



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

SAYISAL TASARIM PROJESİ

ŞEVVAL ÖZEREN
220501028

DR. ÖęR. ÜYESİ ERCAN ÖLÇER

02.06.2024

<https://github.com/sevvalozrn>

1 GİRİŞ

1.1 Projenin amacı

- Bu projenin amacı, kullanıcıların basit mantık devrelerini tasarlayabileceği ve simüle edebileceği bir platform oluşturmaktır. Platform, mantık kapıları, giriş çıkış elemanları, bağlantı elemanları ve kontrol tuşları gibi temel araçları içerecek ve kullanıcıların bu elemanları tasarım alanına ekleyerek işlevsel devreler oluşturmaya olanak tanıyacaktır. Bu proje, basit mantık devreleri oluşturmak ve lojik kapıların çalışma şeklini öğrenmek için kullanılabilir.
- **Projede Gerçekleştirilmesi Beklenenler:**
 - Araçlar Kısmı ve Özellikleri:

Araçlar kısmında her bir mantık kapısını eklemek için “Mantık Kapıları”, devre girdileri ve çıktılarının kontrolü için “Giriş çıkış elemanları”, devre elemanları arası bağlantı sağlanması için “Bağlantı Elemanları” ve son olarak da devrenin çalışması, durması, baştan başlatılması gibi işlevleri sağlamak amaçlı “Kontrol Tuşları” araçları eklenmelidir.
 - Her Bir Araca Ait Özellik Tablosu:

Her bir araç devre oluşumu için devre yapım alanına eklendikten sonra etiket(isim) bilgisi, özellikleri ve tasarım alanından kaldırmak amaçlı bir “Delete” seçeneği bulunduran özellik tablosu bulunmalıdır. Özelliklerine göre; Mantık kapıları için giriş bağlantı sayısı, Giriş çıkış elemanları için renk ve başlangıç değeri (sadece giriş elemanları için), Bağlantı elemanları için ise renk. Tabii bu özelliklerin hepsinin başında etiket de bulunmakta. Bu özellik tablosuna aracın üzerinde sağ tık yapıldığında ulaşılabilinmekte.
 - Devre İçin Bağlantı Elemanları:

Araçları birbirine bağlayan bir kablo özelliği görmesi amacıyla “Bağlantı Kablosu”, kablo üzerinde gerektiğinde ek bağlantı kabloları çıkarabilmek amaçlı da “Bağlantı Noktası” aracı bulunmaktadır.
 - Tasarım Alanına Eleman Ekleme ve Devre Oluşturabilme:

Tasarım alanına aynı devre elemanından birden fazla eklenebilmelidir.
 - Devre Simülasyon Kontrolü:

Çalıştır, reset ve durdur tuşları ile devrenin çalışmasını simüle etme.
 - Giriş-Çıkış Elemanları İşlevleri:

Simülasyon esnasında kullanıcıların her bir elemanın giriş kutusunu seçerek giriş değerini değiştirebilmelidir. Çıkış elemanları için ise; Kullanıcı devrenin çıkışında led veya çıkış kutusu kullanabilmeli, Çıkış kutusu çıkış değerini göstermeli, Led kullanıldığında devrenin çıkış değerine göre ledin yanması ve sönmesi sağlanmalıdır.

2 GEREKSİNİM ANALİZİ

2.1 Arayüz gereksinimleri

- Kullanıcı arayüzü gereksinimleri:
 1. Araçlar Menüsü
Mantık Kapıları Araçları: AND, OR, NOT gibi temel mantık kapılarını içeren simgeler. Giriş Çıkış Elemanları: Giriş kutuları ve çıkış elemanları (LED, çıkış kutusu) simgeleri. Bağlantı Elemanları: Elemanlar arasında bağlantılar kurmak için kullanılacak çizgi araçları. Kontrol Tuşları: Simülasyonu başlatma, durdurma ve resetleme işlevleri için butonlar.
 2. Tasarım Alanı:
Kullanıcıların mantık devrelerini rahatça tasarlayabileceği geniş bir alan. Elemanları araç çubuğundan tasarım alanına kolayca sürükleyip bırakabilme.
 3. Araç Özellikleri Penceresi:
Her bir devre elemanının üzerine sağ tıklayarak özelliklerini görüntüleme ve değiştirme işlevlerine sahip olmalı. Seçilen araca göre (mantık kapısı, giriş/çıkış elemanı, bağlantı) dinamik olarak farklı kategoriler içeren bir pencere olmalı. Elemanların etiketleri ve renkleri değiştirilebilmeli. Giriş elemanları için başlangıç değerini belirleyebilme olmalı.
 4. Devre Oluşumu ve Simülasyon Kontrolü:
Elemanlar arasında bağlantı çizgileri ve bağlantı noktaları oluşturmak için araçlar bulunmalı. Devre simülasyonunu başlatma, durdurma ve sıfırlama işlevleri. Simülasyon esnasında giriş elemanlarının değerlerini değiştirebilme. Değer değişikliklerinin anında devreye yansıtılması. Devrenin çıkış değerini gösteren bir kutu veya devrenin çıkış değerine göre LED'in yanması veya sönmesi durumları.
 5. Kullanıcı Dostu Tasarım:
Her seviyeden kullanıcının rahatça kullanabileceği basit ve sezgisel bir tasarım.
- Donanım arayüzü gereksinimleri:
 1. Bilgisayar ve İşletim Sistemi:
Desteklenen İşletim Sistemleri: Windows 7/8/10 veya daha üstü.
Minimum Donanım Gereksinimleri: En az 4GB RAM, 2GHz çift çekirdekli işlemci ve 500MB boş disk alanı.
 2. Yazılım Kütüphaneleri:
C# Donanım Kütüphaneleri: Donanım arayüzlerini C# ile kolayca kontrol edebilmek için .NET uyumlu kütüphaneler.

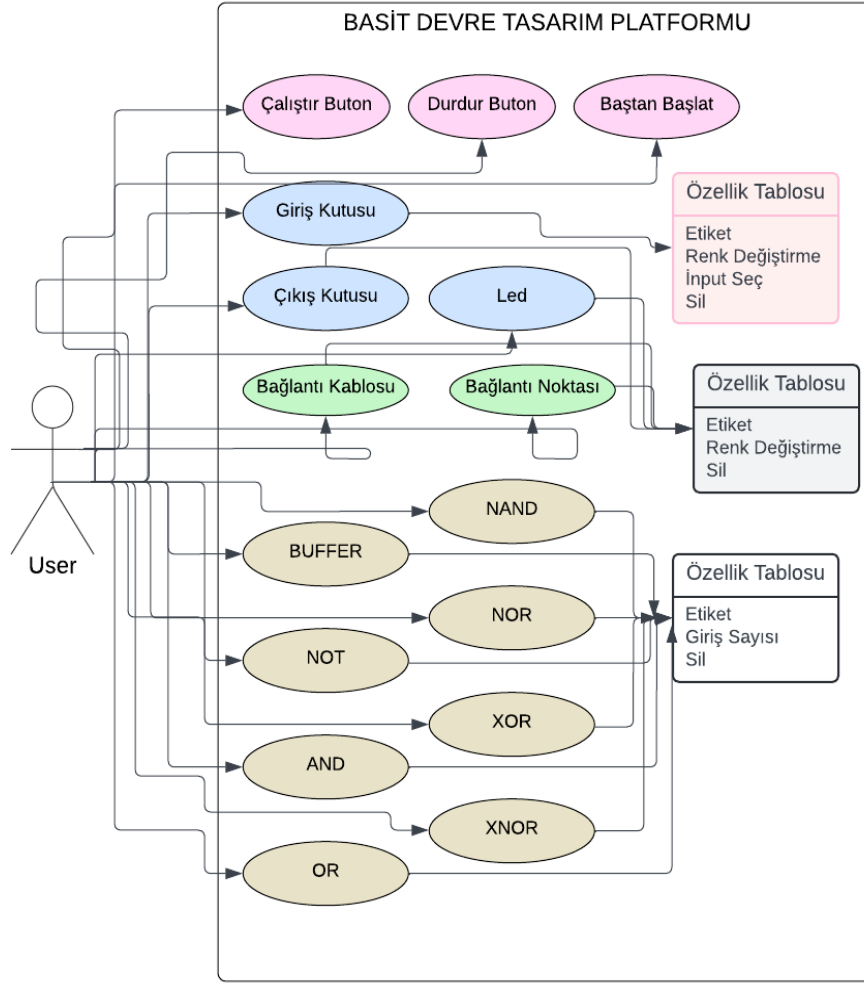
2.2 Fonksiyonel gereksinimler

- Fonksiyonel gereksinimler
Kullanıcıların mantık devrelerini tasarlayabilmesi, simüle edebilmesi ve bu süreçte ihtiyaç duyacakları tüm fonksiyonel gereksinimler aşağıda verilmiştir:
 - Kullanıcıların AND, OR, NOT gibi temel mantık kapılarını tasarım alanına ekleyebilmesi; Mantık kapılarının giriş bağlantı sayılarının dinamik olarak ayarlanabilmesi.
 - Giriş elemanlarının (input) tasarım alanına eklenebilmesi ve başlangıç değerlerinin ayarlanabilmesi; Çıkış elemanlarının (output) tasarım alanına eklenebilmesi ve çıkış değerlerinin gösterilebilmesi.
 - Araçların sürükle-bırak yöntemi ile tasarım alanına eklenebilmesi, tasarım alanındaki elemanların konumlarının ve boyutlarının ayarlanabilmesi, tasarım alanındaki elemanların sağ tıklama ile özelliklerinin görüntülenmesi ve

düzenlenebilmesi, tasarım alanına aynı türden birden fazla eleman eklenebilmesi gibi gereksinimler sağlanmalıdır.

- Simülasyonu başlatmak, durdurmak ve resetlemek için kontrol butonlarının kullanılması, çıkış elemanlarının durumu hakkında görsel geri bildirim sağlanması.

2.3 Use-Case diyagramı



3 TASARIM

3.1 Mimari tasarım

- Mimari tasarım:
 1. **Katmanlı Mimari:** Projemiz, katmanlı bir mimari yaklaşımıyla tasarlanmıştır. Temel olarak üç katmana sahiptir:
 - a. **Kullanıcı Arayüzü Katmanı:** Kullanıcı arayüzünü oluşturan, Windows Forms tarafından desteklenen bir GUI (Grafiksel Kullanıcı Arayüzü) katmanıdır. Kullanıcı arayüzü, kullanıcının projeyi görüntülemesini ve etkileşimde bulunmasını sağlar.

- b. **Uygulama Mantık Katmanı:** Bu katman, projenin iş mantığını içerir. İkonların eklenmesi, taşınması, bağlantı kurulması gibi temel işlevler bu katmanda gerçekleştirilir.
 - c. **Veri Erişim Katmanı:** Bu katman, projenin verilerine erişim sağlar. Ancak, bu projede basit bir veri erişim katmanı gerekli değildir, çünkü veritabanı veya dış veri kaynaklarına erişim gerekli değildir.
2. **Olay Temelli Programlama:** Projemizde olay temelli bir programlama yaklaşımı kullanılmıştır. Kullanıcı arayüzündeki etkileşimler (fare tıklamaları, sürükleyip bırakma vb.) olaylarla tetiklenir ve bu olaylar uygun iş mantığı fonksiyonlarına yönlendirilir.
3. **Görsel Bileşenlerin Kullanımı:** Projemiz, Windows Forms kullanılarak geliştirildiği için Windows Forms bileşenleri ve kontrolleri (PictureBox, Panel, ContextMenuStrip vb.) kullanılarak oluşturulmuştur. Bu bileşenler, kullanıcı arayüzünü oluşturmak için kullanılmıştır.
4. **Modüler Tasarım:** Projemizde her bir işlevsellik, kendi fonksiyonlarına ve özelliklerine sahip bağımsız bileşenler şeklinde tasarlanmıştır. Örneğin, ikon ekleme, taşıma, silme gibi işlevler kendi metodlarına ve etkileşimlerine sahip olan modüller olarak tasarlanmıştır.
5. **Tekrar Kullanılabilirlik:** Projemizin modüler tasarımı sayesinde, her bir işlevselliği farklı projelerde veya farklı bileşenlerde tekrar kullanabiliriz. Örneğin, ikon ekleme ve taşıma işlevselliği, başka bir Windows Forms uygulamasında da kullanılabilir.
6. **Kodun Organizasyonu:** Kod, uygun bir şekilde sınıflara ve yöntemlere ayrılarak okunabilirlik ve bakım kolaylığı sağlanmıştır. Her sınıf ve yöntem, belirli bir işlevselliği temsil eder ve kodun mantıksal olarak bölünmesini sağlar.

3.2 Kullanılacak teknolojiler

- Yazılım C# dili kullanılarak, Windows Forms Uygulaması(.NET Framework) şablonu üzerinden hazırlanmıştır.

3.3 Kullanıcı arayüzü tasarımı

- Kullanıcı arayüzü tasarımı ile ilgili açıklama:

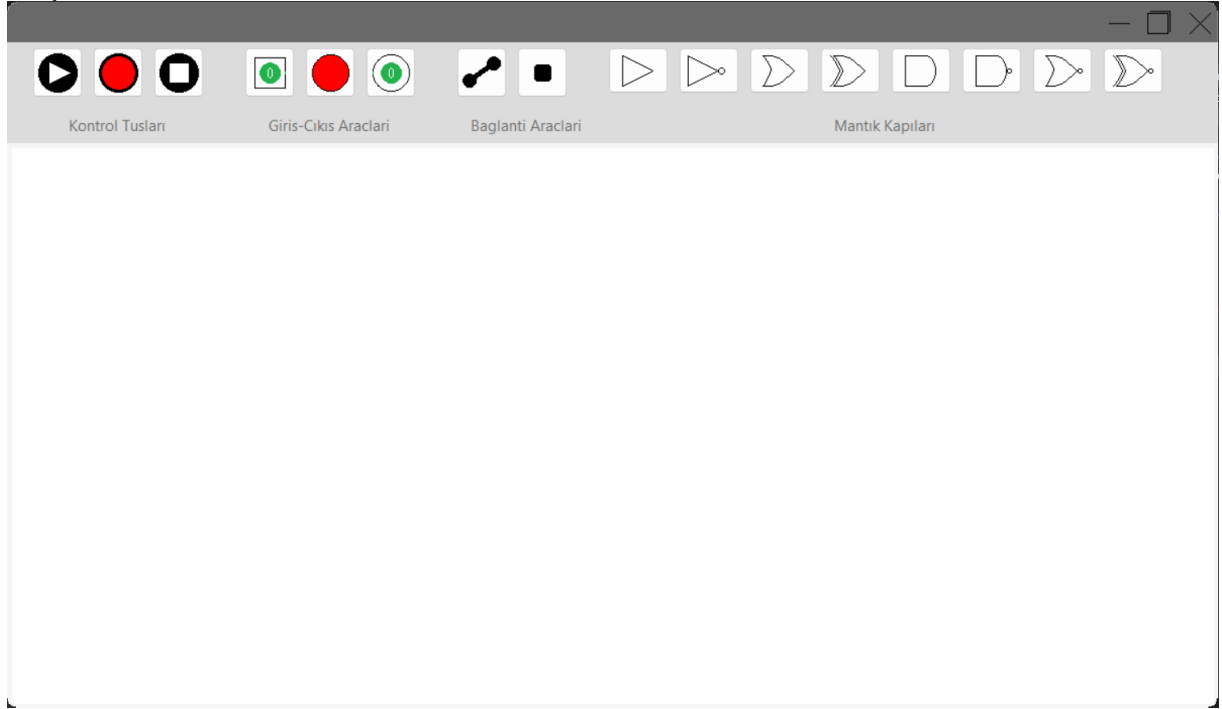
Bu projede Windows Forms uygulaması içindeki bir formun arka planında bulunan bir resim kutusu (PictureBox) üzerinde çizim yapma ve simgeleri sürükleyip bırakma özelliği ile yerleştirme yeteneğine sahiptir.

- Simgeler ekleme: Kullanıcı, "BUFFER", "NOT", "AND", "OR", "NOR", "NAND", "XOR", "XNOR" mantık kapıları simgelerini PictureBox üzerine ekleyebilir.
- Simgelerin özellikleri: Her simge için sağ tıklama menüsü, simge adı, giriş sayısı, renk değiştirme, silme gibi seçenekleri içerir.
- Bağlantı oluşturma: Kullanıcı, simgeler arasında bağlantılar çizmek için fareyi kullanabilir. Bu bağlantılar, mantık kapılarının çıkışlarını girişlere bağlamak için kullanılabilir.
- Bağlantı noktaları: Her simge, bağlantı noktaları olarak işlev gören küçük paneller içerir. Kullanıcılar, bağlantı noktalarına bağlantılar oluşturabilir.
- Bağlantı özellikleri: Kullanıcı, çizgilerin renklerini değiştirebilir veya silme işlemi yapabilir.

Kullanıcı arayüzü, simgeleri sürükleyip bırakmayı, sağ tıklama menüsüyle simge özelliklerini ve bağlantı noktalarını yönetmeyi içerir. Ayrıca, çizgi çizme ve çizgilerin özelliklerini düzenleme gibi işlevselliği de sağlar.

- Yazılımdan ekran çıktıları alınarak üzerinden açıklama yapılması:

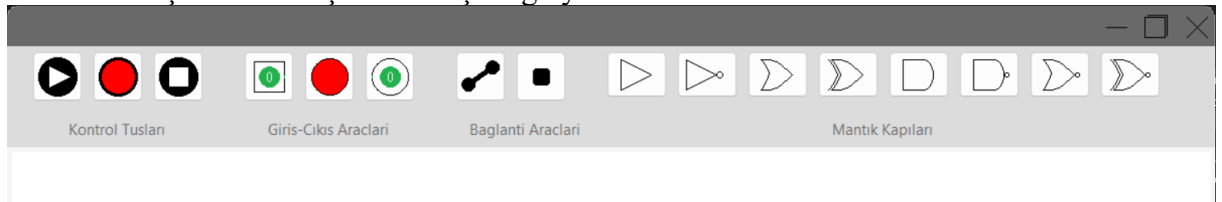
Uygulama çalıştırıldığında kullanıcıyı ilk olarak pencerenin merkezinde açılan bu pencere karşılamaktadır.



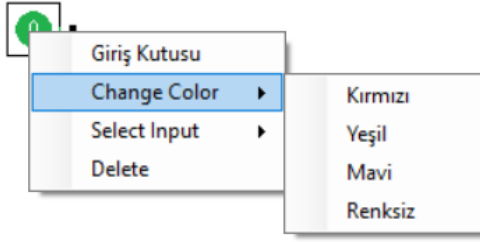
Pencereyi tam ekran yapmak veya orijinal boyutunda tutmak, pencereyi görev çubuğuna saklamak ve pencereyi kapatmak işlevlerine yerine getiren basit tasarımlı butonlar bulunmaktadır.



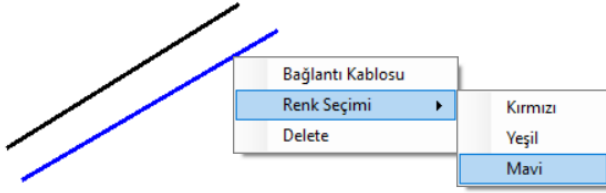
Kullanıcının devre tasarımını yapması için geniş bir alan ve kullanacağı araçlar ile kontrol butonlarını içeren bir araçlar menü çubuğu yer almaktadır.



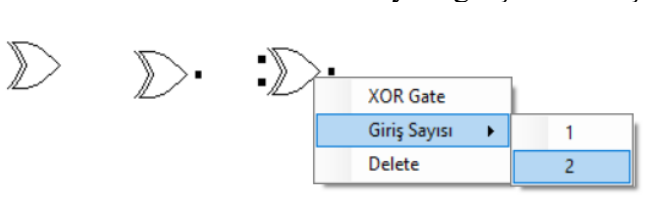
Kullanıcının ekrana eklediği araçlar üzerinde sağ tık yapılması sonucunda ufak bir özellikler tablosu görülmektedir. Giriş ve Çıkış değerleri için özellik tablosu: etiket, renk değişimi, input değeri(sadece girdiler için) ve aracı silmek amacıyla delete seçeneklerini barındırmaktadır.



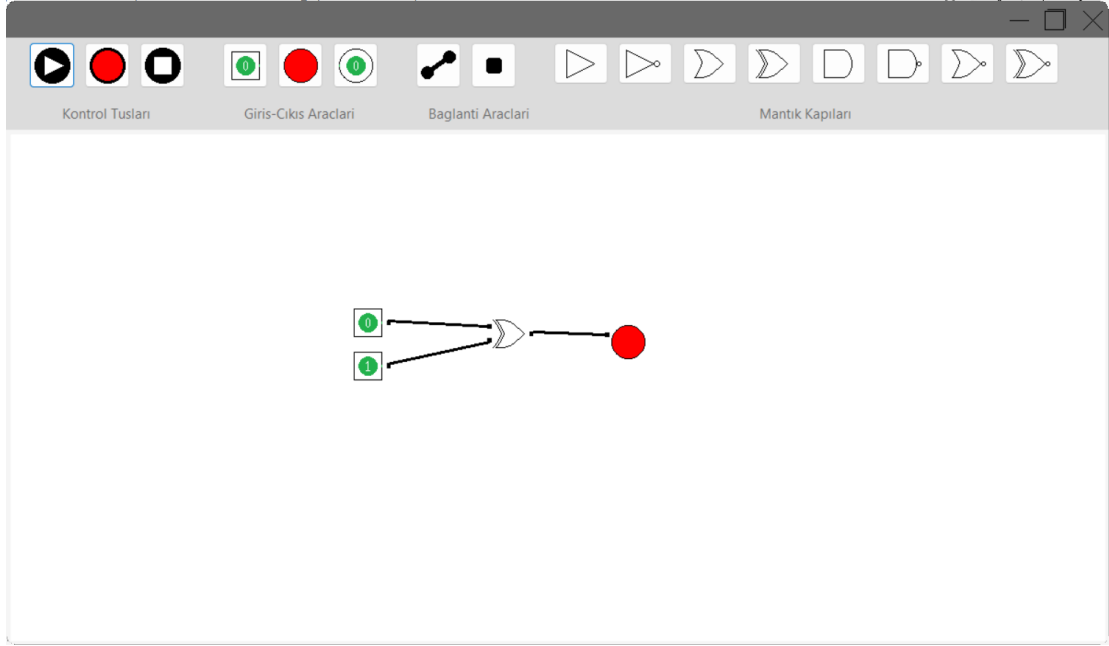
Bağlantı araçlarında da etiket, renk değişimi ve yine silme özellikleri bulunmaktadır. Bağlantı kablosu ilk eklendiğinde siyah renk gözükürken seçeneklerde kırmızı, mavi veya yeşil renklerine dönüştürülebilmektedir.



Mantık kapıları ise tasarım alanına ilk eklendiğinde en soldaki gibi gözükmektedir. Üzerinde sağ tık yapıldığında menü açılmakla birlikte çıkış noktası için bir bağlantı noktası eklenir. Açılan özellikler menüsünden 1 veya 2 girişli olacak şekilde seçim yapılabilmektedir.



- Uygulamanın nasıl çalıştırılacağı ile ilgili açıklama:
Uygulama kodun çalıştırılması ile başlatılmaktadır. Uygulama açıldıktan sonra oluşturulacak devreye yönelik elemanlar eklenerek ve bağlantı kabloları ile bağlanarak devre simülasyonu için devre tasarımı yapılır.
Aşağıda örnek olarak basit bir devre tasarımı yapılmıştır.



Devre oluşturulduktan sonra da çalıştır tuşuna basılarak devrenin çalışma simülasyonu gerçekleştirilmelidir.

Ancak bu projede çalıştır, durdur, baştan gibi butonlar işlevsizdir, devre simülasyonu yapılamamaktadır.

4 UYGULAMA

4.1 Kodlanan bileşenlerin açıklamaları

- `public partial class Form1 : Form`

Form1, Windows Forms uygulamasının ana formunu temsil eder. Bu sınıf, formun tasarımını ve olaylarını yöneten sınıftır.

- `public Form1() { }`

Form1 sınıfının constructor'ı. Bu metot, form oluşturulduğunda çağrılır.

InitializeComponent() metoduyla form bileşenleri başlatılır. Bağlam menülerini ve çizgi bağlam menüsünü başlatır:

```
InitializeComponent();
```

```
InitializeContextMenu();
```

```
InitializeLineContextMenu();
```

Fare olaylarına ilişkin işlemler atanır:

```
picBox.MouseDown += PictureBox_MouseDown;
```

```
picBox.MouseMove += PictureBox_MouseMove;
```

```
picBox.MouseUp += PictureBox_MouseUp;
```

```
picBox.Paint += picBox_Paint;
```

- Bağlam Menüsü İçin İşlevler:

Bağlam menüsü, fare sağ tıklamasıyla açılan menülerdir. İkonları üzerine sağ tıklandığında bu menüler görünür.

- InitializeContextMenu(): İkonların bağlam menülerini başlatır.

- ShowContextMenu(): Belirli bir ikon üzerinde bağlam menüsünü görüntüler.
- DeleteMenuItem_Click(): İkonları silmek için kullanılan işlev.
- AddConnectionPoints(): İkonlara bağlantı noktaları ekler.

- **Bağlam Menüsü İçin Giriş Kutusu İşlevleri:**

Giriş kutusu ikonları için bağlam menüsü işlevlerini tanımlar.

- InitializeContextMenuGirisK(): Giriş kutusu ikonları için bağlam menülerini başlatır.
- DeleteMenuItem_ClickGirisK(): Giriş kutusu ikonlarını silmek için kullanılan işlev.

- **Bağlam Menüsü İçin Çıkış Kutusu İşlevleri:**

Çıkış kutusu ikonları için bağlam menüsü işlevlerini tanımlar.

- InitializeContextMenuCikis(): Çıkış kutusu ikonları için bağlam menülerini başlatır.
- DeleteMenuItem_ClickCikis(): Çıkış kutusu ikonlarını silmek için kullanılan işlev.

○

- **Bağlantı Noktaları Ekleme:**

- AddConnectionPoints(): İkonlara bağlantı noktaları ekler.
- AddConnectionPointsRight(): İkonlara sağ taraftan bağlantı noktaları ekler.

○

- **Çizgi Çizme İşlevleri:**

- Cizgi_Click(): Çizgi çizme işlevini başlatır.
- Line: Çizgi sınıfı, çizgilerin özelliklerini (başlangıç, bitiş, renk vb.) tutar.
- IsPointOnLine(): Bir noktanın çizginin üzerinde olup olmadığını kontrol eder.

- **Bağlantı Noktalarını Bulma:**

- FindConnectionPoint(): Belirli bir noktanın yakınındaki bağlantı noktasını bulur. Bununla bağlantı kabloları yakınındaki noktaya otomatik bağlanır.

4.2 Karşılaşılan zorluklar ve çözüm yöntemleri

- Geliştirme süreci C# dilinde tecrübemin olmamasından kaynaklı zorlayıcı oldu. Projeye başlarken ne yapacağım konusunda neredeyse hiçbir fikrim yoktu. Bu yüzden Youtube üzerinden oluşturulmuş benzer yapıdaki projelerin yapım aşamaları ve C# ile ilgili tutorial videoları izleyerek bu zorluğu giderdim.

4.3 Proje isterlerine göre eksik yönler

Projede gerçekleştirilmesi beklenen kriterler arasında oluşturulmuş devrenin mantık kapılarına göre çıktı vermesi yani devrenin simülasyonun sağlanması da yer almaktaydı ancak bu projede simülasyon kısmında eksikler bulunmaktadır.

5 KAYNAKLAR

- <https://lucid.app>
- <https://youtu.be/PivjmGq0W-Y?si=ec64O9PV9CZIts8e>

- <https://youtu.be/XIpUOnGC2lY?si=zXGnD9iOtT6M1zx1>
- <https://learn.microsoft.com>