



T.C
KOCAELİ SAĞLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ
YAZILIM MÜHENDİSLİĞİ

PROJE KONUSU:
SAYISAL TASARIM PROJESİ

YAVUZHAN KURŞUN
230502047
İBRAHİM KEÇİCİ
230502049

DERS SORUMLUSU:
PROF. DR. NEVCİHAN DURU

31.05.2024

1 GİRİŞ

1.1 Projenin amacı

Bu projenin amacı, dijital mantık devrelerini simüle edebilecek bir uygulama geliştirmektir. Uygulama, kullanıcıların çeşitli mantık kapılarını ve bileşenlerini kullanarak kendi mantık devrelerini oluşturmalarına olanak tanıyan bir arayüz sağlar. Bu simülatör, mantık kapılarının davranışlarını anlamak ve dijital tasarım becerilerini geliştirmek isteyen öğrenciler ve mühendisler için eğitici bir araç olarak kullanılabilir. Uygulamanın temel amaçları şu şekilde özetlenebilir:

Eğitim ve Öğrenme Aracı Sağlama:

Dijital mantık devrelerinin temel prensiplerini öğrenmek isteyen öğrenciler için interaktif bir eğitim aracı sunmak.

Tasarlama ve Simülasyon:

Kullanıcıların kendi dijital mantık devrelerini tasarlayabilmesi ve bu devrelerin nasıl çalıştığını simüle edebilmesi.

Görsel Geri Bildirim Sağlama:

Mantık kapıları ve bileşenlerin davranışlarını görsel olarak gözlemleme imkanı sunarak, kullanıcıların mantık devrelerinin işleyişini daha iyi anlamasını sağlamak.

Kolay Kullanım:

Kullanıcı dostu bir arayüz ile bileşenlerin kolayca eklenip yerleştirilmesini ve bağlantıların oluşturulmasını sağlamak.

Çeşitli Mantık Kapıları Desteği:

Farklı türde mantık kapıları (NOT, AND, OR, NAND, NOR, XOR, XNOR) ve temel bileşenler (giriş ve çıkış kutuları, LED'ler) kullanarak çeşitli devreler oluşturma imkanı sunmak.

Simülasyon Sonuçlarını Gözlemleme:

Kullanıcıların devrelerini çalıştırıp, girişlerin nasıl çıktılara dönüştüğünü gözlemlemelerini sağlamak.

Bu proje, kullanıcıların dijital mantık devrelerinin temel işlevlerini anlamalarına yardımcı olmayı hedeflerken, aynı zamanda dijital tasarım projelerinde kullanılabilecek pratik bir araç sunar. Öğrencilerin ve mühendislerin mantık kapıları arasındaki ilişkileri ve dijital devrelerin işleyişini kavramalarına katkıda bulunur.

2 GEREKSİNİM ANALİZİ

2.1 Arayüz gereksinimleri

Ana Ekran ve Tuval:

- Tuval Alanı: Mantık kapıları ve bileşenlerin yerleştirileceği geniş bir alan.
- Arka Plan: Bileşenlerin net görünmesi için açık renkli arka plan.

Kontrol Paneli:

Bileşen ekleme butonları:

- Mantık kapıları (NOT, BUFFER, AND, OR, NAND, NOR, XOR, XNOR)
- Giriş kutuları (0 ve 1)
- Çıkış kutusu, LED

Simülasyon Kontrolleri:

- Çalıştır Butonu: Simülasyonu başlatır.
- Reset Butonu: Tüm bileşenleri ve tuvali temizler.
- Durdur Butonu: Devam eden simülasyonu durdurur.

Durum Geri Bildirimi:

- Bilgilendirme Mesajları: Simülasyon durumlarını bildirir.
- Bileşen Durumu: LED'lerin yanıp yanmadığını ve çıkış kutularının değerlerini gösterir.

Kullanıcı Etkileşimi:

- Bileşenlerin Yerleştirilmesi: Sürükle-bırak yöntemiyle tuvale yerleştirme.
- Bileşenlerin Düzenlenmesi: Konum değiştirme ve silme.
- Giriş Değerlerinin Ayarlanması: Giriş kutularının değerlerini değiştirme.

Estetik ve Kullanılabilirlik:

- Renk Kodlaması: Farklı bileşenler için farklı renkler.
- Açıklayıcı Etiketler: Butonlar ve bileşenlerin işlevlerini belirten etiketler.

2.2 Fonksiyonel gereksinimler

Bileşen Ekleme ve Düzenleme:

- Mantık Kapıları Ekleme: Kullanıcı, tuvale NOT, BUFFER, AND, OR, NAND, NOR, XOR ve XNOR mantık kapılarını ekleyebilmelidir.

- Giriş ve Çıkış Kutuları Ekleme: Kullanıcı, giriş kutuları (değerleri 0 veya 1) ve çıkış kutuları ekleyebilmelidir.
- LED Ekleme: Kullanıcı, LED bileşenleri ekleyebilmelidir.
- Bileşenlerin Yerini Değiştirme: Kullanıcı, tuvale eklenen bileşenlerin konumlarını değiştirebilmelidir.
- Bileşenleri Silme: Kullanıcı, tuvalden bileşenleri silebilmelidir.

Simülasyon Kontrolleri:

- Simülasyonu Başlatma: Kullanıcı, devredeki bileşenlerin giriş değerlerine göre çıktılarının hesaplandığı simülasyonu başlatabilmelidir.
- Simülasyonu Durdurma: Kullanıcı, devam eden simülasyonu durdurabilmelidir.
- Simülasyonu Sıfırlama: Kullanıcı, tuvaldeki tüm bileşenleri ve bağlantıları temizleyebilmelidir.

Bileşenler Arası Bağlantılar:

- Giriş ve Çıkış Bağlantıları: Mantık kapıları, giriş ve çıkış kutuları ile LED'ler arasında bağlantılar oluşturulabilmelidir. Her mantık kapısının giriş ve çıkışları uygun şekilde bağlanabilmelidir.
- Dinamik Değer Güncelleme: Giriş kutularının değerleri dinamik olarak değiştirildiğinde, bağlı mantık kapılarının ve çıkışların çıktıları otomatik olarak güncellenmelidir.

Mantık Kapıları İşlevselliği:

- NOT Kapısı: Tek giriş değeri terslenmelidir.
- BUFFER Kapısı: Tek giriş değeri doğrudan çıkışa aktarılmalıdır.
- AND Kapısı: İki giriş değeri VE işlemi ile hesaplanmalıdır.
- OR Kapısı: İki giriş değeri VEYA işlemi ile hesaplanmalıdır.
- NAND Kapısı: İki giriş değeri VE işlemi yapıldıktan sonra sonuç terslenmelidir.
- NOR Kapısı: İki giriş değeri VEYA işlemi yapıldıktan sonra sonuç terslenmelidir.
- XOR Kapısı: İki giriş değeri farklı ise sonuç 1, aynı ise sonuç 0 olmalıdır.
- XNOR Kapısı: İki giriş değeri aynı ise sonuç 1, farklı ise sonuç 0 olmalıdır.

Görsel ve İşlevsel Geri Bildirim:

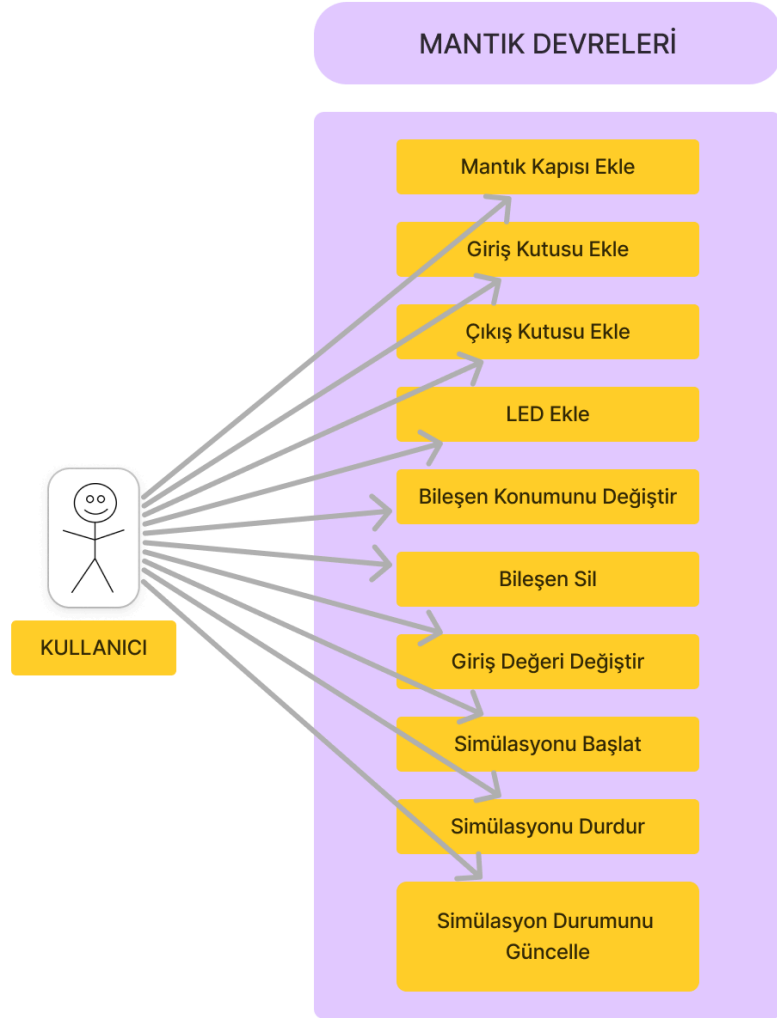
- LED Durumu: LED'ler, bağlı oldukları kapılardan gelen çıktıya göre yanmalı veya sönmelidir.
- Çıkış Kutusu Değeri: Çıkış kutuları, bağlı oldukları kapılardan gelen çıktıları kendi içlerinde göstermelidir.

- Bilgilendirme Mesajları: Simülasyonun başlangıcı, durdurulması ve tamamlanması gibi durumlarda kullanıcıya bilgilendirme mesajları gösterilmelidir.

Kullanıcı Etkileşimi ve Kullanılabilirlik:

- Kullanıcı Dostu Arayüz: Bileşenlerin kolayca eklenip, yerleştirilip, düzenlenebileceği kullanıcı dostu bir arayüz sağlanmalıdır.
- Kolay Kullanım: Simülasyon kontrolleri ve bileşen ekleme işlemleri kolay ve sezgisel olmalıdır.

2.3 Use-Case diyagramı



3 TASARIM

3.1 Mimari tasarım

Kullanıcı Arayüzü Katmanı:

Kullanıcıların sistemle etkileşimde bulunduğu katmandır. Bu katman, kullanıcıların veri girişi yapabileceği formlar, raporlama ekranları ve diğer görsel öğeler içerir.

İş Mantığı Katmanı:

Uygulamanın iş kurallarını, veri işleme mantığını ve iş süreçlerini içerir. Kullanıcı arayüzünden gelen istekleri işler, verileri işler ve gerekli işlemleri gerçekleştirir.

Nesne Tabanlı Programlama:

Projede nesne tabanlı programlama prensipleri kullanılarak her bir varlık için bir sınıf oluşturulmuştur. Bu sınıfların birbirleriyle etkileşimi sağlanmıştır.

3.2 Kullanılacak teknolojiler

Projede, Python programlama dili ve Tkinter kütüphanesi kullanılarak bir masaüstü uygulaması geliştirilmiştir. Python, geniş bir kütüphane yelpazesi ve kolay okunabilir bir sözdizimi ile kullanımı yaygın olan bir programlama dilidir. Tkinter, Python'un standart kütüphanelerinden biridir ve GUI (Grafiksel Kullanıcı Arayüzü) oluşturmak için kullanılır.

Python: Ana programlama dili olarak kullanılmıştır. Kullanımı kolaydır ve geniş bir topluluk tarafından desteklenmektedir.

Tkinter: Python'un standart kütüphanelerinden biri olan Tkinter, GUI uygulamaları oluşturmak için kullanılmıştır. Basit ve hızlı bir şekilde kullanıcı arayüzleri oluşturmak için idealdir.

Bu teknolojilerin kullanılması, uygulamanın hızlı bir şekilde geliştirilmesini sağlamış ve kullanıcı dostu bir arayüzün kolayca tasarlanmasını mümkün kılmıştır.

3.3 Kullanıcı arayüzü tasarımı

- Kullanıcı arayüzü tasarımı ile ilgili açıklama.
- Yazılımdan ekran çıktıları alınarak üzerinden açıklama yapılması
- Uygulamanın nasıl çalıştırılacağı ile ilgili açıklama

4 UYGULAMA

4.1 Kodlanan bileşenlerin açıklamaları

MantıkKapi Sınıfı:

Mantık kapılarını temsil eden bir sınıftır. NOT, BUFFER, AND, OR, NAND, NOR, XOR, XNOR kapıları için temel sınıftır.

- `__init__()`: Sınıfın başlatıcı metodu. Kapı türüne göre ilgili kapıyı oluşturur.

- `draw_gate()`: Mantık kapısını çizmek için kullanılan bir metottur.
- `calculate_output()`: Kapının çıkışını hesaplamak için kullanılan bir metottur.

GirisKutusu Sınıfı:

Giriş değerlerini temsil eden bir sınıftır.

- `__init__()`: Sınıfın başlatıcı metodu. Giriş kutusunun koordinatlarını ve başlangıç değerini ayarlar.
- `set_value()`: Giriş değerini ayarlamak için kullanılan bir metottur.

CikisKutusu Sınıfı:

Çıkış değerlerini temsil eden bir sınıftır.

- `__init__()`: Sınıfın başlatıcı metodu. Çıkış kutusunun koordinatlarını ayarlar.
- `set_value()`: Çıkış değerini ayarlamak için kullanılan bir metottur.

Led Sınıfı:

LED'leri temsil eden bir sınıftır.

- `__init__()`: Sınıfın başlatıcı metodu. LED'in koordinatlarını ve durumunu ayarlar.
- `set_state()`: LED'in durumunu ayarlamak için kullanılan bir metottur.

4.2 Görev dağılımı

Rapor ve ödevi beraber ortak hazırladık. Kimi zaman buluşup yaptık kimi zaman internet üzerinden konuşarak.

Kaynakça

- [Python Tkinter - GeeksforGeeks](#)
- [callable\(\) in Python - GeeksforGeeks](#)
- [Learn Tkinter from Scratch to Create Desktop Applications](#)
- <https://www.analyticsvidhya.com/blog/2024/02/logic-gates-in-python/>

Github

- [Yavuzhan-Github](#)
- [İbrahim-Github](#)