



T.C
KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR/YAZILIM MÜHENDİSLİęİ

PROJE KONUSU:
SAYISAL TASARIM PROJESİ

ÖęRENCİ ADI:
DAMLA KEKLİK
ECENAZ BOZDOęAN
ÖęRENCİ NUMARASI:
230501033
220502040

DERS SORUMLUSU:
DR. ÖęR. ÜYESİ NUR BANU ALBAYRAK

TARİH:02.06.2024

1 GİRİŞ

1.1 Projenin amacı

Bu proje, sayısal devrelerin ve mantık kapılarının çalışma prensiplerini simüle eden bir uygulamadır. Kullanıcılar, çeşitli mantık kapılarını (AND, OR, NOT, NAND, NOR, XOR, XNOR, Buffer) ve giriş/çıkış elemanlarını (Giriş, Çıkış, LED) tuval üzerine yerleştirerek, bağlantılar oluşturabilir ve bu devrelerin nasıl çalıştığını gözlemleyebilirler.

Projede Gerçekleştirilmesi Beklenenler:

1. Mantık Kapıları ve Giriş/Çıkış Elemanlarının Tuval Üzerine Yerleştirilmesi:

- Kullanıcı, menüden seçtiği mantık kapılarını ve giriş/çıkış elemanlarını tuval üzerine yerleştirebilir.
- Yerleştirilen elemanlar fare ile sürüklenip taşınabilir.

2. Mantık Kapıları ve Giriş/Çıkış Elemanları Arasında Bağlantı

Kurulması:

- Kullanıcı, elemanlar arasında bağlantılar oluşturarak bir devre tasarlayabilir.
- Bağlantılar dinamik olarak güncellenir ve doğru renk ile gösterilir (aktif olduğunda yeşil, pasif olduğunda mavi).

3. Mantık Kapılarının ve Devrelerin Doğru Şekilde Değerlendirilmesi:

- Her bir mantık kapısı, bağlı olduğu girişlerden gelen değerlere göre doğru sonuçları üretir.
- Kapılar arasında veri akışı sağlanarak tüm devre değerlendirilir.

4. Giriş ve Çıkış Elemanlarının Etkileşimli Kullanımı:

- Giriş kapıları, kullanıcı tarafından tıklanarak 0 veya 1 (pasif veya aktif) duruma getirilebilir.
- Çıkış elemanları (LED) ve diğer kapılar, bağlı oldukları mantık kapılarının çıktısına göre renk değiştirerek durumlarını gösterir.

5. Simülasyonun Başlatılması ve Durdurulması:

- Kullanıcı, simülasyonu başlatabilir ve devrenin çalışmasını gerçek zamanlı olarak izleyebilir.
- Simülasyon durdurulup tekrar başlatılabilir veya sıfırlanabilir.

6. Topolojik Sıralama ve Doğru Değerlendirme Sırası:

- Mantık kapılarının doğru sırayla değerlendirilmesi için topolojik sıralama yapılır.

7. Kullanıcı Dostu Arayüz ve Hata Yönetimi:

- Kullanıcıya bilgilendirici mesajlar ve hata uyarıları gösterilir.
- Kullanıcı hatalarından kaynaklanan sorunlar, programın düzgün çalışmasını engellemeden çözülür.

8. Devre Elemanlarının Durumlarının Canlı Güncellenmesi:

- Simülasyon sırasında devre elemanlarının durumu ve bağlantılar dinamik olarak güncellenir.
- Kullanıcı, devrenin anlık durumunu kolayca izleyebilir.

2 GEREKSİNİM ANALİZİ

2.1 Arayüz gereksinimleri

Kullanıcı Arayüzü Gereksinimleri

1. Başlatma ve Durdurma Kontrolleri:

- Simülasyonu başlatmak, durdurmak ve sıfırlamak için kontrol düğmeleri.

2. Mantık Kapısı Ekleme:

- Kullanıcıların AND, OR, NOT, NAND, NOR, XOR, XNOR ve Buffer gibi mantık kapılarını ekleyebilmeleri.

3. Giriş/Çıkış Elemanları:

- Giriş ve çıkış kapıları (input/output gates) ile LED lambaları ekleyebilme.

4. Bağlantı Oluşturma:

- Mantık kapıları ve diğer elemanlar arasında bağlantı oluşturma işlevi.

5. Simülasyon Ekranı:

- Kapıların, bağlantıların ve LED lambalarının görsel olarak yerleştirildiği ve hareket ettirilebildiği bir tuval.

6. Etiketleme:

- Kapılar ve elemanlar için etiketler (AND, OR, vb.) ile görsel ayırım.

7. Giriş Durumu Değiştirme:

- Giriş kapılarının durumlarını değiştirmek için düğmeler.

8. Bağlantı Görselleştirme:

- Bağlantıların renkleri ile aktif veya pasif olduklarını görselleştirme (örneğin, yeşil aktif, mavi pasif).

Donanım Arayüzü Gereksinimleri

1. Fare ve Klavye:

- Fare ile kapıları ve bağlantıları sürükleyip bırakma, tıklama işlevleri.

2. Ekran:

- Yeterli çözünürlükte bir ekran, simülasyonun görselleştirilmesi için.

3. İşlemci ve Bellek:

- Simülasyon işlemlerinin sorunsuz çalışması için yeterli işlemci gücü ve bellek kapasitesi.

2.2 Fonksiyonel gereksinimler

1. Mantık Kapısı Ekleme:

Kullanıcıların AND, OR, NOT, NAND, NOR, XOR, XNOR ve Buffer gibi mantık kapılarını simülasyona ekleyebilmesi.

2. Giriş ve Çıkış Elemanları:

Kullanıcıların giriş (input) ve çıkış (output) kapıları ekleyebilmesi. LED lambaları gibi çıkış elemanlarını ekleyebilmesi.

3. Bağlantı Oluşturma ve Düzenleme:

Mantık kapıları ve diğer elemanlar arasında bağlantı oluşturabilme. Bağlantıları sürükleyerek düzenleyebilme veya silebilme.

4. Simülasyonu Başlatma ve Durdurma:

Kullanıcıların simülasyonu başlatabilmesi ve durdurabilmesi. Simülasyonu sıfırlayarak yeniden başlatabilme.

5. Giriş Değerlerinin Ayarlanması:

Kullanıcıların giriş kapılarının durumlarını (0 veya 1) değiştirebilmesi.

6. Durum Görselleştirme:

Kapıların ve bağlantıların durumlarının görsel olarak gösterilmesi. Bağlantıların aktif (örneğin, yeşil) veya pasif (örneğin, mavi) olarak renklerle belirtilmesi.

7. Etiketleme ve İsimlendirme:

Mantık kapılarının ve diğer elemanların isimlendirilmesi ve etiketlenmesi.

8. Hata ve Uyarı Mesajları:

Geçersiz bağlantılar veya devreler için hata ve uyarı mesajlarının gösterilmesi.

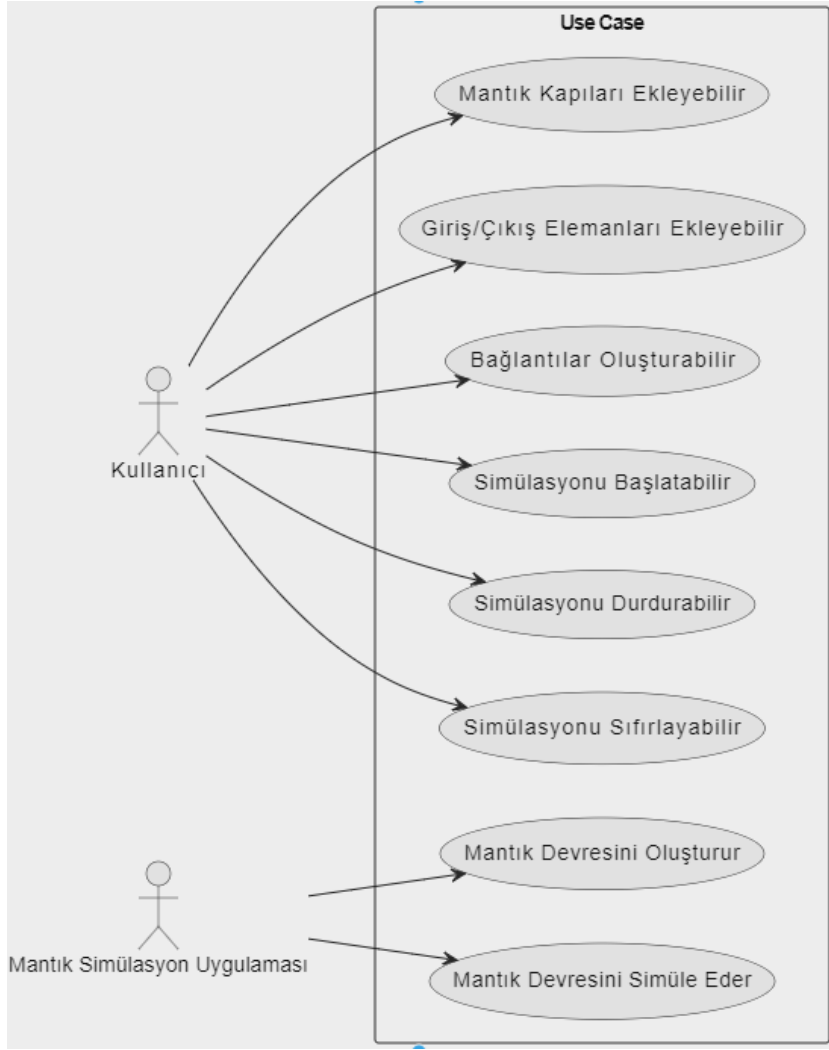
9. Kullanıcı Dostu Arayüz:

Basit ve anlaşılır bir kullanıcı arayüzü ile kullanıcıların kolayca simülasyonu yönetebilmesi.

10. Gerçek Zamanlı Güncelleme:

Giriş değerlerinin değişmesiyle birlikte devre durumu ve çıkışların anında güncellenmesi.

2.3 Use-Case diyagramı



3 TASARIM

3.1 Mimari tasarım

1. Kullanıcı Arayüzü Katmanı:

Ön Yüz: Kullanıcının simülatör ile etkileşimde bulunduğu katman. Mantık kapılarını eklemek, bağlantıları oluşturmak ve giriş/çıkış değerlerini değiştirmek gibi işlevleri sağlar.

Grafik Kütüphaneleri: Mantık kapılarının ve bağlantıların görsel olarak temsil edilmesi için kullanılan kütüphaneler

2. Uygulama Katmanı:

Mantık Kapısı Modülleri: AND, OR, NOT, NAND, NOR, XOR, XNOR ve Buffer gibi farklı mantık kapısı türleri için ayrı modüller. Her bir modül, belirli bir mantık kapısının davranışını simüle eder.

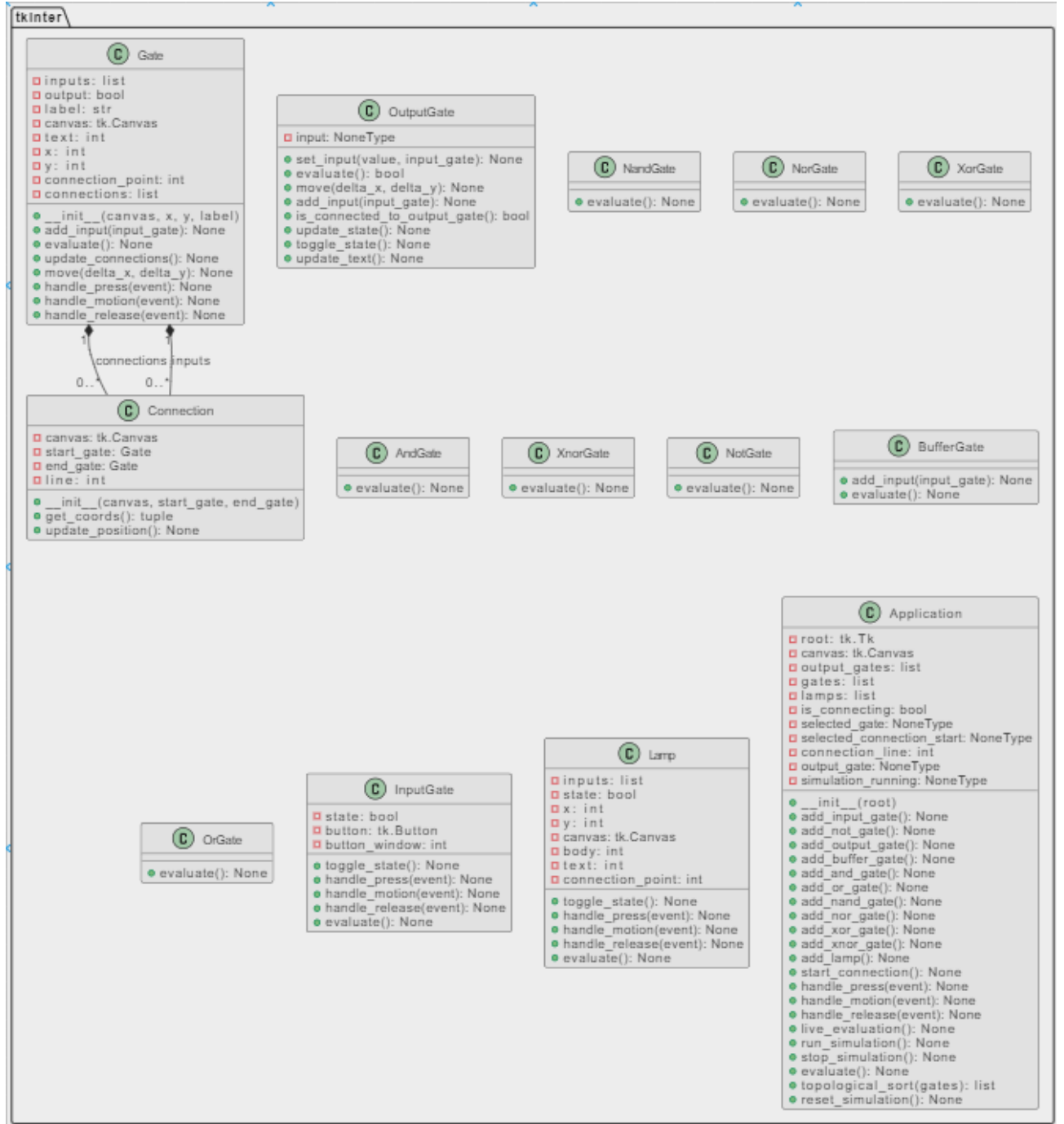
Simülasyon Tasarımı: Devrenin çalışmasını ve girişlerin değişmesiyle çıkışların nasıl etkileneceğini hesaplayan tasarım. Bu tasarım, kullanıcı girişlerine ve mantık kapısı modüllerine göre devreyi günceller.

3. Hata Yönetimi ve Güncelleme:

Hata ve Uyarı Sistemi: Geçersiz bağlantılar, çakışmalar veya hatalı girişler için kullanıcıya geri bildirim sağlayan sistem.

Gerçek Zamanlı Güncelleme: Kullanıcı girişlerine anında tepki vererek devrenin durumunu günceller.

- Modül diyagramı



3.2 Kullanılacak teknolojiler

Yazılım, Python programlama dili kullanılarak yazılmıştır.

Kullanılacak Harici Kütüphaneler Hakkında Bilgi

Tkinter: Python'un standart GUI (Graphical User Interface) kütüphanesidir.

Kullanıcı arayüzü oluşturma, düğme ve menü gibi bileşenleri kolayca ekleme ve düzenleme imkanı sağlar. Bu projede mantık kapıları ve bağlantıları görsel olarak temsil etmek için Tkinter kullanılacaktır.

MessageBox: Tkinter ile birlikte gelen bir modüldür. Kullanıcılara bilgi, uyarı ve hata mesajları göstermek için kullanılır.

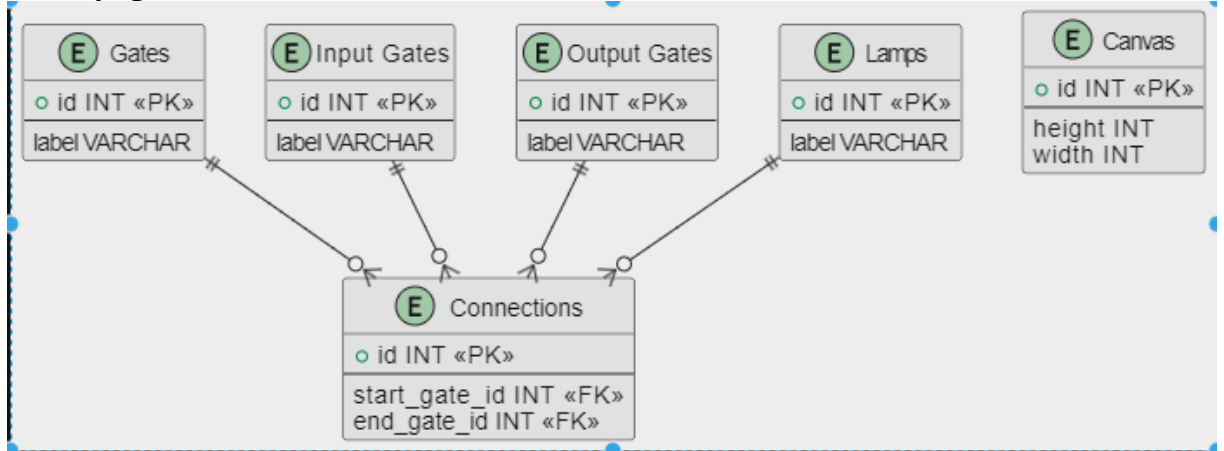
Varsa Diğer Teknolojiler ile İlgili Açıklama

Topolojik Sıralama: Mantık kapılarının doğru sırayla değerlendirilmesini sağlamak için kullanılacaktır. Bu yöntem, döngüsel bağımlılıkları olmayan yönlendirilmiş grafikleri sıralamak için kullanılır.

Canlı Değerlendirme (Live Evaluation): Mantık devresinin çalışmasını sürekli olarak değerlendirmek ve güncellemek için bir zamanlayıcı kullanımı. Bu, kullanıcı girişine göre devrenin dinamik olarak nasıl değiştiğini göstermek için önemlidir.

3.3 Veri tabanı tasarımı

ER diyagramı



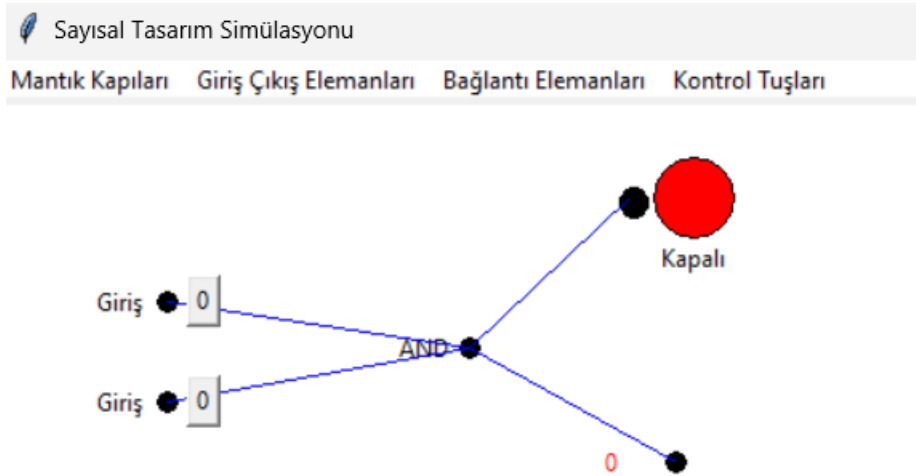
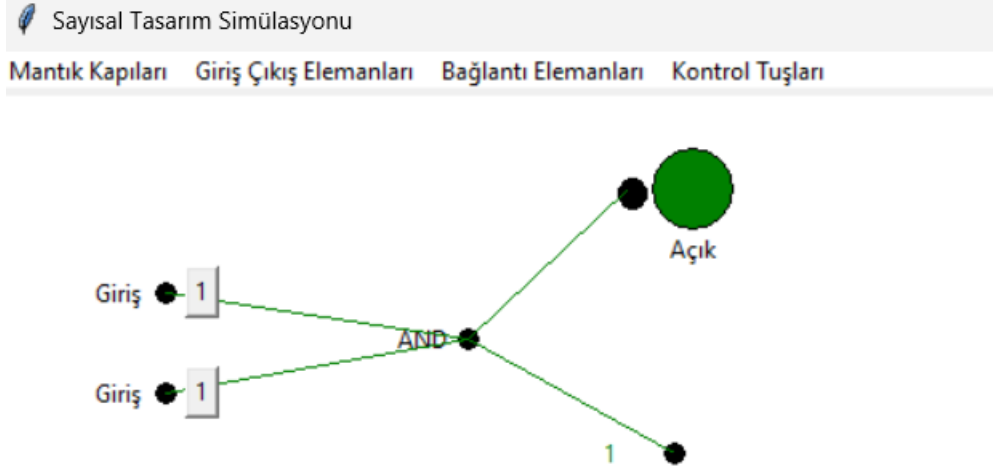
3.4 Kullanıcı arayüzü tasarımı

Uygulamaya kodu Python'a koyup tkinter kütüphanesini de ekleyip çalıştırınca ulaşılır.

Kullanıcı arayüzü tasarımı, verilen kod örneğinde tkinter modülü kullanılarak gerçekleştirilmiştir. Bu kod, mantık kapılarını, giriş çıkış elemanlarını ve bağlantı noktalarını görsel olarak temsil eden bir arayüz sağlar. Kullanıcı, arayüz üzerindeki menüler aracılığıyla farklı mantık kapılarını ekleyebilir, bunları birbirine bağlayabilir ve ardından simülasyonu başlatıp durdurabilir.

Arayüz, kullanıcıların kapıları sürükleyip bırakarak yerleştirebileceği ve ardından bu kapıları birbirine bağlayabileceği bir alana sahiptir. Ayrıca, kullanıcıların simülasyonu başlatıp durdurabileceği ve tüm sistemini

sıfırlayabileceği bir kontrol menüsü de bulunmaktadır. Bu şekilde, kullanıcılar basit bir dijital devre simülasyonu oluşturabilir ve çalıştırabilirler.



Şekildeki gibi uygulama girişte girilen değerlere göre bağlanan mantık kapısına bağlı sonuçlar verir.

4 UYGULAMA

4.3Kodlanan bileşenlerin açıklamaları

Connection: Bağlantıyı temsil eden bir sınıftır. İki mantık kapısı arasında bağlantı oluşturur ve bu bağlantıyı görsel olarak temsil eder.

Gate: Mantık kapılarının soyut bir sınıfıdır. Farklı mantık kapıları bu sınıftan türetilir. Her kapı, bağlantı noktalarını ve giriş/çıkış işlevselliğini sağlar.

InputGate: Giriş kapılarını temsil eden bir alt sınıftır. Kullanıcı tarafından

kontrol edilebilir ve giriş durumunu değiştirir.

OutputGate: Çıkış kapılarını temsil eden bir alt sınıftır. Girişlerini değerlendirir ve sonucu görsel olarak sunar.

Lamp: LED lambalarını temsil eden bir sınıftır. Bağlı oldukları kapılardan gelen sinyalleri gösterir.

Application: Uygulama sınıfı, tkinter kullanarak GUI (Grafiksel Kullanıcı Arayüzü) oluşturur. Kullanıcıların kapıları ekleyebileceği, bağlayabileceği ve simülasyonu yönetebileceği bir arayüz sağlar.

TestApplicationMethods: Dijital mantık devresi simülasyonunun doğru çalıştığını doğrulamak için bir dizi test içerir.

4.4Görev dağılımı

- Mantık kapılarını üçer üçer bölüp yazdık kalan kısımları da görev dağılımı yapıp bitirdik.
- Raporda da şemalar ve yazı kısmında görev dağılımı yaptık.

4.5Karşılaşılan zorluklar ve çözüm yöntemleri

- Çıkış tek bir tane oluyordu ama çıkışı applicationa düzgün bağlamadığımız için olmuyormuş sorunu çözdük.

4.6Proje isterlerine göre eksik yönler

- Özellik tablosundaki değerler tasarım alanına yerleştirilen elemanlar üzerinde sağ fare tuşu ile tıklanarak görüntülenebilir ve değiştirilebilir kısmında ne açıklanmaya çalıştığını anlayamadığımız için kendimize göre yaptık.
- Mantık kapılarını isimleriyle belirttik fotoğraf koymak yerine.

5 TEST VE DOĞRULAMA

5.3Yazılımın test süreci

Giriş Kapıları (InputGate):

İki giriş kapısı eklenir. Biri açık (True) yapılır, diğeri kapalı (False) olarak bırakılır.

OR Kapısı (OrGate):

İki giriş kapısından gelen sinyalleri alır. Girişlerden en az biri açık (True) ise çıkışı True yapar.

Lamba (Lamp):

OR kapısının çıkışına bağlanır ve OR kapısının çıkışı True olduğunda yanması beklenir.

Uygulama (Application):

Test, tüm bileşenlerin doğru şekilde eklenip bağlandığını ve simülasyonun

doğru şekilde çalıştığını doğrular.

5.4Yazılımın doğrulanması

Test Sonuçları

Tam ve Doğru Çalışan Bileşenler

InputGate:

- Test: İlk giriş kapısı (input_gate1) doğru şekilde oluşturuldu ve durumu True olarak ayarlandı.
- Sonuç: Giriş kapısı durumu doğru şekilde değiştirildi.

OrGate:

- Test: İki giriş kapısı OR kapısına doğru şekilde bağlandı ve OR kapısının çıkışı doğru değerlendirildi.
- Sonuç: OR kapısı girişlerden en az biri True olduğunda doğru çıkış verdi.

Lamp:

- Test: OR kapısının çıkışı lambaya doğru şekilde bağlandı ve OR kapısının çıkışı True olduğunda lamba yandı.
- Sonuç: Lamba, OR kapısının çıkışı True olduğunda doğru şekilde yandı.

Application:

- Test: Giriş kapıları, OR kapısı ve lamba uygulama içinde doğru şekilde eklendi ve bağlandı.
- Sonuç: Uygulama, bileşenlerin doğru şekilde eklenmesini ve bağlanmasını sağladı.

Github

<https://github.com/ajnasss/project6.git>

<https://github.com/Damla-keklik/3.git>