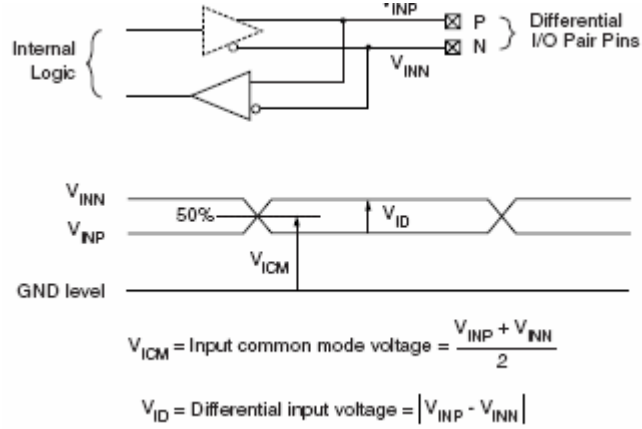
3.6.1 DC Giriş ve Çıkış Seviyeleri

V_{IL} ve V_{IH} değerleri önerilen giriş gerilimleridir. V_{OL} ve V_{OH} ise önerilen çalışma koşullarında garantilenmiş çıkış gerilimleridir.

IOSTANDARD Attribute	V _{CCO} for Drivers ⁽²⁾			V _{REF}			V _{IL}	V _{IH}
	Min (V)	Nom (V)	Max (V)	Min (V)	Nom (V)	Max (V)	Max (V)	Min (V)
LVTTTL	3.0	3.3	3.45	V _{REF} is not used for these I/O standards			0.8	2.0
LVCMOS33 ⁽⁴⁾	3.0	3.3	3.45				0.8	2.0
LVCMOS25 ^(4,5)	2.3	2.5	2.7				0.7	1.7
LVCMOS18 ⁽⁴⁾	1.65	1.8	1.95				0.38	0.8
LVCMOS15 ⁽⁴⁾	1.4	1.5	1.6				0.38	0.8
LVCMOS12 ⁽⁴⁾	1.1	1.2	1.3				0.38	0.8
PCI33_3	3.0	3.3	3.45				0.9	1.5
PCI66_3	3.0	3.3	3.45				0.9	1.5
PCIX	3.0	3.3	3.45					
HSTL_I_18	1.7	1.8	1.9	0.8	0.9	1.1	V _{REF} - 0.1	V _{REF} + 0.1
HSTL_III_18	1.7	1.8	1.9	-	1.1	-	V _{REF} - 0.1	V _{REF} + 0.1
SSTL18_I	1.7	1.8	1.9	0.833	0.900	0.969	V _{REF} - 0.125	V _{REF} + 0.125
SSTL2_I	2.3	2.5	2.7	1.15	1.25	1.35	V _{REF} - 0.125	V _{REF} + 0.125

Tablo 3.11 Single-Ended Standartları Kullanan Kullanıcı I/O'ları İçin Önerilen Çalışma Koşulları

Fark I/O Standartları



Şekil 3.12 Fark Giriş Gerilimleri

IOSTANDARD Attribute	V _{CCO} for Drivers ⁽¹⁾			V _{ID}			V _{ICM}		
	Min (V)	Nom (V)	Max (V)	Min (mV)	Nom (mV)	Max (mV)	Min (V)	Nom (V)	Max (V)
LVDS_25	2.375	2.50	2.625	100	350	600	0.30	1.25	2.20
BLVDS_25	2.375	2.50	2.625	100	350	600	0.30	1.25	2.20
MINI_LVDS_25	2.375	2.50	2.625	200	-	600	0.30	-	2.2
LVPECL_25 ⁽²⁾	Inputs Only			100	800	1000	0.5	1.2	2.0
RSDS_25	2.375	2.50	2.625	100	200	-	0.3	1.20	1.4
DIFF_HSTL_I_18	1.7	1.8	1.9	100	-	-	0.8	-	1.1
DIFF_HSTL_III_18	1.7	1.8	1.9	100	-	-	0.8	-	1.1
DIFF_SSTL18_I	1.7	1.8	1.9	100	-	-	0.7	-	1.1
DIFF_SSTL2_I	2.3	2.5	2.7	100	-	-	1.0	-	1.5

Tablo 3.12 Fark Sinyal Standartlarını Kullanan Kullanıcı I/O'ları İçin Önerilen Çalışma Koşulları

3.6.2 Bypass Kapasitesi Hesapları

3.6.2.1 Çekirdek V_{CC} Bypass Kapasitesi Hesabı

Bir Xilinx FPGA'inin çekirdeği oldukça küçük, kısa süreli (<50ps) bazı akımlara ihtiyaç duymaktadır. Bu akımların karışımı bütün aygıt içinde toplanarak amperler mertebesine ulaşır. Sayısız özel anahtarlamalar olduğundan çekirdek bypass tasarımı ortalama bir enerji depolama ihtiyacını ortaya çıkarmaktadır. Bu metot bypass kapasite ihtiyaçlarının hesaplanmasında sade bir hesaplama meydana getirir. Bu yüzden tipik güç hesaplamaları kullanılmalıdır. Temel bir denklem (denklem-1) yüksek-frekans kapasite ihtiyacını belirlemede kullanılmaktadır.

$$P = CV^2F \quad (1)$$

C'nin çözümü FPGA'in eşdeğer kapasitansı (C_{EQ}) için bir denklem verir. Burada 'P' çekirdek gücü, 'V' çekirdek gerilimi, 'F' saat frekansıdır.

$$C_{EQ} = \frac{P}{V^2F} \quad (2)$$

C_{EQ} bütün FPGA'lerin anahtarlama kapasitansı denklemini göstermektedir. Bu değer uygun boyuttaki yüksek frekans bypass kondansatörünün seçimini sağlamaktadır. V_{CCINT} değişimini (gürültü) küçük tutmak için bypass kapasitansı FPGA eşdeğer kapasitansından oldukça büyük olmalıdır. Yüksek frekans bypass kapasitansı (C_{BT}) FPGA eşdeğer kapasitansının (C_{EQ}) 25 ile 100 katı arasında bir çarpan olmalıdır. 25 çarpanı (S) V_{CCINT} 'de 25'de 1 yâda %4 değişime neden olur. Ölçek çarpanını 50 alırsak:

$$C_{BT} = (S \times C_{EQ}) \quad (3)$$

Her bir V_{CCINT}/GND ikilisi bir yüksek-frekans bypass kondansatörüne sahip olmalıdır. Her bir kondansatörün en uygun değerini (C_{BI}) belirlemek için toplam C_{BT} , aygıt üzerindeki V_{CCINT} pini sayısına bölünür ve yukarıya doğru en yakın kullanılabilir değere yuvarlanır.

$$C_{BI} = \frac{C_{BT}}{N_P} \quad (4)$$

Denklemler birleştirilerek, her bir yüksek-frekans V_{CCINT} kondansatörünün (C_{BI}) minimum değerini veren denklem elde edilir:

$$C_{BI} = \frac{C_{BT}}{N_P} = \frac{S \times C_{EQ}}{N_P} = \frac{S \times \frac{P}{V^2 F}}{N_P} \quad (5)$$

Örnek olarak, PQ208 kılıfındaki bir XC2S200 ortalama 35 MHz (F) de çalışsın ve 1.8 Watt (P) güç harcasın. V_{CCINT} 'in değişimi %2'den (ölçek çarpanı 50 (S)) büyük olmasın. PQ208 12 adet V_{CCINT} pinine (N_P) sahiptir.

$$\begin{aligned}
P &= 1.8 \text{ Watts} \\
V &= V_{CCINT} = 2.5 \text{ V} \\
F &= 35 \text{ MHz} \\
S &= 50 \\
N_p &= 12
\end{aligned}$$

$$C_{BI} = \frac{50 \times \frac{1.8W}{(2.5V)^2 \times 35MHz}}{12} = 0.034\mu F$$

Her bir yüksek-frekans kondansatörünün kapasitesi minimum 0.034 μ F 'dır. Bu değere yakın ve büyük kapasiteli bir kondansatör seçilmelidir (0.047 μ F yâda 0.1 μ F).

Yüksek-frekans bypass kondansatörü belirlendikten sonra sıra orta ve düşük-frekans kondansatörlerin seçimine geldi. 47 μ F 'dan 100 μ F 'a kadar tantalum kondansatör orta-frekans için uygundur. Eğer tantalum mevcut değilse, düşük endüktanslı alüminyum elektrolitik tipi kullanılabilir. Aygıt, içerisindeki her 1500 CLB için en az bir adet orta-frekans kondansatöre ihtiyaç duymaktadır. Bu, XC2S200 kendisine en fazla 3 cm uzaklıkta monte edilmiş bir adet orta-frekans kondansatörüne sahip olmalıdır.

Her dört FPGA için en az bir adet düşük-frekans kondansatörün (470 μ F 'dan 3,300 μ F 'a kadar) kartın herhangi bir yerine monte edilmesi gerekmektedir.

3.6.2.2 I/O Güç Kaynağı Bypass Kapasitesi Hesabı

Geniş Giriş/çıkış adetli mimariler artık tek bir aygıt üzerinde mevcut durumdadır ve dikkatler V_{CCO} güç kaynağı pinlerine uygun bypass sağlamaya çekilmelidir. V_{CCINT} gibi I/O bypass ihtiyacı da ortalama enerji ihtiyacına bağlı olarak hesaplanır. Farklı I/O kümelerinin sıklıkla farklı gerilimlerde çalışması nedeniyle her bir kümenin bypass ağı ayrı olarak tasarlanmalıdır. Bir kümeye yâda bir gerilim seviyesine bağlı eşdeğer kapasitansı (C_{EQ}) belirlemek için toplam yük sayısı (N_L) yük değeri (C_L) ile çarpılmaktadır.

$$C_{EQ} = (N_L \times C_L) \quad (6)$$

VCCO deęişimini küçük tutmak için bypass kapasitansı eşdeęer yük kapasitansından oldukça büyük olmalıdır. Yüksek frekans bypass kapasitansı (C_{BT}) eşdeęer yük kapasitansının (C_{EQ}) 25 ile 100 katı arasında bir çarpan olmalıdır. Denklem-3, toplam yüksek-frekans bypass kapasitansını (C_{BT}) hesaplamak için kullanılmaktadır. Her VCCO/GND çifti bir yüksek-frekans kondansatörüne sahip olmalıdır. Her bir kondansatörün en uygun deęerini (C_{BI}) belirlemek için toplam C_{BT} , küme içindeki (N_P) VCCO pini sayısına bölünür ve yukarıya doğru en yakın kullanılabilir deęere yuvarlanır. Denklemler birleştirilerek, her bir yüksek-frekans VCCO kondansatörünün (C_{BI}) minimum deęerini veren denklem elde edilir:

$$C_{BI} = \frac{C_{BT}}{N_P} = \frac{S \times C_{EQ}}{N_P} = \frac{S \times N_L \times C_L}{N_P} \quad (7)$$

XC2S200 örneęine devam edersek, tasarımın bir bank içerisinde 36 adet çıkışa sahip ve her bir yükün 10pF olduğunu varsayalım. PQ208 her bir küme içinde 5 adet VCCO pinine sahiptir ve maksimum deęişim %2 olmalıdır.

$$\begin{aligned} S &= 50 \\ N_L &= 36 \\ C_L &= 10 \text{ pF} \\ N_P &= 5 \end{aligned}$$

$$C_{BI} = \frac{50 \times 36 \times 10 \text{ pF}}{5} = 0.004 \mu\text{F}$$

Her bir yüksek-frekans kondansatörünün kapasitesi minimum 0.004 μ F 'dır. Bu deęere yakın ve büyük kapasiteli bir kondansatör seçilmelidir (0.0047 μ F yâda 0.01 μ F).

Her bir'den dörde kadar VCCO kümesi için 47 μ F 'dan 100 μ F 'a kadar bir orta-frekans kondansatörü olmalıdır. Tantalum kondansatör orta-frekans için uygundur. Eęer tantalum mevcut deęilse, düşük endüktanslı alüminyum elektrolitik tipi kullanılabilir. Bu kondansatörler VCCO pinlerinden en fazla 3 cm uzaklıkta monte edilmelidir.

Her gerilim seviyesi için en az bir adet düşük-frekans kondansatör (470 μ F 'dan 3,300 μ F 'a kadar) kullanılmalıdır. Her dört aygıt için bir adet kondansatör kullanılabilir.

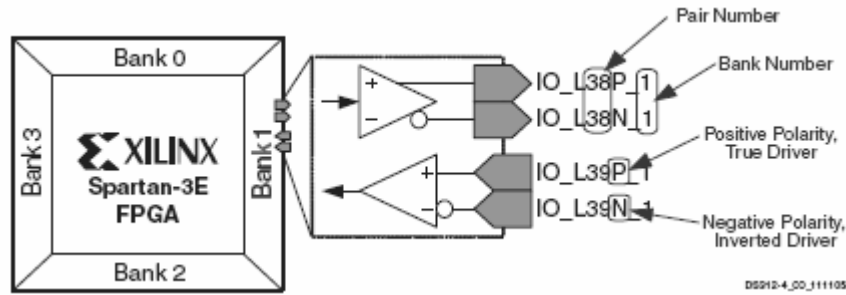
3.7 Spartan-3E Pin Bağlantıları

Aşağıda tablo3.13’de Spartan-3E FPGA pin türleri ve açıklamaları gruplar halinde görülmektedir.

Pin Türü	Açıklama	Pin Adı
I/O	Kısıtlanmamış genel amaçlı kullanıcı I/O pinleri. Birçok pin fark I/O biçiminde beraber eşleştirilebilmektedir.	IO IO_Lxy_#
Giriş	Kısıtlanmamış genel amaçlı sadece-giriş pinleri. Bu pinler çıkış yapısına sahip değildirler.	IP IP_Lxy_#
Çift-Amaçlı	Çift-amaçlı pin yapılandırma işlemi sırasında bazı yapılandırma modlarında kullanılmaktadır ve yapılandırmadan sonra genellikle kullanıcı I/O olarak kullanılabilir. Yapılandırma sırasında bu pin kullanılmıyorsa, I/O türü pin olarak davranmaktadır.	M[2:0] HSWAP CCLK MOSI/CSI_B D[7:1] D0/DIN CSO_B RDWR_B BUSY/DOUT INIT_B A[23:20] A19/VS2 A18/VS1 A17/VS0 A[16:0] LDC[2:0] HDC
VREF	Kullanıcı-I/O veya sadece-giriş pini olan çift-amaçlı pin yâda aynı küme içerisindeki tüm diğer VREF'ler ile birlikte belirli I/O standartları için bir referans gerilim girişini sağlamaktadır. Bir küme içerisinde referans gerilimi olarak kullanılacak ise küme içerisindeki tüm VREF pinleri bağlanmalıdır.	IP/VREF_# IP_Lxx_#/VREF_# IO/VREF_# IO_Lxx_#/VREF_#
CLK	Özel bir saat tampon sürücüsüne bir giriş yâda kullanıcı-I/O'dur. Her kılıf bütün aygıtın opsiyonel saat sinyali için 16 global saat girişine sahiptir. RHCLK girişleri opsiyonel olarak aygıtın sağ-yarı kısmına saat sinyali sağlar. LHCLK girişleri opsiyonel olarak aygıtın sol-yarı kısmına saat sinyali sağlar. Saat pinlerinden bazıları çift-amaçlı yapılandırma pinleri ile paylaşılmaktadır ve Çift-tür olarak ele alınır.	IO_Lxx_#/GCLK[15:2], IP_Lxx_#/GCLK[1:0], IO_Lxx_#/LHCLK[7:0], IO_Lxx_#/RHCLK[7:0]
CONFIG	Tek-amaçlı yapılandırma pini. Kullanıcı-I/O olarak kullanılamamaktadır. Her kılıf iki adet tek-amaçlı yapılandırma pinine sahiptir. Bu pinler VCCAUX tarafından beslenmektedir.	DONE, PROG_B
JTAG	Tek-amaçlı JTAG pini. Kullanıcı-I/O olarak kullanılamamaktadır. Her kılıf dört adet tek-amaçlı JTAG pinine sahiptir. Bu pinler VCCAUX tarafından beslenmektedir.	TDI, TMS, TCK, TDO
GND	Tek-amaçlı toprak pini. GND pinlerinin sayısı kullanılan kılıfa bağlıdır ve tüm GND pinleri bağlanmalıdır.	GND
VCCAUX	Tek-amaçlı yardımcı güç kaynağı pini. VCCAUX pin sayısı kullanılan kılıfa bağlıdır ve tüm VCCAUX pinleri +2.5V'a bağlanmalıdır.	VCCAUX
VCCINT	Tek-amaçlı çekirdek lojik güç kaynağı pini. VCCINT pin sayısı kullanılan kılıfa bağlıdır ve tüm VCCINT pinleri +1.2V'a bağlanmalıdır.	VCCINT
VCCO	Aynı küme içerisindeki tüm diğer VCCO pinleri ile birlikte, bu pin I/O kümesi içerisindeki çıkış tamponlarına besleme sağlar ve bazı I/O standartları için giriş eşik gerilimini belirler.	VCCO_#
N.C.	Bu kılıf pini belirli aygıt/kılıf bileşimlerinde bağlantı olmayan pinleri belirtir fakat aynı kılıftaki daha büyük aygıtlarda bağlanmış olabilmektedir.	N.C.

Tablo 3.13 Spartan-3E Pin Türleri

Eğer bir pin “Lxyy_#” formatında etiketlenmişse bu pin fark sinyali standardını desteklemektedir. Pin adının soneki aşağıdaki anlama gelmektedir. Şekil 3.12’de özel bir örnek olarak 1. kümedeki bir fark giriş ve çıkışı görülmektedir. ‘L’ bu pinin bir fark sinyali çiftinin bir parçası olduğunu belirtmektedir. ‘xx’ bir fark sinyali pin-çiftini belirten her bir kümeye özel iki basamaklı bir tamsayıdır. ‘y’ gerçek sinyal için ‘P’ evirilmiş çıkış için ‘N’ harfini alır. Bu iki pin bir fark sinyali pin-çiftini oluşturur. ‘#’ 0’dan 3’e bir tamsayıdır ve bağlı bulunan küme numarasını belirtir.



Şekil 3.12 Fark Sinyal Çifti Etiketlemesi

Device	Package	Maximum User I/Os and Input-Only	Maximum Input-Only	Maximum Differential Pairs	All Possible I/Os by Type					
					I/O	INPUT	DUAL	VREF	CLK ⁽¹⁾	N.C.
XC3S100E	VQ100	66	7	30	16	1	21	4	24	0
XC3S250E		66	7	30	16	1	21	4	24	0
XC3S100E	CP132	83	11	35	16	2	42	7	16	9
XC3S250E		92	7	41	22	0	46	8	16	0
XC3S500E		92	7	41	22	0	46	8	16	0
XC3S100E	TQ144	108	28	40	22	19	42	9	16	0
XC3S250E		108	28	40	20	21	42	9	16	0
XC3S250E	PQ208	158	32	65	58	25	46	13	16	0
XC3S500E		158	32	65	58	25	46	13	16	0
XC3S250E	FT256	172	40	68	62	33	46	15	16	16
XC3S500E		190	41	77	76	33	46	19	16	0
XC3S1200E		190	40	77	78	31	46	19	16	0
XC3S500E	FG320	232	56	92	102	48	46	20	16	18
XC3S1200E		250	56	99	120	47	46	21	16	0
XC3S1600E		250	56	99	120	47	46	21	16	0
XC3S1200E	FG400	304	72	124	156	62	46	24	16	0
XC3S1600E		304	72	124	156	62	46	24	16	0
XC3S1600E	FG484	376	82	156	214	72	46	28	16	0

Tablo 3.14 Her Bir Paket ve Aygıtı Göre Kullanılabilir Maksimum Kullanıcı I/O Sayıları

208 pinli plastik kare düz paketi (PQFP), PQ208, Spartan-3E ailesinin XC3S250E ve XC3S500E aygıtlarında kullanılmaktadır. Aşağıda tablo 3.15’de PQ208 kılıfının tüm pinleri küme numaraları ve kümedeki pin isimlerine göre listelenmiştir.

Bank	XC3S250E XC3S500E Pin Name	PQ208 Pin	Type
0	IO	P187	I/O
0	IO/VREF_0	P179	VREF
0	IO_L01N_0	P161	I/O
0	IO_L01P_0	P160	I/O
0	IO_L02N_0/VREF_0	P163	VREF
0	IO_L02P_0	P162	I/O
0	IO_L03N_0	P165	I/O
0	IO_L03P_0	P164	I/O
0	IO_L04N_0/VREF_0	P168	VREF
0	IO_L04P_0	P167	I/O
0	IO_L05N_0	P172	I/O
0	IO_L05P_0	P171	I/O
0	IO_L07N_0/GCLK5	P178	GCLK
0	IO_L07P_0/GCLK4	P177	GCLK
0	IO_L08N_0/GCLK7	P181	GCLK
0	IO_L08P_0/GCLK6	P180	GCLK
0	IO_L10N_0/GCLK11	P186	GCLK
0	IO_L10P_0/GCLK10	P185	GCLK
0	IO_L11N_0	P190	I/O
0	IO_L11P_0	P189	I/O
0	IO_L12N_0/VREF_0	P193	VREF
0	IO_L12P_0	P192	I/O
0	IO_L13N_0	P197	I/O
0	IO_L13P_0	P196	I/O
0	IO_L14N_0/VREF_0	P200	VREF
0	IO_L14P_0	P199	I/O
0	IO_L15N_0	P203	I/O
0	IO_L15P_0	P202	I/O
0	IO_L16N_0/HSWAP	P206	DUAL
0	IO_L16P_0	P205	I/O
0	IP	P159	INPUT
0	IP	P169	INPUT
0	IP	P194	INPUT
0	IP	P204	INPUT

Bank	XC3S250E XC3S500E Pin Name	PQ208 Pin	Type
0	IP_L06N_0	P175	INPUT
0	IP_L06P_0	P174	INPUT
0	IP_L09N_0/GCLK9	P184	GCLK
0	IP_L09P_0/GCLK8	P183	GCLK
0	VCC0_0	P176	VCC0
0	VCC0_0	P191	VCC0
0	VCC0_0	P201	VCC0
1	IO_L01N_1/A15	P107	DUAL
1	IO_L01P_1/A16	P106	DUAL
1	IO_L02N_1/A13	P109	DUAL
1	IO_L02P_1/A14	P108	DUAL
1	IO_L03N_1/VREF_1	P113	VREF
1	IO_L03P_1	P112	I/O
1	IO_L04N_1	P116	I/O
1	IO_L04P_1	P115	I/O
1	IO_L05N_1/A11	P120	DUAL
1	IO_L05P_1/A12	P119	DUAL
1	IO_L06N_1/VREF_1	P123	VREF
1	IO_L06P_1	P122	I/O
1	IO_L07N_1/A9/RHCLK1	P127	RHCLK/DUAL
1	IO_L07P_1/A10/RHCLK0	P126	RHCLK/DUAL
1	IO_L08N_1/A7/RHCLK3	P129	RHCLK/DUAL
1	IO_L08P_1/A8/RHCLK2	P128	RHCLK/DUAL
1	IO_L09N_1/A5/RHCLK5	P133	RHCLK/DUAL
1	IO_L09P_1/A6/RHCLK4	P132	RHCLK/DUAL
1	IO_L10N_1/A3/RHCLK7	P135	RHCLK/DUAL
1	IO_L10P_1/A4/RHCLK6	P134	RHCLK/DUAL
1	IO_L11N_1/A1	P138	DUAL
1	IO_L11P_1/A2	P137	DUAL
1	IO_L12N_1/A0	P140	DUAL
1	IO_L12P_1	P139	I/O
1	IO_L13N_1	P145	I/O
1	IO_L13P_1	P144	I/O

Tablo 3.15 Pin Yerleşim Tablosu

Bank	XC3S250E XC3S500E Pin Name	PQ208 Pin	Type
1	IO_L14N_1	P147	I/O
1	IO_L14P_1	P146	I/O
1	IO_L15N_1/LDC0	P151	DUAL
1	IO_L15P_1/HDC	P150	DUAL
1	IO_L16N_1/LDC2	P153	DUAL
1	IO_L16P_1/LDC1	P152	DUAL
1	IP	P110	INPUT
1	IP	P118	INPUT
1	IP	P124	INPUT
1	IP	P130	INPUT
1	IP	P142	INPUT
1	IP	P148	INPUT
1	IP	P154	INPUT
1	IP/VREF_1	P136	VREF
1	VCCO_1	P114	VCCO
1	VCCO_1	P125	VCCO
1	VCCO_1	P143	VCCO
2	IO/D5	P76	DUAL
2	IO/M1	P84	DUAL
2	IO/VREF_2	P98	VREF
2	IO_L01N_2/INIT_B	P56	DUAL
2	IO_L01P_2/CSO_B	P55	DUAL
2	IO_L03N_2/MOSI/CSI_B	P61	DUAL
2	IO_L03P_2/DOUT/BUSY	P60	DUAL
2	IO_L04N_2	P63	I/O
2	IO_L04P_2	P62	I/O
2	IO_L05N_2	P65	I/O
2	IO_L05P_2	P64	I/O
2	IO_L06N_2	P69	I/O
2	IO_L06P_2	P68	I/O
2	IO_L08N_2/D6/GCLK13	P75	DUAL/GCLK
2	IO_L08P_2/D7/GCLK12	P74	DUAL/GCLK
2	IO_L09N_2/D3/GCLK15	P78	DUAL/GCLK
2	IO_L09P_2/D4/GCLK14	P77	DUAL/GCLK
2	IO_L11N_2/D1/GCLK3	P83	DUAL/GCLK
2	IO_L11P_2/D2/GCLK2	P82	DUAL/GCLK
2	IO_L12N_2/DIN/D0	P87	DUAL
2	IO_L12P_2/M0	P86	DUAL
2	IO_L13N_2	P90	I/O
2	IO_L13P_2	P89	I/O

Bank	XC3S250E XC3S500E Pin Name	PQ208 Pin	Type
2	IO_L14N_2/A22	P94	DUAL
2	IO_L14P_2/A23	P93	DUAL
2	IO_L15N_2/A20	P97	DUAL
2	IO_L15P_2/A21	P96	DUAL
2	IO_L16N_2/VS1/A18	P100	DUAL
2	IO_L16P_2/VS2/A19	P99	DUAL
2	IO_L17N_2/CCLK	P103	DUAL
2	IO_L17P_2/VS0/A17	P102	DUAL
2	IP	P54	INPUT
2	IP	P91	INPUT
2	IP	P101	INPUT
2	IP_L02N_2	P58	INPUT
2	IP_L02P_2	P57	INPUT
2	IP_L07N_2/VREF_2	P72	VREF
2	IP_L07P_2	P71	INPUT
2	IP_L10N_2/M2/GCLK1	P81	DUAL/GCLK
2	IP_L10P_2/RDWR_B/ GCLK0	P80	DUAL/GCLK
2	VCCO_2	P59	VCCO
2	VCCO_2	P73	VCCO
2	VCCO_2	P88	VCCO
3	IO/VREF_3	P45	VREF
3	IO_L01N_3	P3	I/O
3	IO_L01P_3	P2	I/O
3	IO_L02N_3/VREF_3	P5	VREF
3	IO_L02P_3	P4	I/O
3	IO_L03N_3	P9	I/O
3	IO_L03P_3	P8	I/O
3	IO_L04N_3	P12	I/O
3	IO_L04P_3	P11	I/O
3	IO_L05N_3	P16	I/O
3	IO_L05P_3	P15	I/O
3	IO_L06N_3	P19	I/O
3	IO_L06P_3	P18	I/O
3	IO_L07N_3/LHCLK1	P23	LHCLK
3	IO_L07P_3/LHCLK0	P22	LHCLK
3	IO_L08N_3/LHCLK3	P25	LHCLK
3	IO_L08P_3/LHCLK2	P24	LHCLK
3	IO_L09N_3/LHCLK5	P29	LHCLK
3	IO_L09P_3/LHCLK4	P28	LHCLK
3	IO_L10N_3/LHCLK7	P31	LHCLK

Tablo 3.15 Pin Yerleşim Tablosu (Devam)

Bank	XC3S250E XC3S500E Pin Name	PQ208 Pin	Type
3	IO_L10P_3/LHCLK6	P30	LHCLK
3	IO_L11N_3	P34	I/O
3	IO_L11P_3	P33	I/O
3	IO_L12N_3	P36	I/O
3	IO_L12P_3	P35	I/O
3	IO_L13N_3	P40	I/O
3	IO_L13P_3	P39	I/O
3	IO_L14N_3	P42	I/O
3	IO_L14P_3	P41	I/O
3	IO_L15N_3	P48	I/O
3	IO_L15P_3	P47	I/O
3	IO_L16N_3	P50	I/O
3	IO_L16P_3	P49	I/O
3	IP	P6	INPUT
3	IP	P14	INPUT
3	IP	P26	INPUT
3	IP	P32	INPUT
3	IP	P43	INPUT
3	IP	P51	INPUT
3	IP/VREF_3	P20	VREF
3	VCCO_3	P21	VCCO
3	VCCO_3	P38	VCCO
3	VCCO_3	P46	VCCO
GND	GND	P10	GND
GND	GND	P17	GND
GND	GND	P27	GND
GND	GND	P37	GND
GND	GND	P52	GND
GND	GND	P53	GND
GND	GND	P70	GND
GND	GND	P79	GND
GND	GND	P85	GND
GND	GND	P95	GND
GND	GND	P105	GND
GND	GND	P121	GND
GND	GND	P131	GND
GND	GND	P141	GND
GND	GND	P156	GND
GND	GND	P173	GND
GND	GND	P182	GND

Bank	XC3S250E XC3S500E Pin Name	PQ208 Pin	Type
GND	GND	P188	GND
GND	GND	P198	GND
GND	GND	P208	GND
VCCAUX	DONE	P104	CONFIG
VCCAUX	PROG_B	P1	CONFIG
VCCAUX	TCK	P158	JTAG
VCCAUX	TDI	P207	JTAG
VCCAUX	TDO	P157	JTAG
VCCAUX	TMS	P155	JTAG
VCCAUX	VCCAUX	P7	VCCAUX
VCCAUX	VCCAUX	P44	VCCAUX
VCCAUX	VCCAUX	P66	VCCAUX
VCCAUX	VCCAUX	P92	VCCAUX
VCCAUX	VCCAUX	P111	VCCAUX
VCCAUX	VCCAUX	P149	VCCAUX
VCCAUX	VCCAUX	P166	VCCAUX
VCCAUX	VCCAUX	P195	VCCAUX
VCCINT	VCCINT	P13	VCCINT
VCCINT	VCCINT	P67	VCCINT
VCCINT	VCCINT	P117	VCCINT
VCCINT	VCCINT	P170	VCCINT

Tablo 3.15 Pin Yerleşim Tablosu (Devam)

4. DS92LV18 18-Bit Bus LVDS Serializer/Deserializer

4.1 Genel Tanım

DS92LV18 Paralel-seri/Seri-paralel dönüştürücü (SERDES) çifti bir BLVDS içindeki 18-bit paralel veriyi gömülü saat bilgisiyle beraber seri veri dizisine çevirmektedir. Bu tek seri dizi kablo ve PCB yolları üzerindeki 18-bit (yâda daha az) verinin transferini saat yolu ve paralel veri arasındaki eğri (skew) problemlerini yok ederek basitleştirmektedir. Veri yolunun daraltılması ile PCB katman sayısı, kablo genişliği, konektör boyu ve pinlerini azaltarak sistem maliyetinden kazanım sağlamaktadır.

Bu SERDES çifti gömülü sistem ve aygıt test kabiliyetine sahiptir. Hat geri-çevrim özelliği kullanıcının alıcı çıkışında seri veriden paralel veriye çevrim yapılırken alıcı ve vericinin seri veri iletişim yolunun güvenilirliğini kontrol edebilmesini sağlar. Yerel geri-çevrim özelliği kullanıcının yerel paralel-veriyolu tarafında alıcının güvenilirliğini kontrol edebilmesini sağlar.

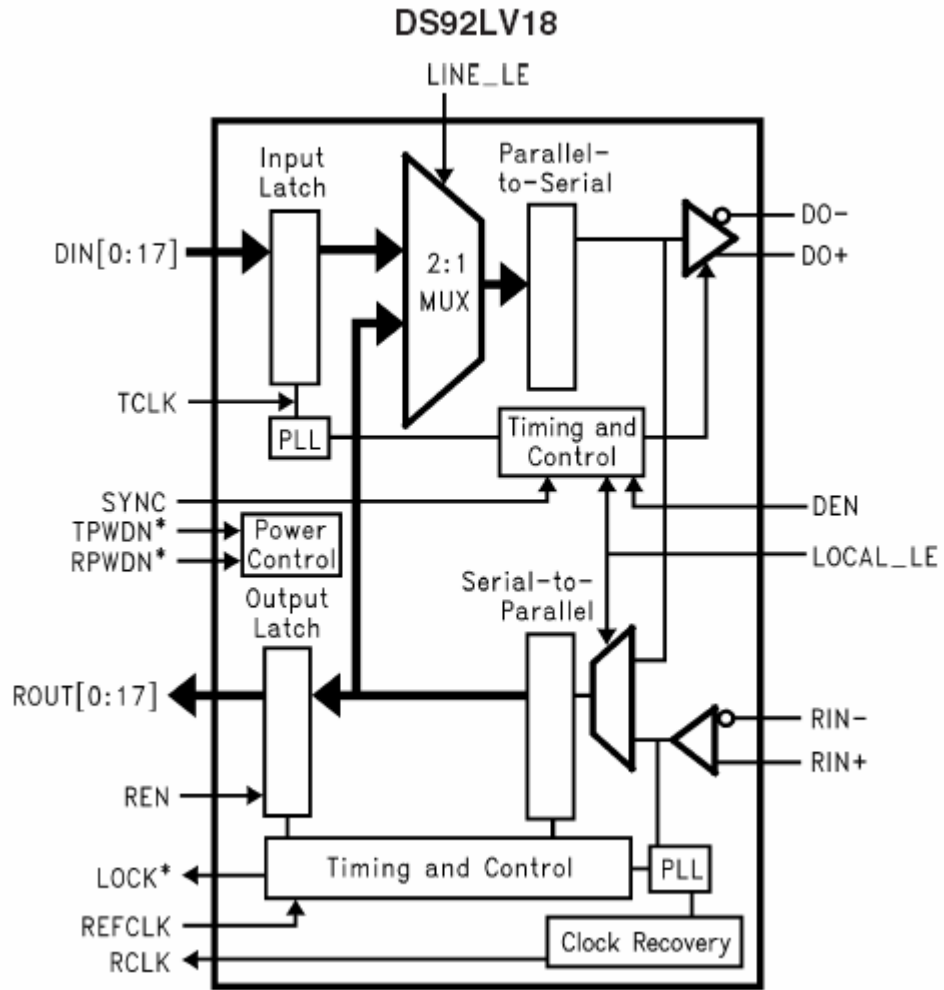
DS92LV18 değiştirilmiş BLVDS işaretlenmesini yüksek-hızlı I/O üzerine dâhil etmektedir. BLVDS, bir seri iletişim yolu üzerinde güvenilir veri transferi için düşük güç ve düşük gürültülü ortam sağlamaktadır. Fark veri yolu içerisinden geçen eş ve karşıt akımlar kenar alan sonuçlarını birleştirerek EMI'yi kontrol etmektedir.

Özellikleri

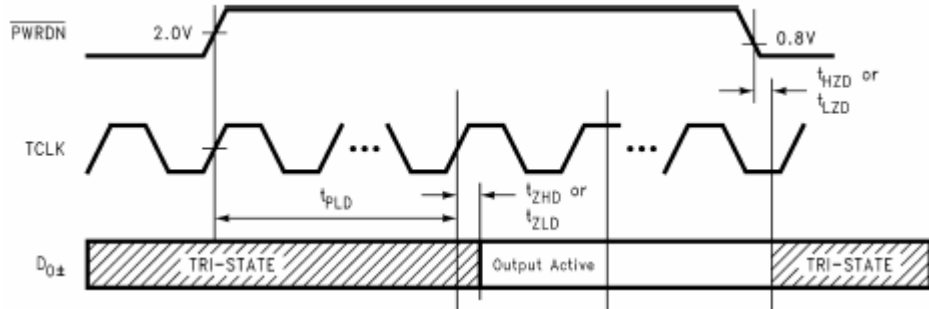
- 15–66 MHz 18:1/1:18 Paralel-Seri/Seri-Paralel dönüştürücü (2.376 Gbps çift yönlü iletişim hızı)
- Ayrık saat, yetkilendirme ve güç kesme pinleri ile beraber bağımsız alıcı ve verici işlemleri
- Yerel olarak oluşturulan saat sinyalinin kullanan kolay sistem tasarımı için geniş \pm %5 referans saat frekansı toleransı
- Hat ve Yerel geri-çevrim modları
- Düşük EMI için backplane ve kablolar içinden sağlam BLVDS seri iletişim

- Harici kodlama gerektirmez
- Dâhili PLL'e sahiptir, ayrıca harici PLL bileşenleri gerektirmez
- Tek +3.3V güç kaynağı
- Düşük güç: PRBS-15 Paternli 66MHz sinyal ile 90mA (tipik olarak) verici
- ± 100 mV alıcı giriş eşiği
- Kilitlenme kaybı tespit ve raporlama pini
- Endüstriyel -40 to $+85^{\circ}\text{C}$ sıcaklık aralığı
- $>2.0\text{kV}$ HBM ESD
- Küçük ve standart 80-pin LQFP kılıf seçenekleri

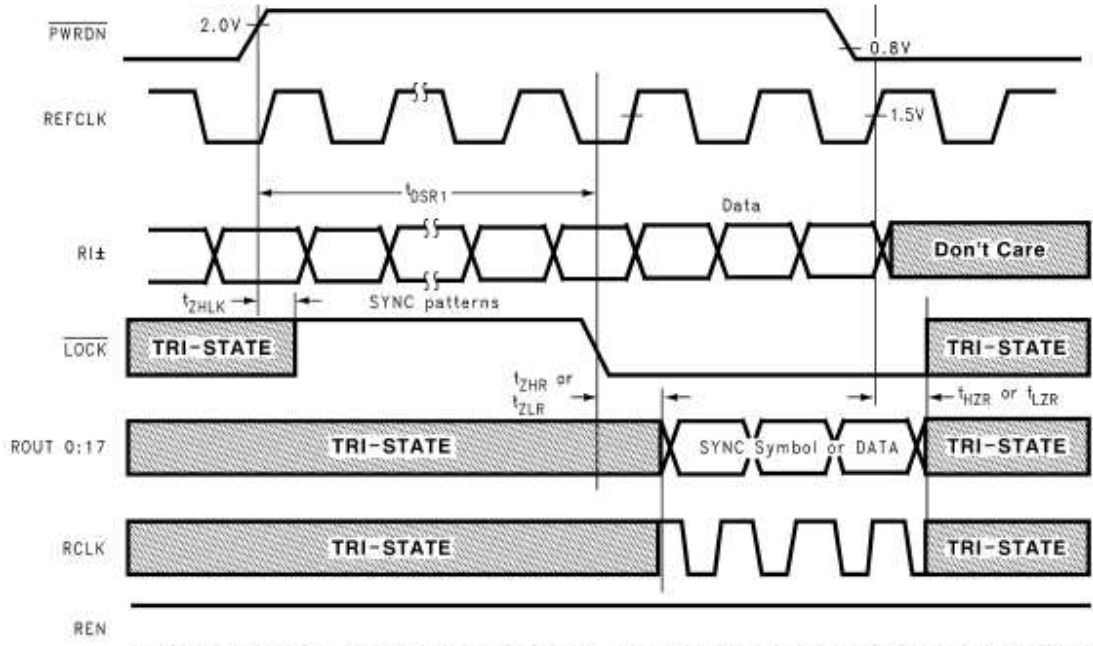
Aşağıdaki şekillerde DS92LV18'in blok yapısı ve SER/DES gecikmeleri görülmektedir.



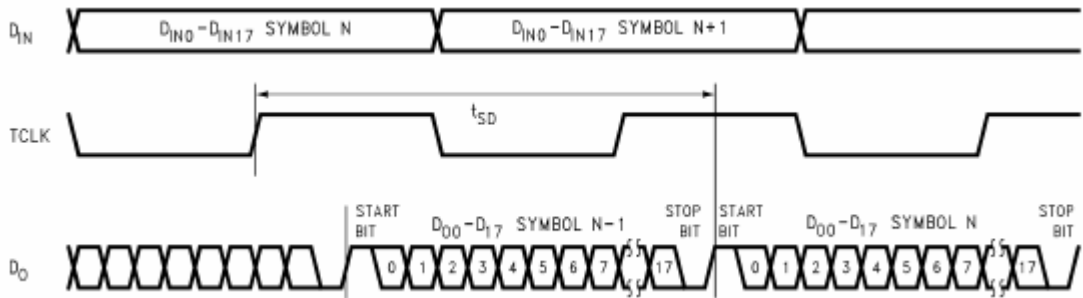
Şekil 4.1 DS92LV18 SERDES Blok Şeması



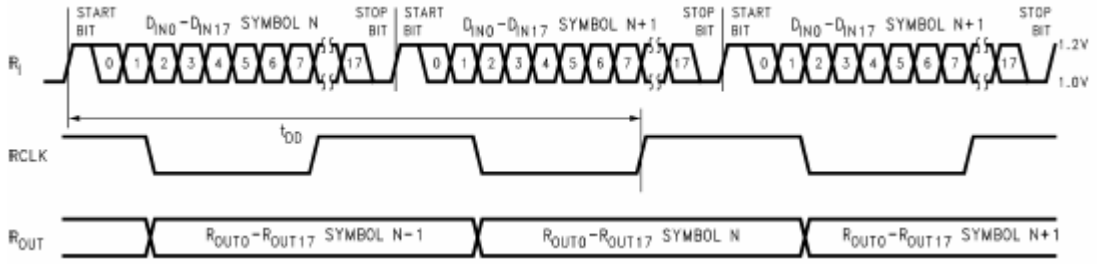
Şekil 4.2 Paralel-seri Dönüştürücü PLL Kilitlenme Süresi ve PWRDN, 3-STATE Gecikmeleri



Şekil 4.3 Seri-paralel Dönüştürücü PLL Kilitlenme Süresi ve PWRDN, 3-STATE Gecikmeleri



Şekli 4.4 Paralel-seri Dönüştürücü Gecikmesi



Şekli 4.5 Seri-paralel Dönüştürücü Gecikmesi

4.2 İşlevsel Tanım

DS92LV18 paralel-seri ve seri-paralel dönüştürücüyü tek bir yonga üzerinde birleştirmektedir. Paralel-seri dönüştürücü 18-bitlik bir LVCMOS veya LVTTL veri yolunu alır ve bunu gömülü saat bilgisi ile beraber seri veri dizisi şekline dönüştürür. Sonra seri-paralel dönüştürücü saat ve veriyi yeniden elde ederek 18-bit genişliğindeki sonucu çıkışa verir.

Aygıt birbirlerinden bağımsız olarak çalışabilen ayrık alıcı ve verici bloğuna sahiptir. Her biri çeşitli uygulamalarda verimli işlemlere olanak sağlamak için güç kesme kontrolüne sahiptirler. Örnek olarak alıcı-verici gereksiz bir veri yolunda beklemede işlem yapabilir fakat yinede gücünü korur. Bu eleman paralel-seri çevirici, seri-paralel çevirici veya iki-yönlü SER/DES olarak yapılandırılabilir.

DS92LV18 alıcı ve verici bloklarının her biri üç çalışma durumuna sahiptir. Bunlar Başlatma, Veri Transferi ve Yeniden senkronize olma durumları. Ek olarak iki pasif durum da vardır: Güç-kesme ve 3-STATE. Aşağıdaki bölümde her bir işlem modu ve pasif durumu açıklanmaktadır.

Başlatma

DS92LV18 veri almadan yâda göndermeden önce diğer DS92LV18'den gelen yâda giden bağlantıları başlangıç durumuna getirmelidir. Başlatma, paralel-seri ve seri-paralel dönüştürücünün PLL'inin yerel saatlerinin senkronizasyon için kilitlenmesi olayıdır. Yerel saatler aynı frekansta yâda farklı kaynaklar ise belirli sınırlar içerisinde olmalıdır. Paralel-seri dönüştürücüler yerel saatle eşleştikten sonra ikinci

ve son başlatma adımı olarak seri-paralel dönüştürücüler paralel-seri dönüştürücülerle senkronize olur.

Adım 1: VCC, Paralel-seri ve/veya Seri-paralel dönüştürücülere uygulandığında bunların çıkışları 3-STATE durumundadır ve dâhili devre yonga-üzeri güç-açma devresi aracılığıyla kapalı tutulur. VCC, VCC OK (2.2V) gerilimine ulaştığında her bir aygıttaki PLL yerel saate kilitlenmeye çalışmaya başlar. Paralel-seri çevirici için yerel saat, gönderilen saat olan TCLK'dır. Seri-paralel dönüştürücü için yerel saat ise REFCLK pinine uygulanmaktadır. Yerel bir kart-üzerindeki osilatör yâda diğer bir kaynak TCLK ve REFCLK pinine belirli saat girişi sağlamaktadır.

Paralel-seri dönüştürücü çıkışları PLL, TCLK'ya kilitlenene kadar 3-STATE durumundadır. TCLK'ya kilitlendikten sonra Paralel-seri dönüştürücü bloğu artık veri yâda senkronizasyon paterni göndermeye hazırdır. SYNC pini yüksek seviyeye geldikten sonra Paralel-seri dönüştürücü bloğu senkronizasyon paternleri (sync-pattern) oluşturur ve gönderir.

Seri-paralel dönüştürücü çıkışı PLL, REFCLK'ya kilitlenene kadar 3-STATE durumunda kalacaktır. Aynı zamanda Seri-paralel dönüştürücü LOCK çıkışı da PLL'i RIN pinlerinden gelen bir sync-pattern yâda veriye kilitlenene kadar yüksek durumda kalacaktır.

Adım 2: Seri-paralel dönüştürücü PLL'i başlatma durumunu tamamlamak için Paralel-seri dönüştürücüye senkronize olmalıdır. Seri-paralel dönüştürücüye olan veri akışını üreten Paralel-seri dönüştürücü bu Başlatma Durumu adımı süresince rasgele (tekrar etmeyen) veri paterni yâda sync-patternleri göndermelidir. Seri-paralel dönüştürücü belirli bir zaman zarfında sync-patternleri ile kilitlenecektir. Rasgele veriye kilitlenme, veri paternine bağlıdır ve bu yüzden kilitlenme süresi belirlenmemektedir.

Gelen LVDS veri dizisine kilitlenmek için Seri-paralel dönüştürücü sync-patternindeki saatin yükselen kenarlarını saptar ve buna kilitlenir. Seri-paralel dönüştürücü, Paralel-seri dönüştürücüden gelen rasgele veri dizisine kilitlenmeye

çalışırken saatin yükselen kenarını saptamak için bir dizi işlem gerçekleştirir ve kilitlenir. Çünkü bu kilitlenme yordamı veri paternine bağlıdır ve bunun ne kadar süreceğini saptamak mümkün değildir. Seri-paralel dönüştürücünün PLL'i gömülü saate kilitlendiği noktada LOCK pini düşük seviyeye gelir ve geçerli veri çıkışta görülür.

Veri Transferi

Başlatma işleminden sonra, DS92LV18 Paralel-seri dönüştürücü verileri Seri-paralel dönüştürücüye aktarabilmektedir. Seri veri dizisi paralel-seri dönüştürücü tarafından eklenen on sekiz veri bitlerini belirten başla ve dur bitlerini içermektedir. Başla biti her zaman yüksek, dur biti de her zaman düşük seviyedir. Başla ve dur bitleri aynı zamanda seri dizi içerisinde gömülü saat bitleri olarak işlev görmektedir.

Paralel-seri dönüştürücü bloğu veriyi DIN0-DIN17 paralel girişlerinden almaktadır. TCLK sinyali yükselen kenarda gelen veriyi tutar. SYNC girişi 6 TCLK çevrimi kadar yüksek seviyede ise DS92LV18 DIN0-DIN17'den gelen veriyi tutmaz.

Paralel-seri dönüştürücü TCLK frekansında 20 kerede veri ve saat bitlerini (18+2 bit) göndermektedir. Örnek olarak TCLK 60MHz ise seri iletim oranı $60 \times 20 = 1200$ Mbps'dir. Saat bitlerini çıkardığımızda gerçek veri transfer hızı $60 \times 18 = 1080$ Mbps'dir. TCLK veri kaynağı tarafından sağlanır ve 15MHz ile 66MHz aralığı içerisinde olmalıdır.

Seri-paralel dönüştürücü kanalı bir Paralel-seri dönüştürücüden gelen girişle senkronize olduğunda LOCK pinini düşük seviyeye çeker ve eşzamanlı olarak geçerli veriyi çıkışa verir. Seri-paralel dönüştürücü gömülü saate kilitlenir ve bunu çoklu dâhili veri stroboskobları oluşturmak için kullanır ve sonra yeniden elde edilen saat sinyalini RCLK'ya verir. Yeniden elde edilen saat (RCLK çıkış pini) ROUT[0:17] pinlerindeki veri ile senkronizedir. LOCK düşük seviyede olduğu sürece ROUT[0:17] üzerindeki veri geçerlidir, aksi takdirde veri geçersizdir.

ROUT[0:17], LOCK ve RCLK sinyalleri 66 MHz saat hızında en az üç CMOS giriş kapısı (15pF toplam yük) sürmektedir. Bu sürme kapasitesi çoklu Seri-paralel dönüştürücü çıkışlarının çoklu hedef ASIC girişlerine sürülmesini yani veri yolu kullanımına olanak sağlamaktadır. REN Seri-paralel dönüştürücüdeki ROUTn ve RCLK pini için 3-STATE durumunu denetler.

Seri-paralel dönüştürücü giriş pinleri alıcı güç-kesme (RPWDN='0') ve güç-kapalı (VCC=0) durumları süresince yüksek empedanstadır.

Tekrar Senkronizasyon

Seri-paralel dönüştürücü kilitli olma konumunu kaybederse otomatik olarak yeniden senkronize olmayı deneyecektir. Örnek olarak eğer gömülü saat kenarı art arda iki kez alınmazsa, PLL kilitlenmeyi kaybeder ve LOCK pini yüksek seviyeye çekilir. Sonra Seri-paralel dönüştürücü bir rasgele veri dizisine kilitlenmeyi denediği işlem moduna girecektir. PLL, gömülü saat kenarını arar ve bunu tespit eder, sonra senkronizasyon yordamında ilerler.

LOCK sinyalinin lojik durumu, ROUT üzerindeki verinin geçerli olup olmadığını belirtir. Sistem, ROUT üzerindeki verinin geçerli olup olmadığını belirlemek için LOCK pinini gözlemlemelidir. Çünkü LOCK sinyalinin PLL'in gelen veri dizisi ile senkronizasyonu kaybetmesine olan tepkisinde kısa bir gecikme vardır. Sistem LOCK sinyali yüksek seviyeye gelmeden önceki çevrimler için verinin doğruluğunu saptaması gerekmektedir.

Güç-Kesme

Güç-kesme durumu paralel-seri ve seri-paralel dönüştürücünün başlatma işlemi için beklerken meşgul eden bir düşük güç uyku modudur. Aynı zamanda askıda kalan bir veri transferi yokken gücü azaltmak için TPWDN ve RPWDN'ı kullanabilirsiniz. Seri-paralel çevirici RPWDN düşük seviyeye sürüldüğünde güç-kesme moduna girer. Güç-kesem modunda PLL durur ve çıkışlar kaynak akımını μA seviyesine kadar azaltan 3-STATE konumuna girer. Seri-paralel dönüştürücü bloğunu Güç-

kesme modundan çıkarmak için sistem RPDWN'ı yüksek seviyeye çeker. Seri-paralel dönüştürücü güç-kesme modundan çıktığında, otomatik olarak Başlatma durumuna girer. Sistem, sonra veri transferi başlamadan önce Başlatma için gerekli olan zamana olanak sağlamalıdır.

TPWDN pini Paralel-seri dönüştürücü bloğunun akım kaynağının μA mertebesinde olduğu düşük güç tüketimine girmesine zorlamak için düşük seviyeye sürülür. Paralel-seri dönüştürücünün PLL'i durur ve çıkışlar 3-STATE konumuna geçer.

Paralel-seri dönüştürücü bloğunu Güç-kesme modundan çıkarmak için sistem RPDWN'ı yüksek seviyeye çeker. Paralel-seri dönüştürücü güç-kesme modundan çıktığında, Başlatma durumu için hazır olmadan önce PLL'i, TCLK'ya kilitlenmelidir. Sistem, sonra veri transferi başlamadan önce Başlatma için gerekli olan zamana olanak sağlamalıdır.

3-STATE

Sistem REN pinini düşük seviyeye sürdüğünde, Seri-paralel dönüştürücünün çıkışları 3-STATE durumuna girer. Bu RCLK ve alıcı çıkış pinlerini 3-STATE konumuna alacaktır. Sistem REN pinini yüksek seviyeye sürdüğünde, Seri-paralel dönüştürücü tüm diğer kontrol pinleri pasif (RPWDN) kaldığı sürece önceki konumuna dönecektir.

Sistem DEN pinini düşük seviyeye sürdüğünde, Seri-paralel dönüştürücünün LVDS çıkışları 3-STATE durumuna girer. Sistem DEN pinini yüksek seviyeye sürdüğünde, Seri-paralel dönüştürücü tüm diğer kontrol ve veri giriş pinleri DEN düşük seviyeye sürülmeden önce aynı koşullar içerisinde pasif kaldığı sürece önceki konumuna dönecektir.

Geri-çevrim Test İşlemi

DS92LV18 aygıt işlevselliği ve iletim hattı sürekliliğini test etmek için iki Geri-çevrim modunu barındırmaktadır. Hat Geri-çevrim kontrol sinyalini aktif ederek seri

veri giriři (RIN \pm) seri veri ıkıřına (DO \pm) ve paralel veri ıkıřına (ROUT[0:17]) baęlanmaktadır. Seri veri, Paralel-seri ve Seri-paralel donüřtürücü blokları ierisinden geer.

Yerel Geri-evrim kontrol sinyalini aktif ederek paralel veri giriři (DIN[0:17]), paralel veri ıkıřına (ROUT[0:17]) geri baęlanmaktadır. Baęlantı yolu SER/DES ifti tüm iřlevsel bloklarını iermektedir. Seri veri ıkıřı (DO \pm), Yerel Geri-evrim iřlem modunda otomatik olarak pasif yapılır.

5. VHDL'E GİRİŞ

5.1 VHDL NEDİR?

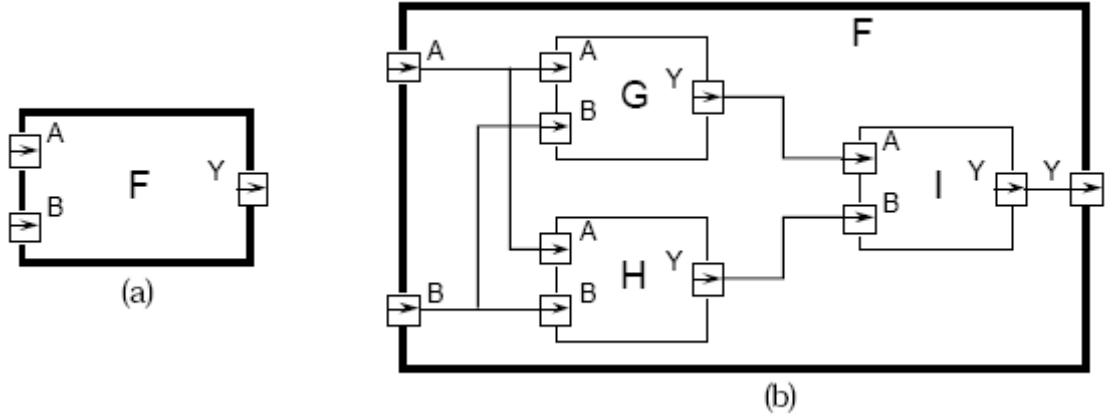
VHDL sayısal elektronik sistemleri tanımlamak ve bir FPGA içine gömmek için kullanılan bir programlama dilidir. Bu dil Birleşik Devletler Hükümeti'nin Çok Yüksek Hızlı Tümdevreler (Very High Speed Integrated Circuits-VHSIC) programı çerçevesinde 1980 yılında başlatıldı. Program çerçevesinde geliştirme çalışmaları sürerken tümdevrelerin (ICs) yapılarını ve fonksiyonlarını tanımlamak için standart bir programlama diline ihtiyaç duyulduğu anlaşıldı. Bu sebepten dolayı VHSIC Hardware Description Language (VHDL) geliştirildi.

VHDL 1986'da IEEE standardı olması için IEEE' ye teklif sunuldu. 1987'nin Aralık'ında IEEE 1076 standardı olarak kabul edilmesine kadar VHDL birçok revizyona uğradı.

5.2 VHDL YAPISI

Birinci olarak bir tasarımın yapısı tamamen VHDL ile tanımlanabilir. Bu tasarımı alt tasarımlara ayrıştırır ve bu alt tasarımların birbirleriyle bağlantıları sağlanabilir. İkinci olarak bu dil yakın bir programlama dili olan dillerle hazırlanmış fonksiyonların kullanılmasına izin verir. Üçüncü olarak yapılan tasarımlar işlevi olan sistemde kullanılmadan önce simülasyonu yapılabilir. Tasarımcılar hızlı bir şekilde alternatifleri geliştirip bir prototip olmadan bu alternatiflerin doğruluklarını sınavabilirler.

5.2.1 Yapısal Tanımlama



Şekil 5.1 Bir yapı tanımlama örneği

Bir sayısal elektronik sistemi giriş ve/veya çıkışlarıyla beraber bir parça olarak tanımlanabilir. Çıkışın elektriksel değerleri girişin elektriksel değerlerinin bir ve birkaç fonksiyonu olarak karşımıza çıkarlar. Şekil 5.1.a'da bir sayısal sistemin modül halinde gösterimi mevcuttur. Görüldüğü üzere F modülünün A ve B olmak üzere iki girişi ve bir Y çıkışı vardır. VHDL terminolojisi kullanılarak F bir tasarım (entity) olarak düşünülürse girişler ve çıkışlar port diye nitelendirilebilir.

Modüle işlevi kazandırmanın bir yolu modülü alt modüllere ayırmaktır. Bu modüllerin portlarını oluşturduktan sonra signal tipi yapılarla birbirleriyle haberleştirmektir. Şekil 5.1.b'de bir bütün yapının daha küçük yapılara nasıl ayrıldığı gösterilmiştir. Burada F modülü G, H ve I alt modüllerine ayrıştırılıp bunların istenen fonksiyonu gerçekleştirecek konfigürasyonunu yapmak gerekecektir. Bu tür tanımlamaya "yapısal tanımlama" denir. F bir yapısal tanımlama olduğu gibi G, H ve I alt modülleri de yapısal tanımlama olabilir.

5.2.2 Davranışsal Tanımlama

Birçok durumda bir işlevi yapısal olarak tanımlama uygun olmayabilir. Bazen bir modül diğer yapısal tanımlanmış bir hiyerarşinin en alt tabanında olabilir. Bazen

sadece satıcıdan alınan tümdevrelerle sistem tasarlanmak istenebilir. Belki de içsel bir yapıya ihtiyaç duyulmayan fonksiyonlar gerçekleşmek istenebilir. Bu tür tanımlamaları yapmak için “fonksiyonel” veya “davranışsal tanımlama” adı verilen bir tanımlama tipi ile tanımlanırlar. Yani bu tip tanımlamalarda tasarımın yapısını tanımlamadan davranışını tanımlamak yeterlidir. Bunu göstermek için daha önce şekil 5.1’deki örneği tekrar ele almak gerekecektir. Burada F fonksiyonunu dışaran-veya (EXOR) kapısı olarak düşünülebilir.

$$Y = \overline{A} \cdot B + A \cdot \overline{B}$$

Daha karmaşık davranışlar sadece girişin fonksiyonu olarak tanımlanamayabilirler. Geri beslemeli sistemlerde ayrıca çıkışlar da zaman düzleminde bir fonksiyondurlar.

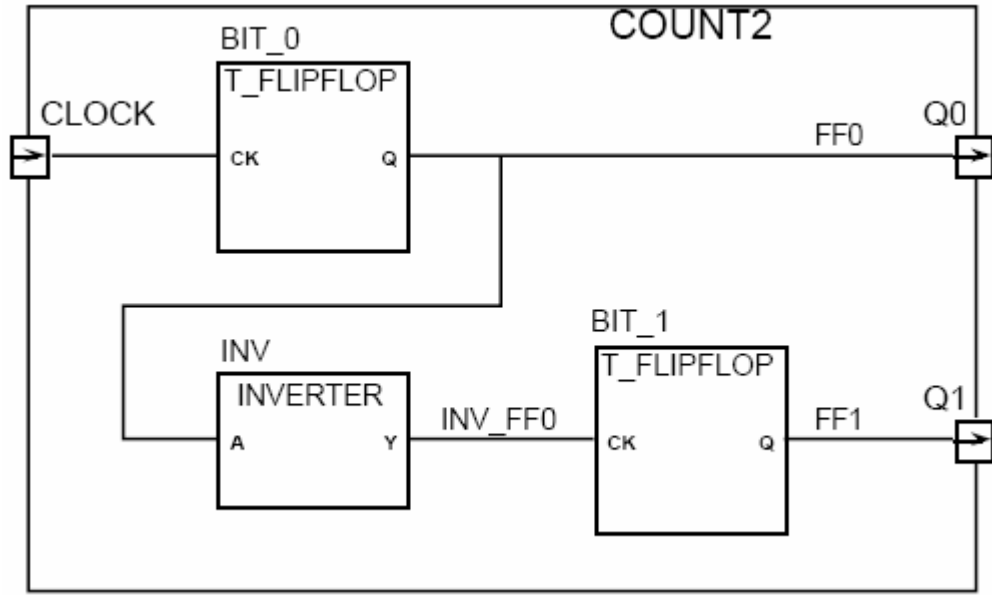
Özet olarak biz bir tasarımda gerektiği yerde yapısal tanımlama gerektiği yerde ise davranışsal tanımlama kullanıyoruz.

5.2.3 Özgül Bir VHDL Örneği

Burada basit ve tipik bir VHDL örneği incelenecektir. Bu örnekte iki bitlik bir sayıcının VHDL ile tanımlaması verilmiştir. Bu örnek en azından bu programlama dilinin nasıl kullanılacağı hakkında bir ön fikir verme açısından önemlidir. Burada tasarım tanımlanmaya, program için dışarıyla bağlantı sağlayan ve dışsal ara yüz görevi gören portları tanımlamakla başlanıyor.

```
entity count2 is  
  generic (prop_delay : Time := 10 ns);  
  port (clock : in bit;  
        q1, q0 : out bit);  
end count2;
```

Yukarıdaki program parçasında count2 diye bir tasarımın varlığından söz ediliyor. Bunun yapısında clock isminde bir giriş ve q1,q2 isminde iki çıkış olduğu bildiriliyor. Bunların veri tipinin de bit olduğu belirtiliyor. Ayrıca time tipinden generic, ilk değeri 10ns olan, prop_delay (yayıma zamanı) tanımlanmaktadır.



Şekil 5.2 Count2 tasarımının yapısı

Buradaki yayılma zamanının 10 ns olması verilen ilk değerdir(default). Bunun yerine daha sonra yeniden yayılma zamanı hesaplanıp değiştirilmediği sürece bu değer kullanılacağı anlamına gelir.

Dışsal ara yüz tanımlandıktan çıkışların girişin hangi fonksiyonu olacağı programın mimari(architecture) kısmında gerçekleştirilir. Bir sistemin birden çok mimarisi aynı tasarım içerisinde tanımlanabilir. Bunların her biri sistem için farklı bir bakış alanı oluştururlar. Mesela yukarıda blok şeması gösterilen sayıcının tasarımı şöyle tanımlanabilir. Dikkat edilirse bu bir davranışsal tanımlamadır ve blok şema ile alakası yoktur.

```

architecture behaviour of count2 is
begin
    count_up: process (clock)
        variable count_value : natural := 0;
    begin
        if clock = '1' then
            count_value := (count_value + 1) mod 4;
            q0 <= bit'val(count_value mod 2) after prop_delay;
            q1 <= bit'val(count_value / 2) after prop_delay;
        end if;
    end process count_up;
end behaviour;

```

Sayıcı tanımlanırken sistemin davranışı, parametresi saat darbesi olan count_up ismiyle etiketlenmiş bir process ile sağlanmıştır. Bu process'in içindeki program parçası ancak işretlerin değer değişimi ile yürütülebilir. Yani program parçasının çalışması için saat darbesi gereklidir. Ancak bu durumda periyodik bir değer değişimi gözlenir ve program yürütülür. Count_value adlı değişkeni ise saydırmayı kontrol amaçlı kullanılmıştır. Zaten değişkenler sadece processler içinde kullanılabilir. Bunun nedeni ileride açıklanacaktır.

İki bitlik sayıcı ayrıca şekil 5.2'de gösterildiği gibi T-flip-flopları ve bir evirici alt modüllerinden oluşmuş yapısal tanımlamayla tanımlanabilir. Bu VHDL dilinde şöyle yazılabilir:

```
architecture structure of count2 is
    component t_flipflop
        port (ck : in bit; q : out bit);
    end component;

    component inverter
        port (a : in bit; y : out bit);
    end component;

    signal ff0, ff1, inv_ff0 : bit;
begin
    bit_0 : t_flipflop port map (ck => clock, q => ff0);
    inv : inverter port map (a => ff0, y => inv_ff0);
    bit_1 : t_flipflop port map (ck => inv_ff0, q => ff1);
    q0 <= ff0;
    q1 <= ff1;
end structure;
```

Dikkat edilirse bu mimaride iki tip bileşen(component) tanımlanmıştır. Bunlar T-flip-flopları ve eviricilerdir. Her component giriş ve çıkışlarıyla kendi başına bir sistemmiş gibi tanımlanır. İlk başta bunu anlamak için diğer programlama dillerinde fonksiyon veya prosedür gibi düşünebiliriz. Fakat ileride anlayacağımız üzere tamamen farklıdır çünkü VHDL'de de kendine ait prosedür ve fonksiyonları mevcuttur.

5.3 VHDL Terimleri

Öncelikle yazılacak programlarda karşımıza çıkacak olan terimleri incelememiz gerekmektedir. Bu terimler yazılan her temel programda genelde bulunur. Bunları anlamak programın yapısını anlamak ve yeni yapılar oluşturmak için önemlidir.

- **Entity.** Kelime olarak anlamı mevcudiyet, varlık ve vücuttur. Bütün tasarımlar entity'lerle ifade edilir.
- **Architecture.** Kelime anlamı olarak mimaridir. Yukarıda tanımlanan programın vücudu yani temel program içinde barındırdığı mimariye göre simülasyonu yapılır. Mimari (architecture) programın yani yapının davranışını belirler. Bir temel yapı (entity) içinde birkaç adet mimari olabilir. Mesela bir mimari, davranışı belirlerken diğeri de yapıyı belirleyebilir.
- **Configuration.** Bu deyim genel olarak programda kullanılacak konfigürasyonu belirtir. Configuration tasarım için bir parça listesi olarak tanımlanabilir. Bu hangi **entity** hangi özelliklerin kullanılacağını ifade eder. Genel olarak parçaların nerede ve nasıl kullanılacağını belirten bir mönü gibi ifade edilebilir.
- **Package.** Kelime anlamı olarak pakettir. Programı yazarken kullanılacak olan alt programlar ve veri tipleri bu yapıyla paketlenir. Bunu tasarım yaparken kullanacağımız bir araç kutusu (toolbox) gibi düşünebiliriz.
- **Driver.** Kelime anlamı olarak sürücüdür. Burada sinyal kaynağı anlamında kullanılmıştır. Eğer sinyalin iki kaynağı varsa ve her iki kaynak da aktifse kaynağın iki sürücüsü var denir. Diğer bir deyimle veri sağlayıcılarına sürücü
- **Bus.** Bus genel olarak akla veri yolu kavramını getirir. Ama VHDL de bus özel bir işarettir ki bu işaretin kaynağı sönüktür. Kullanım itibarıyla yine yol özelliği taşır.
- **Attribute.** Kelime anlamı olarak öznitelik, özelliktir. Biz VHDL de kendimiz yeni veri tipleri tanımlarız. Bunu yaparken veri tipinin özelliklerini kendimiz belirleriz. Aynı durum nesne tanımlarken de geçerlidir. Biz nesneyi oluştururken onun herhangi bir durum karşısında onun davranışını tanımlarız yani ona bir özellik kazandırırız.

- **Generic.** Bu VHDL terimi bir entity'in içine bilgi gönderebilmek için kullanılan bir parametredir. Bunu da tanımlarken veri tipini belirtmek gerekir. Mesela entity
İnen kenar ve çıkan kenar gecikmeli bir kapı seviye modeli ise inen kenar ve çıkan kenar arasındaki gecikmeyi biz entity'e ancak Generic tipi parametrelerle bildiririz.
- **Process.** VHDL'in temel yürütme birimlerinden birisidir. Genelde biz bunu sıralı program (for, if kullanılarak yazılan program) yazarken kullanırız. Çünkü ilerde anlatacağımız üzere VHDL normal dillerde olduğu gibi programı satır satır yürütmez. Program satırları hepsi birden paralel olarak çalışır. FPGA işlemcilerin hızı da buradan kaynaklanmaktadır. Ama paralel çalışan bir programda if deyimi ve for döngüsü gibi yapıların kullanılması olanaksızdır. Bu yüzden bu yapıları kullanmak için biz Process yapısını kullanırız. Process ile tanımlı program parçaları normal programlar gibi satır satır çalışır.

5.4 Veri Tipleri

VHDL 'de birçok veri tipi bulunur. Biz öncelikle nesnel tiplerden bahsedeceğiz. Daha sonra skaler sayılardan bileşik dizilere, yapılardan dosya tiplerine kadar birçok tipin incelenmesi yapılacaktır. Bu bölüm nesne türleri ve veri tipleri adında iki başlıkla incelenecektir. VHDL gerçek hayattaki nesnelere benzer nesnelere vardır. Bunlar bazen bir veri yolu bazen bir değişken olabilir. Birinci bölümde bunlar incelenecektir. Tabi ki bunların kullanım yerleri farklıdır.

5.4.1 Nesne Tipleri (*Object Types*)

Programlama dilleri bildiğimiz gibi yapısal ve nesneye yönelik olmak üzere ikiye ayrılır. Yapısal dillere örnek olarak hepimizin duyduğu ya da kullandığı C, Pascal, Ada, Fortran... verilebilir. Nesnel diller dillerde nesnelere gerçek hayattaki nesnelere model alınarak oluşturulur. Gerçek hayatta her nesnenin kendisine uygulanan etkilere göre bir davranışı veya bir işlevi vardır. Nesnel programlarda yapılan iş bundan farklı değildir. Nesne oluşturulur ve ona davranışlar ve işlevler kazandırılır.

Modüler programlama için uygun olduğu için büyük hacimli yazılımlarda çok kullanılır. Bilinen ilk nesnel dil Smalltalk'tır. Sonraları C++, Java... gibi nesnel diller oluşturulmuştur.

VHDL yapısal dil özelliği gösterdiği gibi içinde nesnel özellikler barındırır. Bir VHDL nesnesi ancak aşağıdaki tiplerden birisiyle veya birkaçıyla oluşturulur.

- **Signal(SİNYAL)** : Bu component portları arasında karşılıklı bağlaşmayı sağlayan kablo gibi davranır.
- **Variable(DEĞİŞKEN)** : Değişken veriler için kullanılır. Genelde lokal olarak sadece Process'ler içinde kullanılır.
- **Constant(SABİT)** : Sabit değerler için kullanılır onun değeri sistem oluşumunun sonuna hiç değişmez.

5.4.1.1 İşaretler (Signal)

Sinyal nesneleri entity'leri birbirine ve aynı zamanda form modellerine bağlar. Sinyallerin görevi entity'leri birbirleriyle dinamik verilerle haberleştirmektir. Sinyal tanımlaması şöyle olur:

SIGNAL sinyalin_adi: sinyalin_tipi [:=başlangıç değeri];

SIGNAL tanımlamasının ardından sinyal ismi gelir. Sinyal isimleri ihtiyaç kadar belirtilir.

Belirttiğimiz isim sayısınca sinyal üretilir. Tanımlamada da belirtildiği gibi sinyal nesnelere ilk değer atanabilir. Son olarak sinyaller entity, architecture ve package(paket) yapılarının içinde tanımlanabilir. Paketler her yerde kullanılabilceği için burada yapılan tanımlamalar global özellik taşır. Çünkü bu tanımlamalar ancak paket kullanıldığı zaman aktif olurlar.

Sinyaller için bir takım kapsama(yerel, genel, ...) kuralları bulunmaktadır. Sinyaller; varlık(entity), mimari ve paket tanımlama anlarında tanımlanabilirler. Paket içi tanımlamalarda sinyaller genel bir tanımlama yapılmış gibi davranırlar. Çünkü paket

bilindiği üzere birçok mimari veya varlık tarafından kullanılabilir. Ayrıca değer atamaları da tanımlandığı yerde olmayabilir.

5.4.1.2 Değişkenler (Variable)

Değişkenler yerel olarak sadece Process'ler veya altprogramlar içinde geçici değerleri tutmak için kullanılırlar. Sinyal tanımlamanın tersine değişken tanımlamada, değişkenin tanımlandığı yerde değer atanamayabilirler. Değer atamaları daha sonraki satırlarda olabilir. Değişkenler ortaya çok hızlı çıkıp işini yapıp ortadan kaybolurlar yani hafızada yer kaplamazlar. Bir değişkenin tanımlanması genel hatlarıyla aşağıdaki gibi olur;

VARIABLE değişkenin_adi: değişkenin_tipi [:=başlangıç_değeri] ;

Burada her değişken adı birkaç tane değişken adı olabilir. Her isme ait bir değişken üretilir. variable_type değişkenin veri tipini tanımlar. Bu veri tipi kendi tanımladığımız bir veri tipi olabilir. Yukarıda belirtildiği gibi değişkenler ancak ve ancak altprogram ve process tanımlamalarında kullanılırlar. İlgili örneği aşağıda görebiliriz;

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
ENTITY and5 IS
    PORT ( a, b, c, d, e : IN std_logic;
           q : OUT std_logic);
END and5;

ARCHITECTURE and5 OF and5 IS
BEGIN
PROCESS(a, b, c, d, e)
    VARIABLE state : std_logic;
    VARIABLE delay : time;
BEGIN
    state := a AND b AND c AND d AND e;
    IF state = '1' THEN
        delay := 4.5 ns;
    ELSIF state = '0' THEN
        delay := 3 ns;
    ELSE
```

```
        delay := 4 ns;  
    END IF;  
    q <= state AFTER delay;  
    END PROCESS;  
    END and5;
```

Bu örnek 5-girişli bir AND kapısının mimarisini belirtmiştir. Burada yukarıda belirtildiği gibi sadece process içinde iki adet değişken bildirimi yapılmıştır. Bunlardan birisi durum (state), diğeri ise gecikme(delay) değişkenidir. Durum değişkeni girişlerin AND fonksiyonu için geçici verileri tutmak için kullanılır. Yine aynı biçimde, çıkış değerlerini sıralarken oluşan gecikme değerlerini de bu delay değişkeni tutar. Bu iki değişkenin de tuttuğu veriler statik(static) değildir. Çünkü bunların değerleri girişlere(a,b,c,d,e) bağlı olarak değişir.

Şurası unutulmamalıdır ki; verileri tutmak için değişkenler yerine sinyaller(signals) de kullanılabilir. Ama kullanılmadı. Bunun birçok nedeni vardır. Bunlardan başlıcaları;

- Doğal(içsel) olarak değişkenler veri tutma ve değiştirmede sinyallerden daha üstündür. Çünkü değişkenler ve yaptığı işlevler aniden oluşurlar. Fakat sinyallerin oluşması için bir plan oluşturulmaktadır.
- Değişkenler daha az yer kaplarlar. Sinyaller yapısı bakımından oluşturulması için o andaki durum la ilgili daha fazla veriye ihtiyaç duyarlar ve bu veriler elde edildiğinde çok yer kaplarlar.
- Sinyal kullanırken veri senkronizasyonunu sağlamak için WAIT deyiminin kullanılmasına ihtiyaç vardır.

Burada tanımlanan a,b,c,d veya e girişlerinden herhangi birisinin değer değişiminde programın içinde tanımlanmış process kısmı çalışmaya başlar. Durum değişkeni olan state tüm girişlerin değerini tutmak üzere görevlendirilir. Daha sonra state değişkeni tabanına dayanılarak gecikmeleri tutmak üzere de delay değişkeni görevlendirilir. Gecikmelere dayanılarak bu değerler delay değişkenine sevk edilir. Bu değerlere dayanılarak çıkış sinyali q, durum değişkeni olan state değişkeninin değerini alır ve çıkış üretilmiş olur.

5.4.1.3 Sabitler (Constant)

Sabitler bir tasarımda çok özel verileri tutmak için kullanılırlar. Sabitler tasarımcıya çok iyi dokümantasyona sahip bir model ortaya koyma olanağı sağlar. Eğer tasarımı yapılan modelde değişmez değerlere ihtiyaç varsa sabitler kullanılır. Tasarımcı isterse sabitlerin değerini elle değiştirip programı tekrar derleme ve optimum değeri bulma olanağına kavuşur.

Bir sabitin çok iyi dokümantasyon sağlamasından kasıt şudur; mesela tasarlanan modelde çok sayıda pi sayısı kullanılacak olsun. Her kullanışta pi sayısının direkt değeri olan 3.1414'ü kullanmaktansa bir adet sabit tanımlamalıdır. Bu tanımlama aşağıdaki gibi olur;

```
CONSTANT PI: REAL := 3.1414;
```

Burada bu tanımlamanın yapılması programın okunurluğunu artırmıştır. Programa bir tertip düzen katmıştır.

Genel sabit bildirimini aşağıdaki gibi olur;

```
CONSTANT sabitin_adi: veri_tipi [ := başlangıç_değeri ];
```

Sabitlerin tanımlanmasında esneklikler söz konusudur. VHDL değer ataması için ertelemeye olanak sunmaktadır. Yani sabitin değeri sabiti tanımlanan yerde atanmak zorunda değildir. Bir sabit paket tanımlanırken tanımlanabilir, değeri ise paket gövdesi tanımlanırken (koşullu atama olabilir) atanabilir.

Sabitlerin de sinyaller gibi bir takım kapsama kuralları vardır;

- Bir sabit birçok varlık (entity) tarafından kullanılan bir pakette tanımlanıyorsa, o sabit genel (global) bir sabit olmuş olur.
- Bir sabit eğer bir process içinde tanımlanmışsa sadece o process içinde kullanılabilir.

- Bir mimaride tanımlanan bir sabit processler dâhil hiçbir ifade tarafından kullanılmayabilir.

Şimdi veri tipleri adı altında diğer veri tiplerini açıklamaya başlayabiliriz:

5.4.2 Veri Tipleri (*Data Types*)

Bundan önceki bölümde sinyaller, değişkenler ve sabitler olmak üzere nesnel tipler açıklanmıştır. Bunlar en sade haldeki nesnelere dir. Eğer daha karmaşık nesnelere oluşturulmak istenirse diğer veri tiplerini bilmek gereklidir. VHDL çok geniş bir çerçevede birçok veri tipini destekler.

En başta; yeni bir veri tipi oluşturmak istenirse bir veri tipi tanım şablonuna gerek vardır. Bir veri tipi tanım şablonu, veri tipinin adı ve tipin tanım bölgesinden oluşur. Daha ileride verilen örneklerle konu daha iyi anlaşılacaktır. Yeni bir veri tipi paket tanımlama anında, varlık(entity) tanımlama anında, mimari tanımlama anında, altprogram ve process tanımlama anlarında oluşturulabilir. Yeni bir tip tanımlama şöyle olur;

TYPE tipin_adı **IS** tipin_markası;

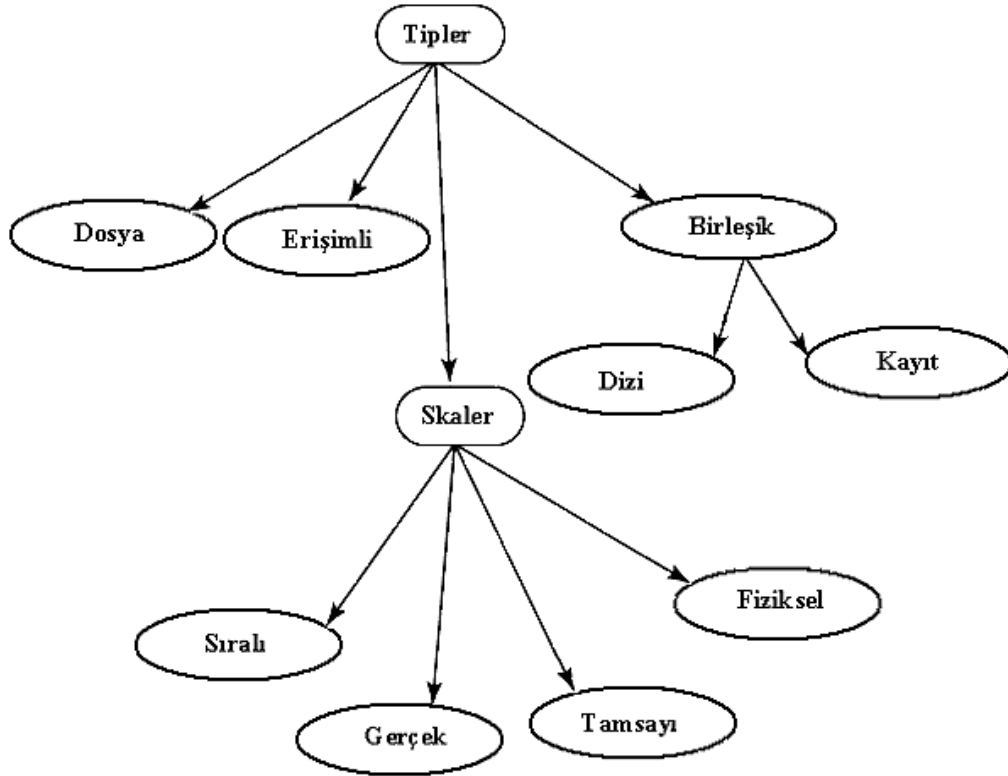
Burada tipin_markası biraz anlamsız gelebilir. Bu kısımda normal veri tiplerinden herhangi birisi veya bunlardan oluşmuş bir karmaşık veri tipleri olabilir. Bundan sonraki birkaç örnekte bunların nasıl kullanıldığı daha iyi anlaşılacaktır. Sinyaller ve sabitler için var olan kapsama kuralları buradaki kullanımlarında da geçerlidir.

Şekil 1.'de VHDL'de geçerli olan veri tipleri gösteren diyagram görülmektedir. Burada dört ana kategori göze çarpmaktadır. Bunlar; skaler tipler, birleşik tipler, erişimli ve dosyasal tiplerdir. Skaler tipler tamsayı ve gerçek sayı basit(sade) tiplerden oluşmaktadır. Birleşik tipler ise diziler ve kayıt tiplerini içerir. Erişimli tipler kasıt ise klasik programlama dillerindeki işaretçilere denktir. Son olarak dosyasal tipler ise tasarımcıya dosya nesnelere oluşturma olanağı sağlar.

5.4.2.1 Skaler Tipler (*Scalar Types*)

Skaler veri bir zaman diliminde sadece bir tane deęer tutabilen nesnelerdir. Bir nesne kendi kendine birden çok deęer ierebilir. Bir zaman diliminde bu deęerlerden sadece bir tanesini tutabilir. Skaler tipler Őu drt kısımdan oluŐur;

- Tamsayılar
- Gerek sayılar
- Sıralı sayılar
- Fiziksel sayılar



Őekil 5.3 VHDL Veri Tipleri Diyagramı

5.4.2.1.1 Tamsayılar (Integer)

Buradaki kullanılan tamsayılar tamamen matematikte kullanılan tamsayılara benzerler. Bu tamsayılarla matematikte yaptığımız toplama, çıkarma, çarpma ve bölme işlemlerini yapabiliriz. VHDL LRM tamsayıların maksimum aralığını değil de minimum aralığı olan (-2,147,483,647)'dan +2,147,483,674'ye kadar olan kısmını destekler. Bunlar Standart Library(*kütüphane*)'deki Standart pakette tanımlanmıştır.

Bu standart paket VHDL diline ihtiyaç olan bütün veri tiplerini tanımlanmıştır. Fakat bu paket normal bir varlık veya paket tarafından özel olarak çağrılmaz.

Aşağıda tamsayı ile ilgili birkaç örnek verilmiştir;

```
ARCHITECTURE test OF test IS  
BEGIN  
  PROCESS(X)  
    VARIABLE a : INTEGER;  
    VARIABLE b : int_type;  
  BEGIN  
    a := 1; --Ok 1  
    a := -1; --Ok 2  
    a := 1.0; --error 3  
  END PROCESS;  
END test;
```

İlk iki ifadede(1 ve 2) pozitif ve negatif tamsayıların atamaları gösterilmektedir. Üçüncü ifadede ise tamsayı olarak tanımlanmış bir değişkene gerçek sayı gibi atama yapılmıştır. Bu ifade derleyicide bir sorun olarak algılanacak ve hata mesajı bildirecektir. Çünkü VHDL veri tipi bakımından oldukça güçlü bir dildir.

5.4.2.1.2 Gerçek Sayılar (Real)

Gerçek sayılar matematikte bilinen gerçek sayılardan VHDL’de nesnelere oluşturmak için kullanılır. Ayrıca kesirli sayılar gibi tamsayıların gösteremediği sayıları göstermek için de kullanılırlar. Tamsayılar gibi gerçek sayıların minimum değer aralıkları Standart Kütüphanedeki Standart pakette tanımlanmıştır. Bu aralık -1.0E+38’den başlar ve +1.0E+38’de son bulur. Aşağıdaki örnekteki birkaç ifade kullanım biçimleri görülmektedir;

```
ARCHITECTURE test OF test IS
    SIGNAL a : REAL;
BEGIN
    a <= 1.0;           --Ok 1
    a <= 1;            --error 2
    a <= -1.0E10;     --Ok 3
    a <= 1.5E-20;    --Ok 4
    a <= 5.3 ns;     --error 5
END test;
```

Burada görüldüğü üzere gerçek sayı tipinden bir sinyal(signal) nesnesi tanımlanmıştır. Bütün gerçek sayılar yazılırken mutlaka bir ondalık kısmı bulunur. Bu ondalık kısmın değeri sıfır bile olsa belirtilir. Bu derleyicinin gerçek sayıları diğer tiplerden ayırt etmesini sağlar. Yukarıdaki örnekte birinci ifade bir nesneye bir değer atamasını göstermektedir. Anlatıldığı gibi ondalık kısmın değeri sıfır olmasına rağmen bildirim yapılmıştır. Nitekim ikinci ifadede sayısal değer aynı olmasına rağmen imla hatası yapılmıştır. Çünkü orada yapılan atama; tanımlanan sinyalin tipinin bir tamsayı olduğunu bildirmektedir. Beşinci ifadede ise imla hatası yoktur fakat tip uyumsuzluğu vardır. Çünkü tanımlanan nesnenin tipi zaman(time) değil gerçek sayıdır.

5.4.2.1.3 Sıralı Sayılar (Enumerated)

Sıralı sayılar soyut kavramları modellemek için çok güçlü tiplerdir. Bu tipten verilerin veri tiplerini tasarımcı oluşturabilir. Bu değerler karakter veya sayı olabilirler. Buna örnek bir veri x, abc, siyah gibi isimlerle tanımlanabilir. Bazen bu bir yılın ayları bazen haftanın günleri bazen rasgele isimler olabilirler. Mesela

karakteristik bir örnek vermek gerekirse dört durumlu bir simülasyon sistemi bu tip verilerle şöyle tanımlanabilir;

TYPE fourval **IS** ('X' , '0' , '1' , 'Z');

Görüldüğü üzere bu sistemde dört adet veri gösterilebilir. Bunların açıklaması şöyle olur;

- 'X'—Bilinmeyen değer
- '0'—Mantıksal '0'
- '1'—Mantıksal '1'
- 'Z'—Yüksek empedans durumu

Diğer bir örnek ise aşağıda olduğu gibi elemanları renk olan veri tipidir.

TYPE renk **IS** (kırmızı, sarı, mavi, yeşil, turuncu);

Bu örnekte veri tipleri tamamen soyuttur. Yani fiziksel bir değere sahip değildirler. Bu yüzden sıra dışıdır. Bizim projemize uygun bir örnek olarak bir komut seti bu çeşit bir tanımlama ile tanımlanabilir:

TYPE instruction **IS** (add, sub, lda, ldb, sta, stb, outa, xfr);

Bu çeşit veri tipleri genel olarak mantıksal olarak durum makinesi modellemelerinde kullanılabilirler. Durum makineleri genel olarak ASIC veya FPGA aygıtlarının kontrol mantığını oluşturmada kullanılmaktadırlar. Bu metot işlem sırasının doğru yürütülmesinde çok büyük kolaylık sağlar.

6. LVDS'E GİRİŞ

Düşük-Gerilim Fark İşaretleşme (LVDS) günümüz yüksek performanslı veri iletişim uygulamaları ihtiyaçlarına hitap eden yeni bir teknolojidir. LVDS standardı endüstride en popüler veri iletişim standardı olmaktadır. Bu, iki basit özelliikle desteklenmektedir: “Gigabits @ milliwatts!”

LVDS rakip teknolojilerden önemli derecede az güç tüketirken yüksek veri iletim hızı sağlamaktadır. Ek olarak, aşağıdaki faydaları da sağlamaktadır:

- Düşük-gerilim güç kaynağı uyumluluğu
- Düşük gürültü üretimi
- Yüksek gürültü yok etme
- Güçlü iletişim sinyalleri
- Sistem seviyesinde yongaların içinde bütünleşebilme kabiliyeti

LVDS teknolojisi ürünlerin 100'lerce Mbps'den 2 Gbps'in üzerine kadar geniş bir alanda yüksek veri hızlarına hitap etmelerini sağlamaktadır.

6.1 LVDS Modası

Tüketiciler ev ve ofiste daha fazla gerçekçi görsel bilgilere rağbet etmektedir. Bu, video, 3D grafiklerin ve gerçekçi resim verilerinin kameradan PC'ye, LAN üzerinden yazıcıya aktarılması, telefon ve uydu sistemleri, set-üstü kutular ve sayısal kayıt cihazlarının (VCR) evlere taşınması ihtiyaçlarına sebep olmaktadır. Günümüzde bu yüksek-hızlı sayısal verinin baskı devre kartları (PCB) ve fiber veya udu ağları üzerinden çok kısa ve çok uzun mesafelere taşınması için çözümler mevcuttur. Bu verilerin karttan-karta yâda kutudan-kutuya taşınması düşük gürültü üreten, minimum güç tüketen son derece yüksek performanslı bir çözümü gerektirmektedir.

LVDS bilgiyi taşımak için iki adet sinyal hattı kullanan bir fark sinyali standardıdır. Bu bir dezavantaj gibi görünse de aslında bir fayda sağlamaktadır. Sinyali taşımak

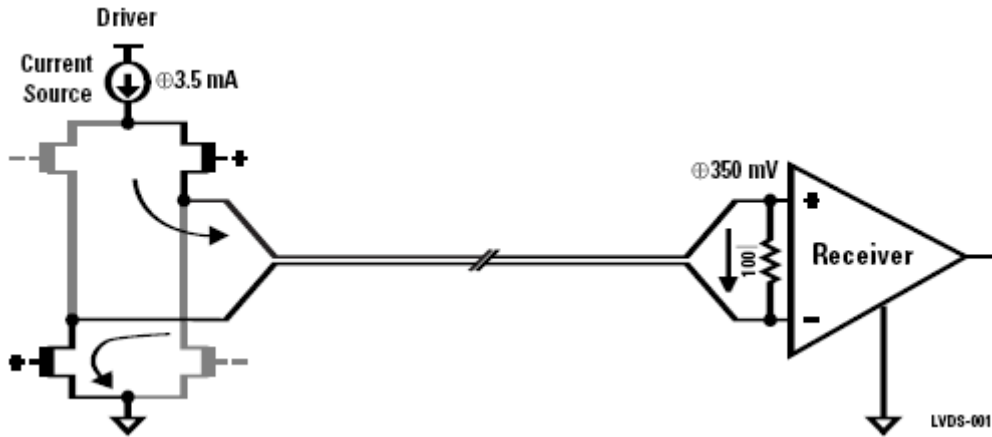
için iki yol kullanım maliyetine karşın ortak-mod bastırma formunda gürültü dayanıklılığını kazandırmaktadır.

Sinyal salınımı sadece bir kaç yüz milivolt seviyesine düşürülebilir çünkü sinyal-gürültü bastırma geliştirilmiştir. Küçük salınımlar yükselme zamanı çok kısa olduğundan yüksek veri iletim hızına olanak sağlamaktadır.

6.2 Düşük Gürültü ve Düşük Güçle Gelen Hız

LVDS saniyede yüzlerce hatta binlerce Megabit (Mbps) hızında tek kanal veri iletim hattına olanak sağlayan düşük salımlı bir fark işaretleme teknolojisidir. Düşük salınım ve akım-modu sürücü çıkışları düşük gürültü oluşturur ve geniş bir frekans aralığında çok düşük güç tüketimi sağlar.

6.2.1 LVDS Nasıl Çalışır?



Şekil 6.1 100Ω fark empedanslı ortam aracılığıyla bağlanmış LVDS sürücü ve alıcının basitleştirilmiş şeması

LVDS çıkışları fark çifti hatlarını süren bir akım kaynağından (nominal 3.5mA) oluşmaktadır. Temel alıcı yüksek DC giriş empedansına sahiptir, bu yüzden çıkış akımının çoğunluğu 100 Ω sonlandırma direncinden akarak alıcı girişinde yaklaşık 350mV oluşturur. Sürücü değiştiğinde geçerli '1' ve '0' lojik durumları oluşturarak dirençten akan akımın yönünü değiştirir.

6.2.2 LVDS Standardı

LVDS, güncel olarak TIA/EIA (Haberleşme Endüstrisi Kurumu/Elektronik Endüstrisi Kurumu) ANSI/TIA/EIA-644-A (LVDS) Standard tarafından standardize edilmiştir.

Genel (çoklu-uygulama) LVDS standardı, ANSI/TIA/EIA-644-A, 1995’de TIA Veri İletişim Arayüzü komisyonu TR30.2’de başlamıştır. 2001’de ANSI/TIA/EIA-644-A olarak gözden geçirilip düzeltilerek yayımlandı. ANSI/TIA/EIA standardı sürücü çıkışı ve alıcı giriş karakteristiklerini tanımlamaktadır, bu sonuçla sadece elektriksel bir standarttır. İşlevsel şartnameleri, protokolleri veya uygulamaya bağımlı olan kablo karakteristiklerini bile kapsamamaktadır. ANSI/TIA/EIA-644-A bütün arayüzleri (konektörler, protokol gibi) belirleyen diğer standartlara referans olması için tasarlanmıştır. Bu, birçok uygulamaya kolayca uyumluluğu sağlamaktadır.

Parameter	Description	Min.	Max.	Units
V_{OD}	Differential output voltage	247	454	mV
V_{OS}	Offset voltage	1.125	1.375	V
V_{OD}	Change to V_{OD}		50	mV
V_{OS}	Change to V_{OS}		50	mV
I_{SA}, I_{SB}	Short circuit current		24	mA
t_r/t_f	Output rise/fall times (200 Mbps)	0.26	1.5	ns
	Output rise/fall times (<200 Mbps)	0.26	30% of t_{ui} †	ns
I_{IN}	Input current		20	μ A
V_{TH}	Receive threshold voltage		+100	mV
V_{IN}	Input voltage range	0	2.4	V

Tablo 6.1. ANSI/TIA/EIA-644 (LVDS) standartları

6.3 LVDS Yongaları

Birçok LVDS Arayüz aygıtı, çeşitli sağlayıcılardan elde edilebilmektedir. LVDS çoğunlukla aşağıdaki yonga türlerinde bulunmaktadır.

Hat Sürücüleri/Alıcıları

Bu aygıtlar tipik olarak LVCMOS gibi single-ended sinyalleri bir backplane yâda kablo üzerinden iletim için daha uygun bir formata dönüştürmek için

kullanılmaktadır. Bunlar çoklu-kanal yapılandırmaları ile birlikte tek-kanal içerisinde de mevcuttur.

SerDes

Paralel-seri/Seri-paralel (Serializer/deserializer) dönüştürücü çifti birkaç düşük-hızlı CMOS hattını çoklamak ve bunları yüksek hızda çalışan tek bir kanal olarak göndermek için kullanılmaktadır. SerDes yongaları tipik olarak backplane ve kablo içindeki hatların yâda konektör pin sayısını azaltmak için kullanılmaktadır.

Anahtarlama Elemanları

Anahtarlama mimarileri, bus mimarileri üzerinde veri iletim hızları yüksek olduğunda desteklenmektedir. LVDS bu yongalar üzerindeki I/O'lar için ortak seçimdir. Anahtarlama Elemanları saat dağıtımı için kullanılabilir. LVDS, güvenilir sinyal bütünlüğü nedeniyle her hangi bir frekanstaki saat sinyalleri için en elverişli işaretleme standartlarından biridir.

6.4 Bus LVDS (BLVDS)

Bus LVDS (BLVDS olarak ta adlandırılır) özel olarak çoknoktalı kablo yâda backplane uygulamalarına hitap eden LVDS teknolojisi temelli yeni bir bus arayüzü devreler ailesidir. BLVDS, LVDS'den farklı olarak çoknoktalı uygulamalarda gerekli olan çift sonlandırmayı yöneten artırılmış sürücü akımı sağlar.

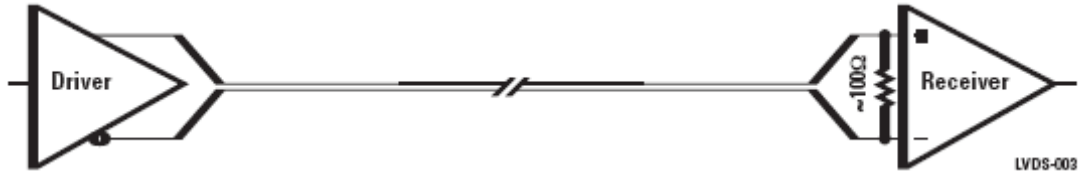
Bus LVDS, yüksek-hızlı bus tasarımlarında karşılaşılan birçok soruna hitap etmektedir.

- Bus LVDS özel bir yukarı-çekme hattı sonlandırma ihtiyacını ortadan kaldırmaktadır.
- Aktif sonlandırma aygıtlarına olan gereksinimi ortadan kaldırmaktadır
- Ortak güç kaynağı hattı (3.3V yâda 5V) kullanmaktadır
- Basit bir sonlandırma şeklini kullanır

- Arayüz aygıtlarındaki güç tüketimini azaltır
- Düşük gürültü üretir
- Kartların çalışırken eklenmesini destekler
- 100'lerce Mbps hızında ağır şekilde yüklenmiş çok-noktalı veri yollarını sürer

Bus LVDS ürünleri yüksek-hızlı, çok-noktalı bus arayüzü problemlerini çözmek için tasarımcılara yeni alternatifler sağlamaktadır. Bus LVDS yüksek performanslı backplane gerektiren kart yoğunluklu Telekom altyapı sistemleri ve datacom uygulamalarından uzun kablo mesafesi ve gürültü bağıışıklığının yararlı olduğu endüstri uygulamalarına kadar geniş bir uygulama alanına sahiptir.

6.5 LVDS Yapılandırmaları



Şekil 6.2. Noktadan-Noktaya Yapılandırma

LVDS sürücü ve alıcıları genelde şekil 6.2'de de gösterildiği gibi noktadan-noktaya yapılandırmada kullanılmaktadır. Tek amaçlı noktadan-noktaya bağlantılar sağladıkları belirgin yollardan dolayı en iyi sinyal kalitesini sağlamaktadır. Bu yapılandırmada, LVDS oldukça düşük güç kullanıyor ve çok az gürültü üretiyorken büyük kablo uzunluklarında yüksek hızlı sinyalleri gönderme yeteneğine sahiptir. Bununla birlikte, aynı zamanda başka topolojiler/yapılandırmalar da mümkündür.

Sistem tasarımcısı arabağlantı sayısını azaltmaya daha fazla ilgi gösterirken, LVDS göz önünde bulundurulması gereken büyük bir teknolojidir. LVDS çift-yönlü işaretleme ve bus uygulamaları için çok uygundur.

Şekli 6.3'de görülen yapılandırma tek bir bükümlü ikili üzerinden çift-yönlü iletişime olanak sağlamaktadır. Veriler aynı anda sadece tek yönde akmaktadır. İki adet sonlandırma direnci gereksinimi sinyali azaltır (ve böylece fark gürültü

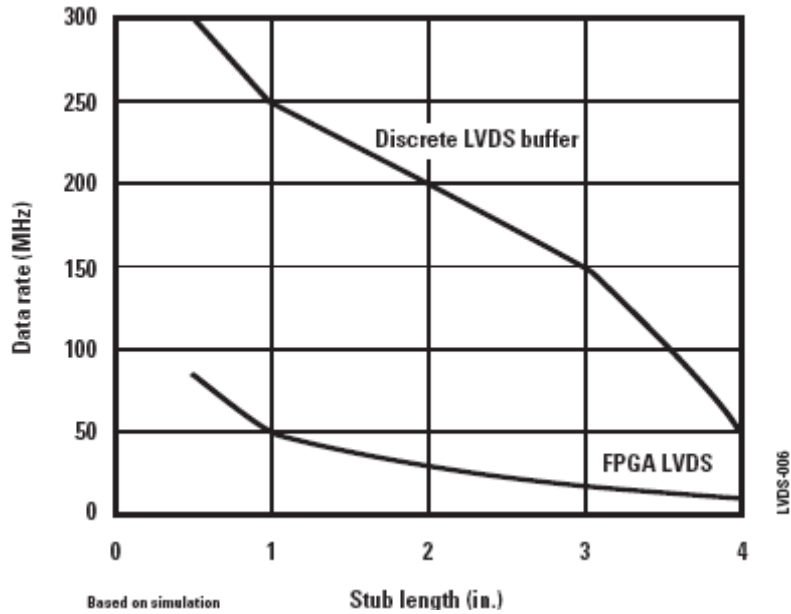
Sistem tasarımcıları sinyal bütünlüğünü sağlayacak uzlaşmalara karşı bu avantajları değerlendirmelidir. Tasarımcılar aynı zamanda ayrık çözümlerin tipik olarak tümleşik FPGA platformlarından daha fazla maliyetli olduğunun da farkında olmalıdırlar.

Birkaç büyük tasarım uğraşları, başarılı bir şekilde sağlıklı backplane arayüzü tasarımı gerçekleştirmek için gereklidir.

1. Stub uzunluğu: Bu, backplane izinden gelip backplane konektörü ile Bus LVDS alıcısına gelen izin uzunluğudur. Çoknoktalı ve çokbağlantılı veriyollarındaki uzun stublar zayıf sinyal güvenilirliğine yol açmaktadır.

2. ESD koruması: Elektrostatik boşalma direnci (ESD) yerleştirme ve sökme olaylarında sistem kartı güvenilirliğini arttırmaktadır.

3. Kapasitif yüklenme: Bu aktif bir aygıtın I/O'sunda görülen kapasitansdır. Yüksek kapasitans düşük empedans ve düşük gürültü toleransı sağlamaktadır.



Şekil 6.5 Backplane Performansının Tümleşik ve Ayrık Çözümler Arasında Karşılaştırılması

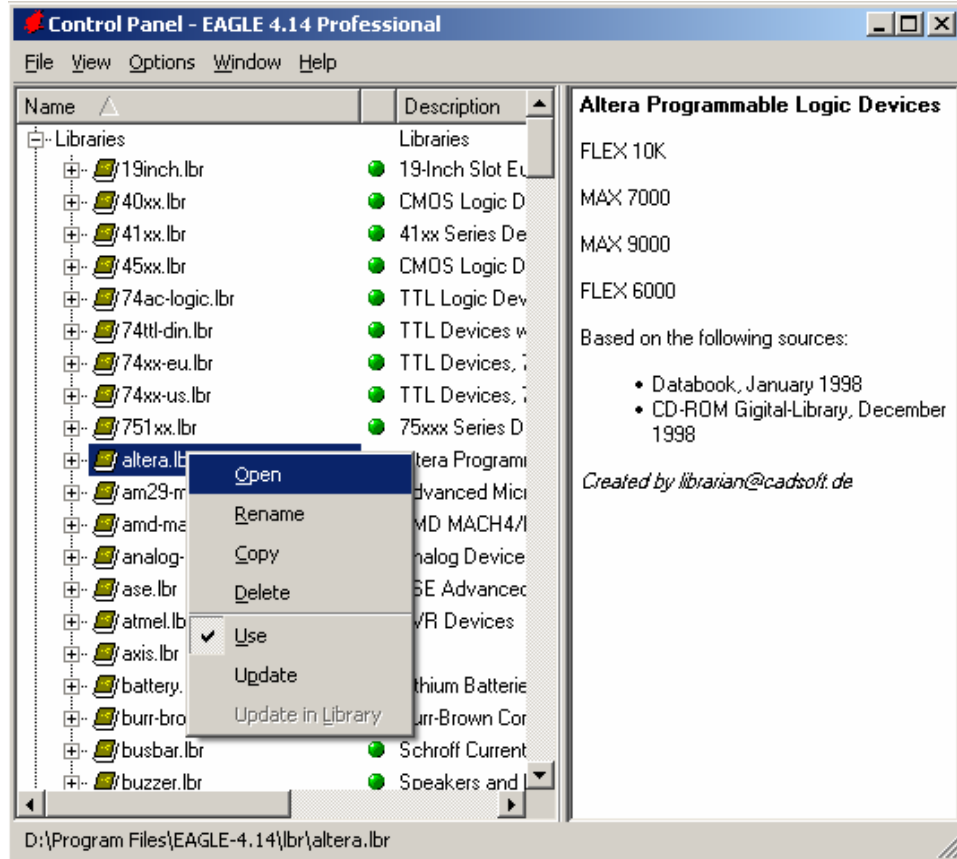
7. EAGLE KULLANARAK DEVRE TASARIMININ GERÇEKLENMESİ

Buraya kadar PCI mimarisi anlatılmaya çalışıldı ve tasarlanan devrede kullanılacak FPGA ve SERDES elemanları incelendi. Devre tasarım için kullanımı kolay olan EAGLE programını tercih edildi. Tercihin bir diğer sebebi de kütüphanesinde PCI Bus şablonunu barındırması ve kütüphanesine yeni elemanların kolay bir şekilde eklenebilmesidir.

Devre tasarımına geçmeden önce kullanılacak devre elemanlarını aşağıda anlatıldığı gibi tek tek EAGLE'ın kütüphanesine eklendi ve daha sonra bu elemanları kullanarak devrenin şematik tasarımını ve ardından PCB çizimi gerçekleştirildi.

7.1 EAGLE'da Kütüphane Oluşturma

Adım 1: İlk önce EAGLE kontrol paneli açılır ve menüden “File > New > Library” seçilir. İkinci bir seçenek olarak panelden mevcut bir kütüphane üzerine sağ tıklanıp “Open” seçilir:

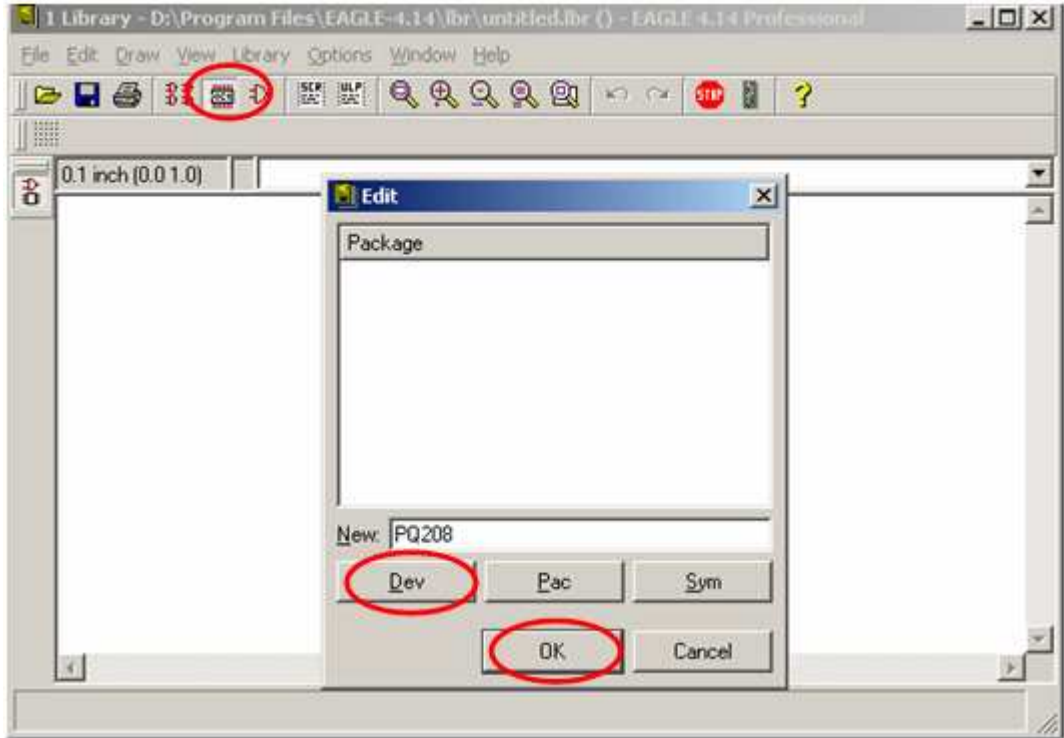


Adım 2: Ekranı boş bir sayfa açılacaktır. Burada oluşturacağımız eleman için aygıt (Device), kılıf (Package) ve sembol (Symbol) tanımlanmalıdır. Yeni bir aygıt oluşturmak için iki yol vardır. Birincisi kütüphane içerisinde bulunan başka bir elemanın paket şeklini yeni elemanımıza kopyalamak, diğeri ise en baştan yeni paket ve sembol oluşturmaktır.

Şimdi tasarımımızın en önemli parçası olan Xilinx Spartan-3E XC3S250E aygıtının kütüphanesini oluşturalım:

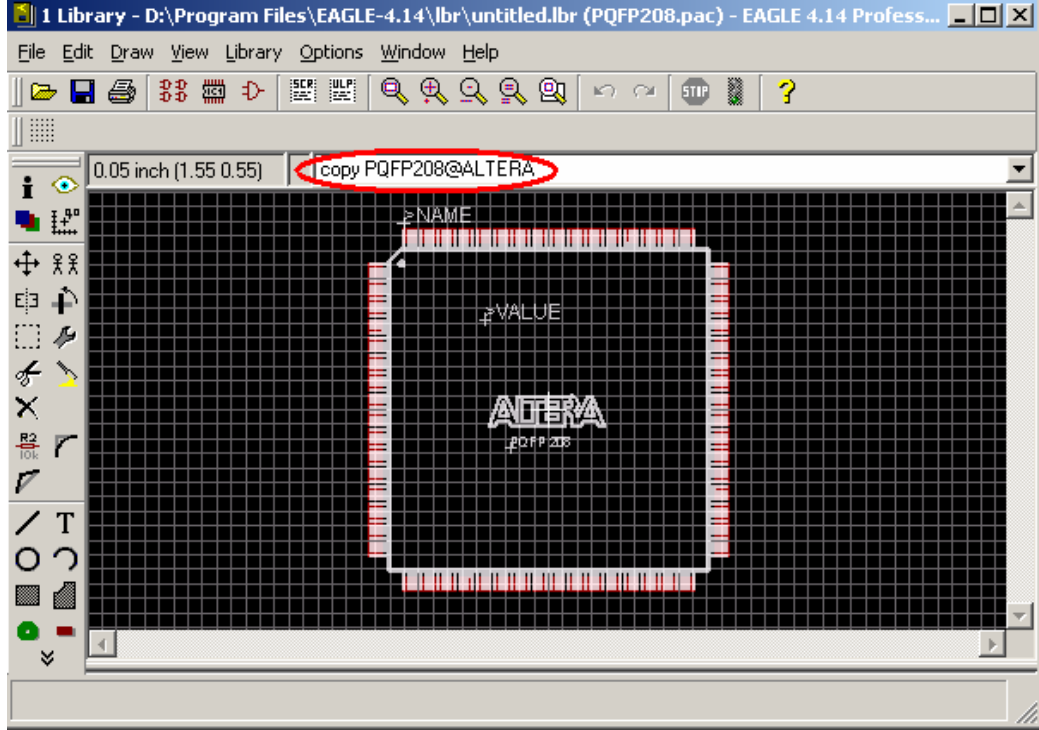
Adım 3: Kullanacak Spartan-3E FPGA aygıtı PQ208 kılıfında seçilmiştir. EAGLE'in kütüphanesinde PQ208 kılıfı, Altera kütüphanesinin içerisinde mevcuttur. Şimdi bunu kütüphanemize ekleyelim.

Önce menüden; Library > Package seçilir. Daha sonra açılan pencerede kılıf ismi "PQ208" olarak yazılır.



Daha sonra komut satırına "copy PQFP208@ALTERA" yazılarak kılıf kütüphanemize eklenmiş olur. "Edit > Delete" komutu kullanılarak "ALTERA" yazısı silinir ve yerine "Edit >Text" komutu seçilir kutucuğa "XILIX" yazılır ve

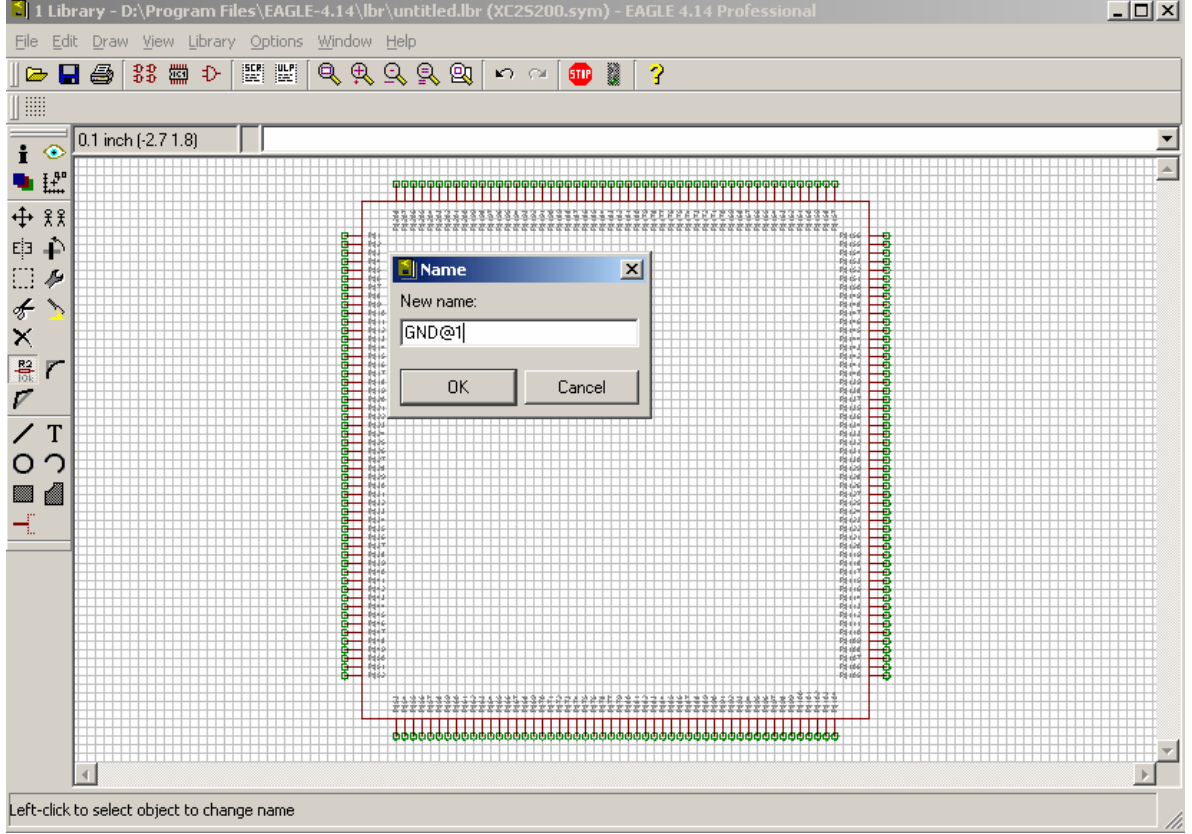
komut penceresinin hemen üzerindeki araç çubuğundan “tPlace” seçilir ve “size”, “angle” seçenekleri kullanılarak yazının boyutu ve yönü ayarlanarak istenen yere yerleştirilir. “File > Save” komutuyla kütüphanemize bir isim verilerek kaydedilir.



Kılıf var olan bir kütüphaneden kopyalandığı için pin numaraları da hazır olarak gelmiştir. Eğer pin numaralarını değiştirmek istenirse “File > Name” komutunu kullanarak pinlerin üzerine tıklayıp tek tek yeni pin numaraları verilebilir. Ayrıca hassas çizimler ve hareketler gerçekleştirebilmek için “View > Grid” komutu ile kullanılacak grid aralıkları mm yâda inç cinsinden girilebilir.

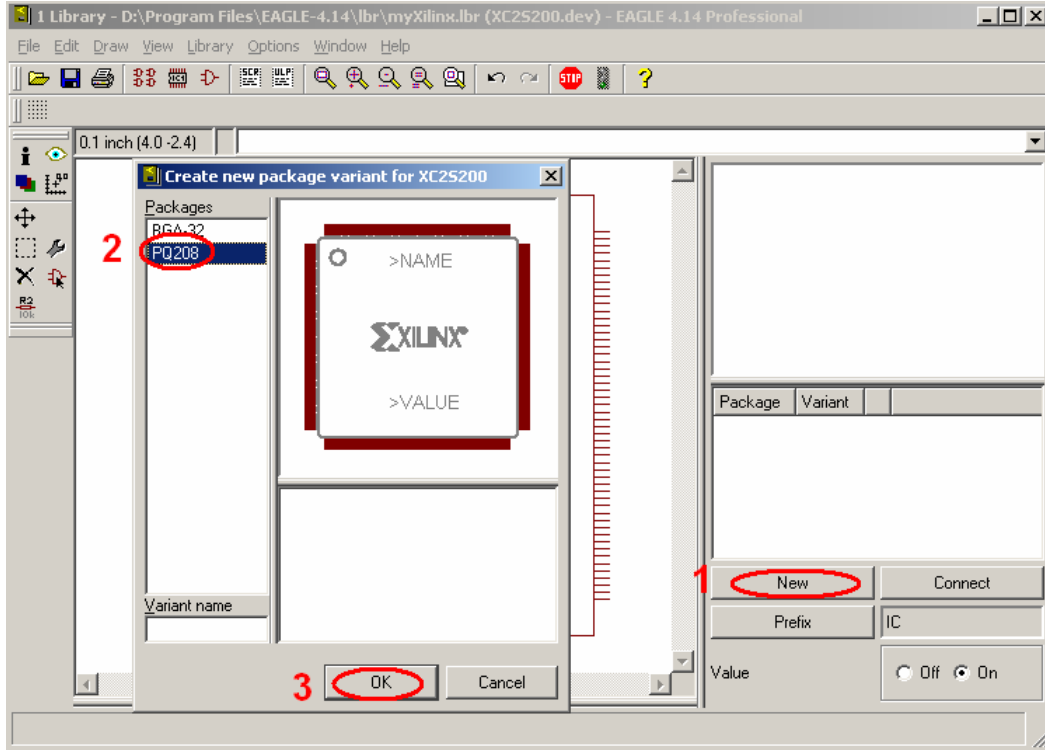
Adım 4: Kılıf oluşturulduktan sonra sıra sembolü oluşturmaya geldi. Bunun için “Library > Symbol” seçilir ve açılan pencerede “new” yazan kısma “XC3S250E” yazılır. Bu durumda ekrana boş bir sayfa gelecektir. Burada elemanımızın şematik tasarımıda görülecek şekli oluşturulacaktır. Önce elemanın pinleri tek tek sayfada uygun yere eklenir. Burada şeklin görüntüsü sembolik olduğundan aygıtlarla birebir örtüşme zorunluluğu yoktur. Kullanılmayan pinler burada tanımlanmayabilir. Ancak biz kullanım kolaylığı açısından FPGA’in PQ208 paketinin şeklini burada oluşturuyoruz. Önce “Draw > Pin” seçilerek pinler aşağıdaki şekilde görüldüğü gibi yerleştirilir ve “Edit > Name” komutuyla her bir pin tek tek tıklanarak pin adları katalog isimlerine uygun olarak yazılır. Burada aynı isimli pinler sonlarına ‘@’

işareti ve sıra numarası verilerek yazılır. Bu şekilde PCB tasarımında bu pinler otomatik olarak birbirine bağlanacaktır. Daha sonra “Draw > Wire” çalıştırılır ve komut satırının üstündeki satırdan “96 Names” seçilerek pinlerin uçlarından FPGA’ın şekli çizilir:

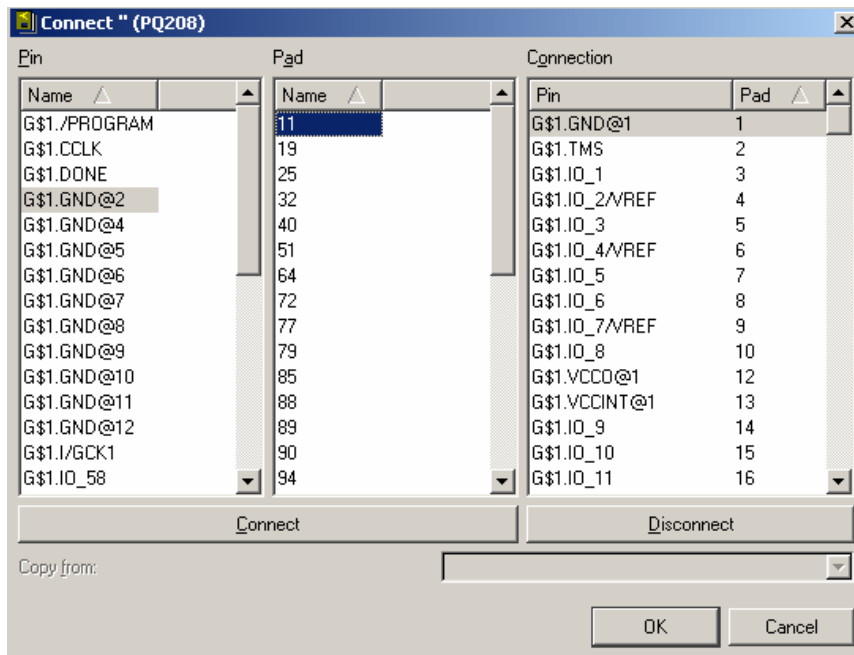


Adım 5: Sembolde oluşturulduktan sonra sıra aygıtın oluşturulmasına geldi. “Library > Device” seçilerek açılan pencerede “Spartan-3EPQ208” yazılarak yeni bir sayfa açılır. Burada daha önce oluşturduğumuz kılıf ve sembolün pin bağlantılarını eşleştireceğiz. Böylece şematik olarak çizdiğimiz devreyi kolayca PCB’ye aktarabiliriz.

Önce “Edit > Add” komutu kullanılarak bir önceki adımda oluşturduğumuz sembol açılan listeden seçilerek sayfaya eklenir. Daha sonra ekranın sağ alt kısmında bulunan kılıf menüsünden “New” tıklanarak açılan pencereden “PQ208” seçilir ve pencere kapatılır.



Sembol ve kılıf eklendikten sonra sıra pin bağlantılarını eşleştirmeye geldi. Şimdi yine sağ alt kısımdaki menüden “Connect” seçilir. Bu durumda aşağıdaki pencere karşımıza çıkar. Burada sembol adı ve pin numaraları ilgili kısımlarda seçilerek “Connect” tuşuna basılır. Sağ kısımdan eşleşen sembol ve pin adları kontrol edilebilir.



Tüm eşleşmeler yapıldıktan sonra “Ok” tuşu ile pencere kapatılır ve kütüphane kaydedilir. Böylece Xilinx Spartan-3E PQ208 aygıtı kütüphaneye eklenmiş oldu. Bu işlemler EAGLE’in kütüphanesinde bulunmayan DS92LV18, DS90LV001, DS90LV004, XCF02S EPROM, regülâtör entegreleri, konektörler ve diğer eleman yukarıda anlatılan adımlar için de uygulanarak kütüphaneye eklenmiştir.

7.2 Devre Şeması Tasarımı

Buraya kadar devrenin tasarımı için gerekli bilgi ve altyapı oluşturuldu. Bu noktadan sonra devre tasarımına geçilebilir. İlk aşama FPGA ile PCI veri yolu arasında bağlanacak sinyallerin ve bu sinyallere karşılık gelecek FPGA pinlerinin belirlenmesidir. Bölüm 2.2’de şekil 2.1’de görülen gerekli sinyaller ve opsiyonel INTA# sinyali ile beraber toplam 50 adet PCI sinyali, kullanacağımız sinyaller olacaktır. PCI ile FPGA arasındaki pin bağlantıları bölüm 2.3.4’de önerilen şekilde yerleştirilecektir. Spartan-3E, zaten hâlihazırda özel olarak adlandırılmış IRDY ve TRDY girişlerine sahiptir. Bu pinler PCI karşılıkları ile eşleşecek şekilde diğer pinlerde önerilen düzende FPGA I/O’larına bağlanmalıdır. Burada çok önemli bir problem ortaya çıkmaktadır: Tasarlanan kart PCI 33 MHz 5V modundadır ancak Spartan-3E, 5V sinyalleşmeyi desteklememektedir. Bu problemi çözmek için PCI veri yolu pinleri ile FPGA arasına 5V’u 3.3V’a çeviren 10-bitlik SN74CBT3861 “Bus Switch” yongalarından 5 adet kullanılmıştır.

FPGA-PCI bağlantısı belirlendikten sonra sıra SERDES ile FPGA’in bağlantısına geldi. Tasarımda kullanılan SERDES yongası verileri 18-Bitlik gruplar halinde gönderip almaktadır. Bu 18-bitin 2 biti karşı taraftaki sistemler için kontrol bitleri olarak düşünülmüştür ancak istenirse bu iki bit kullanılmayabilir yâda veriler 18 bitlik gruplara ayrılarakta gönderilebilir, bu kullanılacak haberleşme protokolüne bağlıdır. 6. bölümde de bahsedildiği gibi FPGA aygıtımız kendi LVDS çıkışlarına da sahiptir. Ancak Şekil 6.5’den görüleceği gibi FPGA’in LVDS çıkışları ayrı LVDS sitemlerden daha yavaştır. FPGA’in LVDS özelliğini test etmek amacıyla bu özelliğe sahip 2 çıkış ve 1 giriş pin çifti kullanıldı. LVDS sinyallerini RJ-45 soketine kadar az kayıpla taşıyabilmek için tasarıma ayrıca DS90LV004 LVDS Buffer

yongası eklendi. Ayrıca SERDES'in LVDS giriş ve çıkış hatları için de iki adet DS90LV001 LVDS buffer yongası kullanıldı.

SERDES bağlantısından sonra FPGA'in yapılandırma dosyasını barındıran PROM bağlantısı belirlenecektir. Seçilen Spartan-3E XC3S250E FPGA'i yaklaşık 1.3MBit yapılandırma verisine ihtiyaç duymaktadır. Bunun için 2MBit kapasiteli seri XCF02S PROM'u seçildi. Bu EPROM Master-Serial Modda (Bkz. şekil 3.10) FPGA'a bağlanacaktır. Bu durumda harici bir saat sinyaline gerek kalmamaktadır. Ayrıca FPGA yapılandırma seçme pinleri M2,M1 ve M0, Master-Serial Moda ayarlanmıştır. PROM'a yapılandırma verisi çeşitli yollarla yüklenebilmektedir. Bu tasarımda PROM'a yapılandırma verisini yüklemek için JTAG portu sisteme eklenmiştir. Ayrıca tasarıma eklenen "Done" ledi sayesinde FPGA'in doğru olarak yüklenip yüklenmediğini gözlemlenebilecektir. FPGA içerisine yüklenecek PCI IP Core, PCI saat sinyalini kullanacaktır. SERDES ise farklı bir saat sinyaline ihtiyaç duyabilir. Devreye gerekirse daha sonra takılabilecek bir kristal için soket eklenerek bu problemin de üstesinden gelinmiştir.

Son aşama, elemanları besleyecek gerilimlerin üretilmesi ve bypass kapasitelerinin belirlenmesidir. FPGA, V_{CCINT} Core gerilimi 1.2V, V_{CCAUX} için 2.5V ve I/O'lar için V_{CCO} 3.3V olmak üzere üç farklı gerilime ihtiyaç duymaktadır. PCI konektöründe 3.3V mevcuttur. Bu yüzden V_{CCO} pinleri buradan beslenebilir. FPGA Core ve V_{CCAUX} gerilimleri ise gerilim regülâtörü vasıtasıyla elde edilecektir. Regülasyon devreleri de belirlendikten sonra son olarak bypass ve dekuplaj kapasiteleri belirlenecektir. SERDES için bypass kapasite değerleri kendi datasheet'inde belirtilmiştir. FPGA için bypass kapasite değerleri bölüm 3.6.2.1 ve bölüm 3.6.2.2'de kullanılan formüllerden yararlanılarak dekuplaj kapasiteleri belirlenmiştir. Kullanılmayan bazı I/O'lar çeşitli amaçlarla kullanılmak için bir konektörde toplanmıştır.

Kartın şematik tasarımı bittikten sonra devrenin baskı devre çizimi için yine eagle programı kullanıldı. Çizim için kartın üretileceği tesisin üretim için kullandığı parametre değerleri alındı ve bu parametrelere uygun olarak baskı devre çizimi gerçekleştirildi.

8. SONUÇ

Bu projede tasarlanmaya çalışılan FPGA tabanlı PCI kartın donanımsal kısmının tasarımı gerçekleştirildi. Bu noktaya gelebilmek için çeşitli aşamalardan geçildi. Her şeyden önce PCI veriyolu mimarisini tanımaya çalışarak işe başlandı. Bunun için PCI v2.2 şartnamesini tasarım için gerekli donanımsal bilgileri elde edecek kadar inceledi. PCI veriyolunun çalışması kavrandıktan sonra bu veriyolu ile haberleşmeyi sağlayacak IP Core yazılımını belirlemeye sıra geldi. Opencores adlı internet sitesinde PCI ile haberleşmemizi sağlayacak program kodları ücretsiz olarak yayınlanmaktadır. Ayrıca bir PCI Core yazılımı ile uğraşmak oldukça vakit alacağından güvenilirliği testlerle kanıtlanmış bir yazılımı kullanmak hem zaman hem de maliyet açısından oldukça kolaylık sağlayacaktır. Tasarlanan kartta bulunacak FPGA istendiği zaman yeniden programlanabildiğinden eğer şartlar gerektirirse başka bir PCI IP Core'da kullanılabilir. Dolayısıyla bu aşamada kullanılacak olan IP Core'un içeriği tasarımı çok fazla etkilememektedir. Son olarak devrenin PCB şeması EAGLE programı kullanarak çizildi ve devrenin fiziksel tasarımı tamamlanmış oldu.

Bu aşamadan sonra elemanlar temin edilerek kartın üretilmesi ile tasarımın donanım kısmı tamamlanabilir. Daha sonra PCI IP Core ile SERDES arasında veri aktarımını sağlayacak program yazılarak kartın testleri yapıp kart çalışır duruma getirildikten sonra aygıt sürücüsü ve işletim sistemi üzerinde çalışacak program yazılabilir.

Yaptığım bu çalışmalar sonucunda PCI veri yolu, FPGA ve LVDS standardı hakkında oldukça bilgi sahibi oldum ve kendimi PCB tasarımı konusunda da geliştirdiğime inanıyorum. Umarım bu çalışmam bu ve buna benzer konularda çalışan yâda çalışacak olan kişilere de faydalı olur.

KAYNAKLAR

1. Peripheral Component Interface Special Interest Group (PCI-SIG),
<http://www.pcisig.com>
2. PCI Local Bus Specification Rev. 2.2,
www.pcisig.com/specifications/conventional/
3. PCI v2.2 Compatible PCI IP CORE
<http://www.opencores.org/projects.cgi/web/pci/home>
4. Douglas L. Parry, "VHDL Programming By Example", Fourth Edition,
McGraw-Hill (2002)
5. Peter C. Ashenden, 1990; "The VHDL Cookbook", First Edition
6. National Semiconductor Corp., 2002; "LVDS Owner's Manual", Third Edition
7. Xilinx Spartan-II 2.5V FPGA Family Complete Data Sheet
www.xilinx.com/bvdocs/publications/ds001.pdf
8. Spartan-3 Starter Kit Board User Guide
www.xilinx.com/bvdocs/userguides/ug130.pdf
9. Spartan-3 Complete Datasheet
www.xilinx.com/bvdocs/publications/ds099.pdf
10. Spartan-3E Starter Kit Board User Guide
www.xilinx.com/bvdocs/userguides/ug230.pdf
11. Spartan-3E Complete Datasheet
www.xilinx.com/bvdocs/publications/ds312.pdf
12. Platform Flash In-System Programmable Configuration PROMS
www.xilinx.com/bvdocs/publications/ds123.pdf
13. JTAG Programmer Guide
www.xilinx.com/support/sw_manuals/2_1i/download/jtag.pdf
14. Cable Hardware Userguide
www.xilinx.com/support/sw_manuals/2_1i/download/huguide.pdf
15. Xilinx Parallel Cable IV
www.xilinx.com/bvdocs/publications/ds097.pdf
16. FPGA Configuration Guidelines
www.xilinx.com/bvdocs/appnotes/xapp090.pdf

17. The Low-Cost, Efficient Serial Configuration of Spartan FPGAs
www.xilinx.com/bvdocs/appnotes/xapp098.pdf
18. Configuration Quick Start Guidelines
www.xilinx.com/bvdocs/appnotes/xapp501.pdf
19. Data Generation and Configuration for Spartan Series FPGAs
www.xilinx.com/bvdocs/appnotes/xapp126.pdf
20. The 3.3V Configuration of Spartan-3 FPGAs
www.xilinx.com/bvdocs/appnotes/xapp453.pdf
21. Powering Xilinx FPGAs
www.xilinx.com/bvdocs/appnotes/xapp158.pdf
22. Powering Xilinx Spartan-II FPGAs
www.xilinx.com/bvdocs/appnotes/xapp189.pdf
23. Power-On Requirements for the Spartan-II and Spartan-IIE Families
www.xilinx.com/bvdocs/appnotes/xapp450.pdf
24. Power-Assist Circuits for the Spartan-II and Spartan-IIE Families
www.xilinx.com/bvdocs/appnotes/xapp451.pdf
25. Power Distribution System (PDS) Design: Using Bypass/Decoupling Capacitors
www.xilinx.com/bvdocs/appnotes/xapp623.pdf
26. Power Management Application Guide for Xilinx® FPGAs
www.intersil.com/data/ag/ag0001.pdf
27. Power Supply Technical Application Note for Xilinx FPGAs
www.bellnix.com/fpga/ANXilinx-E.pdf
28. Power Management Solutions for Xilinx Programmable Logic Devices
www.linear.com/designtools/Product_Guide_Xilinx_RevC1.pdf
29. Using Digital Clock Managers (DCMs) in Spartan-3 FPGAs
www.xilinx.com/bvdocs/appnotes/xapp462.pdf
30. 5 Volt Tolerant and PCI
www.xilinx.com/bvdocs/appnotes/xapp311.pdf
31. Connecting Virtex-II Devices to a 3.3V/5V PCI Bus
www.xilinx.com/bvdocs/appnotes/xapp646.pdf
32. 3.3V PCI Design Guidelines
www.xilinx.com/bvdocs/appnotes/xapp653.pdf

33. LogiCore PCI v3.0 User Guide
www.xilinx.com/products/logicore/pci/docs/pci_ug159.pdf
34. PCI32 v3 and v4 Interface Product Specification
www.xilinx.com/partinfo/pci/pci_ds206.pdf
35. LogiCore PCI v3.0 Getting Started Guide
www.xilinx.com/partinfo/pci/pci_gsg157.pdf
36. PQFP (PQ208/PQG208) Package
www.xilinx.com/bvdocs/packages/pq208.pdf
37. Device Package User Guide
www.xilinx.com/bvdocs/userguides/ug112.pdf
38. National LM1086CS-2.5 1.5A Low Dropout Positive Regulator
www.national.com/pf/LM/LM1086.html
39. National LP3852 1.5A Fast Ultra Low Dropout Linear Regulator
www.national.com/pf/LP/LP3852.html
40. Programmable High-Frequency Crystal Oscillator SG-8002JF Series
<http://www.epson.co.jp/CRYSTAL/>
41. Texas Instruments SN74CBT3861 10-BIT FET BUS SWITCH
<http://focus.ti.com/docs/prod/folders/print/sn74cbt3861.html>
42. National DS92LV18 18-Bit Bus LVDS Serializer/Deserializer
<http://www.national.com/pf/DS/DS92LV18.html>
43. National DS90LV001 - 3.3V LVDS-LVDS Buffer
<http://www.national.com/pf/DS/DS90LV001.html>
44. National DS90LV004 - Four-Channel LVDS Repeaters with Pre-Emphasis
<http://www.national.com/pf/DS/DS90LV004.html>
45. National Semiconductor Corp., 2005; "18-bit SerDes Design Guide"

ÖZGEÇMİŞ

1981 yılında İzmit'te doğdu. İlk ve orta öğrenimini İzmit'te tamamladı. 1995 yılında Kocaeli Endüstri Meslek Lisesi Elektrik Bölümü'nü kazandı, 1996'da Kocaeli Teknik Lisesi Elektronik Bölümü'ne geçti ve 1998 yılında lise eğitimini bitirdi. 2000–2002 yılları arasında Kocaeli Üniversitesi Kocaeli Meslek Yüksek Okulu Elektronik Haberleşme Bölümü'nde ön lisans eğitimi aldı. 2003 yılında Kocaeli Üniversitesi Kocaeli Mühendislik Fakültesi Elektronik ve Haberleşme Mühendisliği Bölümü'ne dikey geçiş yaptı. Elektronik ve Haberleşme Mühendisliği'nde son sınıf öğrencisi olarak mezun olma aşamasındadır.

EK-1 BASKI DEVRE ŞEMASI:

