

1. Giriş

Günlük yaşantımızda farkında olalım ya da olmayalım elektronik cihazlar fazlasıyla yer kaplamaktadır. Teknolojik gelişimin en fazla hissedildiği alanların başında elektronik sektörü yer almaktadır. Bu gelişim aynı zamanda ciddi bir rekabet ortamı oluşturmakta ve teknolojiyi geliştiren kişi ya da kurumları daha hızlı ürün geliştirme-prototipleme konusunda zorlamaktadır. Bu rekabet ortamında daha hızlı ürün geliştirmek, ihtiyaçlara özel çözümler üretmek adına pek çok araç geliştirilmiştir.

Geliştirilen bu araçların en önemlilerinden biri de FPGA'lar ve onlar ile tasarım yapmamızı sağlayan tasarım dilleridir. FPGA'ların her şeyden önce bahsedilmesi gereken -belki de en önemli- özellikleri sayısal (dijital) olarak yapılabilen tüm tasarımların tek bir entegre üzerinde gerçekleştirilebilmesine olanak sağlamalarıdır. Aynı FPGA ile çok basit bir lojik devre tasarımından, karmaşık CPU-MCU tasarımına kadar çok geniş bir yelpazede çalışma yapma imkânı sunulmaktadır.

FPGA ile donanım tasarımı yapabilmek amacıyla geliştirilmiş birçok dil mevcuttur. Bu dillerden yaygın olarak kullanılan diller VHDL ve Verilog'dur. Bu kitapta VHDL dili tanıtılacaktır.

Daha öncede bahsedildiği üzere FPGA kullanılarak çok geniş bir yelpazede tasarım yapılabilmesi FPGA'ların gerek endüstride gerekse eğitim kurumlarında mantık devreleri ile alakalı eğitim ve tasarım derslerinde kullanılabilmesini olanaklı kılmaktadır.

1.1. FPGA Nedir?

FPGA (Field Programmable Gate Array) basitçe ifade etmek gerekirse içerisinde programlanabilir pek çok mantık devresi barındıran bir tümleşik devredir. Bu basit ifadeye karşılık günümüzde kullanılan FPGA'lar çok daha gelişmiş özellikler sunmaktadır. FPGA içinde bulunan mantık devreleri yapılmak istenen tasarıma göre birbirine bağlanarak düzenlenebilmekte ve bu sayede çok basit tasarımlardan çok karmaşık tasarımlara kadar pek çok donanım gerçekleştirilmesini mümkün kılmaktadır. Bu durumu kabaca şu şekilde özetleyebiliriz: FPGA içinde bulunan mantık devrelerini tuğla olarak kabul edersek; bu tuğlaları kullanarak basit bir duvar da inşa edebiliriz kocaman bir saray da.

FPGA'ların sunduğu bir diğer güzel özellik ise tekrar ve tekrar programlanabilir olmasıdır. Bu sayede bir tasarım geliştirirken son derece hızlı bir şekilde tasarladığımız mimarinin kontrolünü yapabilmekte, hatalı olması durumunda ise kolayca düzeltebilmekteyiz. Yeni bir tasarım-prototip geliştirirken bu özellik büyük kolaylıklar sağlamaktadır.

1.2. Neden FPGA?

Her şeyden önce FPGA'lar yüksek oranda paralel veri işleme olanağı sağlamaktadır. Geleneksel bilgisayar mimarisinden farklı olarak (yapılan tasarıma göre değişmekle birlikte) birbiri ile neden-sonuç ilişkisi bulunmayan birimler eş-zamanlı (paralel bir şekilde) çalışabilmektedir. Özellikle görüntü işleme, sinyal işleme gibi yoğun bilgi girişi olan ve elde edilen verinin kısa sürede işlenmesi gibi durumlar için neredeyse rakipsiz bir seçenektir.

Pek çok birimin eş-zamanlı çalışabilmesi (tasarıma bağlı olarak) tüm sistemin, geleneksel bilgisayar mimarisine göre daha düşük frekanslarda çalışabilmesine müsaade etmektedir. Bu sayede örneğin 100-200 MHz civarındaki bir çalışma frekansı ile birkaç GHz mertebesindeki işlemciler ile başa baş işlem gücü alınmaktadır.

FPGA'ların bir diğer avantajı ise düşük güç tüketmesidir. Özellikle robotik, uzay ve havacılık gibi enerji kaynaklarının kısıtlı olduğu, neredeyse gerçek zamanlı işlenmesi gereken pek çok bilginin olduğu ortamlarda eşsiz bir seçenek olmaktadır.

FPGA'lar ayrıca pek çok dijital entegrenin yapacağı görevleri tek başına yapabilecekleri şekilde görevlendirilebildikleri için devre kartlarının daha küçük bir yapıda tasarlanabilmelerine olanak sağlamaktadır. Bu özellikle mobil robotik gibi fiziksel alan kısıtlaması olan tasarımlarda çok önemli bir fayda olmaktadır.

1.3. VHDL Hakkında

VHDL, günümüzde donanım tanımlama dili olarak ön plana çıkmıştır. VHDL yazılım dili kullanılarak, her türlü lojik devre tasarlanabilir. VHDL dili kullanılarak tasarım yapılmasının standart donanım tasarım yöntemine göre bazı önemli üstünlükleri vardır.

- **Tasarım süresi:** Teknolojinin hızla gelişimi beraberinde gerçekleşen bir devrenin kullanım ömrü azalmaktadır. Buda devrenin tasarım zamanının kısıtlanması anlamına gelir. Böyle durumlarda, devrenin en uygun şekilde tasarlanmış olmasının ötesinde tasarım süresinin kısalığı ön plana çıkmaktadır. VHDL dili kullanılarak tasarım yapılması özellikle bu noktada tasarımcıya önemli yararlar sağlar. VHDL dili doğrudan donanım tasarımına göre, tasarım süresi açısından çok daha kısa sürede sonuçlanır.
- **Tasarım esnekliği:** Teknolojideki değişimle birlikte kullanılan elamanların yapıları değişmektedir. Yapı değişikliklerinin daha önce yapılmış tasarımlarda çalışabilmesi için, kullanılan tasarım ortamının buna uygun olabilmesi gerekir. VHDL dili fonksiyon bağımlı olarak çalışır. Dönüştürücü programlar yardımıyla yazılımın donanım yapısı oluşturulur. Teknoloji değişimleri durumlarında sadece bu dönüştürücü programların yeni teknolojiye uygun hale getirilmiş olması yeterli olacaktır.
- **Tasarım kolaylığı:** Genel olarak VHDL dili kullanarak yapılan tasarımlarda, klasik donanım tasarımına göre, donanım bilgisine daha az ihtiyaç duyulur.
- **Yenileme kolaylığı:** Teknolojideki hızlı değişim, gerçekleşen devrelerin değişim sürelerini azaltmıştır. Bu nedenle yapılacak değişimlerin hızlı bir şekilde gerçekleştirilebilmesi gerekmektedir. VHDL dili ile yapılan tasarımlarda değişim esnekliği tasarımcının yazılım gücü ile sınırlıdır.

1.4. Kitap Hakkında

Bu kitap başta öğrenciler olmak üzere konu ile ilgilenen herkesin isteklerini kapsayacak şekilde geniş bir içerikle hazırlanmaya çalışılmıştır. Gerek öğrenciler gerekse eğitimciler için bir başvuru ve ders kitabı olarak kullanılacak şekilde örneklerle desteklenmiş anlatımlar içermektedir. Ayrıca basitten zora doğru giden bir anlatımla pek çok uygulama da kitabın içerisinde yer almaktadır.

Kitapta anlatılan konuların kavranması için temel mantık devreleri (lojik devreler) ile ilgili konuların bilinmesi gerekmekte olup uygulamaların denenebilmesi için bilgisayar gerekmektedir. NEXSY4 deney kartı uygulamaların gerçekte çalıştığını görmek adına tavsiye edilmekle birlikte yazılım ortamı tarafından sağlanan benzetim ortamı da uygulamaları denemek ve çıkışları gözlemek adına kullanılabilir.

Kitabın genel içeriği şu şekilde düzenlenmiştir:

1. Kullanılacak deney kartının ve yazılımın tanıtımı, örnek proje oluşturulması. (Bölüm 2)
2. VHDL Dili anlatımı ve örnek kodlar (Bölüm 3 – Bölüm 8)
3. Örnek uygulamalar (Bölüm 9- Bölüm 10)

Kitapta sunulan örnek uygulamalar benzetim ve Nexys 4 uygulamaları olmak üzere 2 bölüme ayrılmıştır.

Benzetim uygulamaları:

1. Yetki Girişli D-Mandalı
2. Yükselen Kenar Tetiklemeli D İki Durumlusu (D Flip-Flop)
3. Asenkron Resetli D İki Durumlusu (Asenkron Resetli D Flip-Flop)
4. Senkron Resetli D İki Durumlusu (Senkron Resetli D Flip-Flop)
5. Saklayıcı (Register)
6. Kaydırmalı Saklayıcı (Shift Register)
7. Sayaçlar
8. Saat (Clock) Frekans Bölme
9. VHDL’de metin dosyasından veri okuma
10. VHDL’de ROM Bloğu Oluşturmak
11. VHDL’de RAM Bloğu Oluşturmak
12. FIFO Tasarımı
13. Sinyal İşlemede Konvolüsyon
14. Temel İmge İşleme Algoritmaları
15. VHDL ile İmge’de Konvolüsyon İşlemi

Nexys 4 uygulamaları:

1. D İki Durumlusu ve tasarımın FPGA'ya yüklenmesi
2. Led Yakma Uygulaması ve Flash Dosyası Oluşturma
3. Vivado ile Hata Ayıklama (Debug) Uygulaması
4. Display Üzerinde Karakter Kaydırma Uygulaması
5. UART Protokolü Kullanarak Data Kontrolü