

# MİKROİŞLEMCİLER TRAFİK IŞIKLARI PROJESİ RAPORU

## 1. PROJE KONUSU VE DETAYLAR

1.1 Proje temel olarak bir trafik lambasının çalışma prensibinden oluşmaktadır.

1.2 Projenin simülasyon aşamasındaki trafik ışıkları gerçek hayattaki trafik ışıklarıyla aynı mekanizmada çalışmaktadır.

1.3 Projede Kuzey-Güney ve Doğu-Batı yönlerinde hareket edecek 4 adet araç bulunmaktadır ve bu araçların her birinin 25 adet geçişi en kısa sürede tamamlanması istenmektedir

1.4 Bu geçişlerin en kısa sürede tamamlanması için kuzey yönüne giden araç ile güney yönüne giden araç aynı anda hareket etmelidir. Çünkü bu iki araç birbirine çarpamaz aynı anda geçiş yapmaları en kısa sürede tamamlanması koşuluna uygundur.

1.5 Birinci aşama için tüm araçlara kırmızı ışık yanacaktır.

1.6 İkinci aşama için Kuzey-Güney yönünde hareket eden araçlar için sarı-kırmızı ışık aynı anda yanacak, Doğu-Batı yönünde hareket eden araçlar için kırmızı ışık yanacaktır.

1.7 Üçüncü aşama için Kuzey-Güney yönünde hareket eden araçlar için yeşil ışık yanacak ve 25 araç geçişi beklenecektir(Kodda süre ile tanımlanmıştır süre azaltılır veya arttırılırsa araç geçişleri istenildiği sayıda yapılabilir). Doğu-Batı yönünde hareket eden araçlar için kırmızı ışık yanmaya devam edecektir

1.8 Dördüncü aşama için Kuzey-Güney yönünde hareket eden araçlar geçişlerini tamamlamış olacak ve yavaşlamaları için sarı ışık yanacaktır. Doğu-Batı yönünde hareket eden araçlar için sarı-kırmızı ışık yanacak, harekete hazır durumda olacaklardır.

1.9 Beşinci aşama için Kuzey-Güney yönünde hareket eden araçlara kırmızı ışık yanacak ve duracaklardır. Doğu-Batı yönünde hareket eden araçlara sarı-kırmızı ışık yanmaya devam edecek ve harekete hazır durumda olacaklardır.

1.10 Altıncı aşama için Kuzey-Güney yönünde hareket eden araçlara kırmızı ışık yanmaya devam edecektir. Doğu-Batı yönünde hareket eden araçlar için yeşil ışık yanacak ve onlar için de 25 adet aracın geçiş yapması beklenecektir(Kodda süre ile tanımlanmıştır süre azaltılır veya arttırılırsa araç geçişleri istenildiği sayıda yapılabilir).

1.11 Yedinci aşama için Kuzey-Güney yönünde hareket eden araçlara kırmızı ışık yanmaya devam edecektir. Doğu-batı yönünde hareket eden araçlar geçişlerini tamamlamış olacak ve yavaşlamaları için sarı ışık yanacaktır.

1.12 Sekizinci ve son aşamada her bir yön için kırmızı ışık yanacak ve simülasyon sonlandırılacaktır.

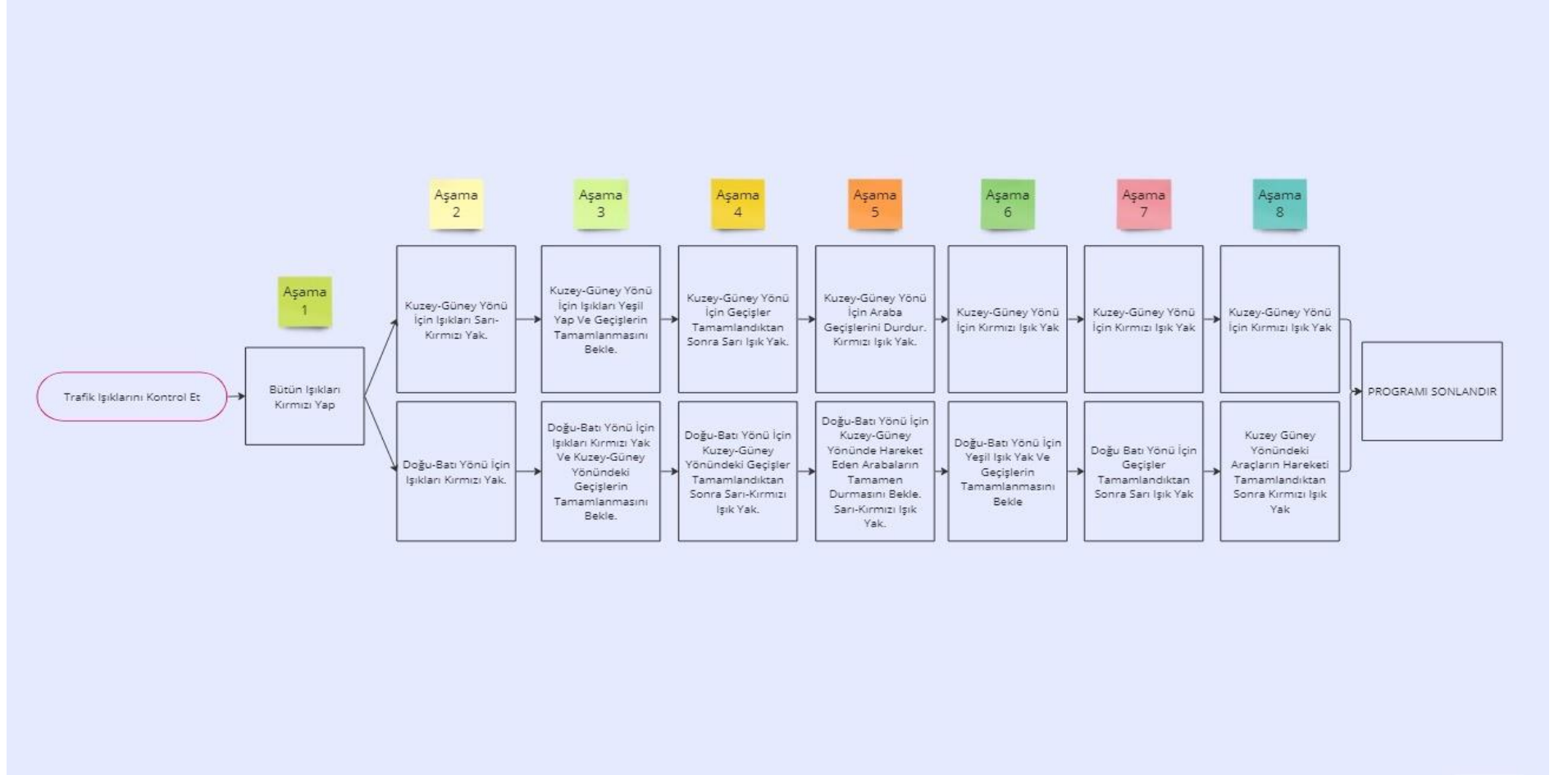
Proje oluřturulurken planlanan btn ařamalar gnlk yařamımızda kullandığımız trafik ışıkları ile aynı özelliklerde çalışmaktadır. Proje ile alakalı resimler ilerleyen bölmlerde paylaşılmıştır.



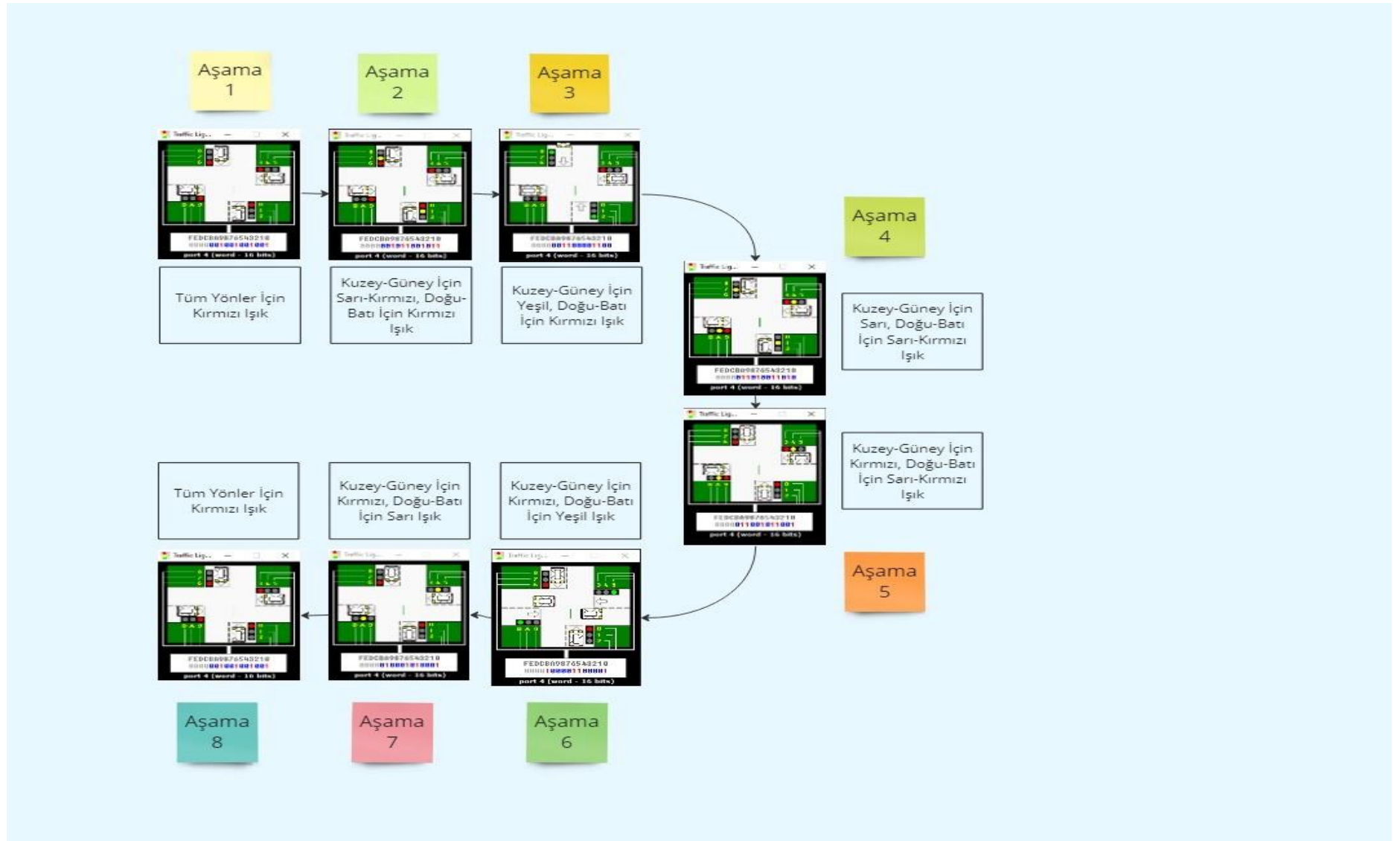
GNLK HAYATTA TRAFİK İŐİKLARININ YANMA ŐEKİLLERİ

## 2. AKIŞ DİYAGRAMLARI

Projemizin 2 adet akış diyagramı bulunmaktadır. Bir tanesi projenin temel olarak anlatıldığı sıradan bir akış diyagramı. Bir tanesi anlatımı kuvvetlendirmek için oluşturulmuş olan proje çıktılarının bulunduğu akış diyagramı



PROJENİN TEMEL OLARAK ANLATILDIĞI AKIŞ DİYAGRAMI



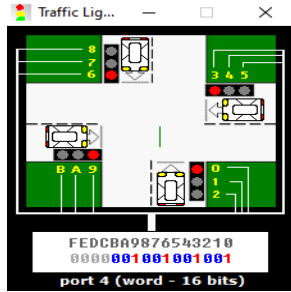
PROJENİN ÇIKTILARININ BULUNDUĞU AKIŞ DİYAGRAMI

### 3. ASSEMBLY KOD AÇIKLAMALARI

Kodun tamamı yanlarındaki açıklama satırları ile birlikte bu bölümün sonunda verilmiştir.

#### 3.1

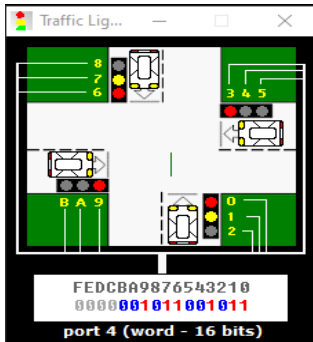
```
MOV CX, 150          ; Sure belirlemek icin kullaniyoruz (CX kaydedicisini sayac olarak kullaniyoruz)
LoopStart1:
MOV AX, 0000001001001001B    ;kuzey-guney kirmizi / dogu-bati kirmizi
OUT 4, AX ; AX kaydedicisindeki verilerin 4. cikis portuna gonderilmesi icin kullanilir.(Cikis portuna belirtilen bit degerlerini atar)
LOOP LoopStart1 ; Donguyu baslatir
```



Tüm ışıklar kırmızı yanarken kullandığımız kod satırları ve uygulama görüntüsü.

#### 3.2

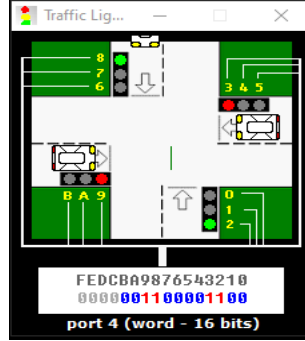
```
MOV CX, 150          ; Sure belirlemek icin kullaniyoruz (CX kaydedicisini sayac olarak kullaniyoruz)
LoopStart2:
MOV AX, 0000001011001011B    ;kuzey-guney sarı-kirmizi / dogu-bati kirmizi
OUT 4, AX ; AX kaydedicisindeki verilerin 4. cikis portuna gonderilmesi icin kullanilir.(Cikis portuna belirtilen bit degerlerini atar)
LOOP LoopStart2 ; Donguyu baslatir
```



Kuzey-Güney yönünde ışıklar sarı-kırmızı, Doğu-Batı yönünde ışıklar kırmızı yanarken kullandığımız kod satırları ve uygulama görüntüsü.

### 3.3

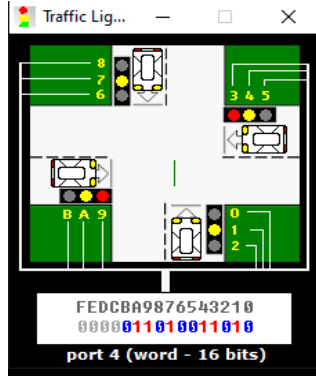
```
MOV CX, 3750      ; Sure belirlemek icin kullaniyoruz (CX kaydedicisini sayac olarak kullaniyoruz)
LoopStart3:
MOV AX, 0000001100001100B      ;kuzey guney icin yesil isik / dogu-bati kirmizi
OUT 4, AX      ; AX kaydedicisindeki verilerin 4. cikis portuna gonderilmesi icin kullanilir.(Cikis portuna belirtilen bit degerlerini atar)
LOOP LoopStart3      ; Donguyu baslatir
```



Kuzey-Güney yönünde ışıklar yeşil, Doğu-Batı yönünde ışıklar kırmızı yanarken kullandığımız kod satırları ve uygulama görüntüsü.

### 3.4

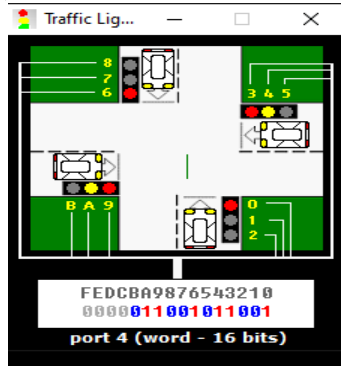
```
MOV CX, 100      ; Sure belirlemek icin kullaniyoruz (CX kaydedicisini sayac olarak kullaniyoruz)
LoopStart4:
MOV AX, 0000011010011010B      ;kuzey-guney sarı/ dogu-bati sarı-kirmizi
OUT 4, AX      ; AX kaydedicisindeki verilerin 4. cikis portuna gonderilmesi icin kullanilir.(Cikis portuna belirtilen bit degerlerini atar)
LOOP LoopStart4      ; Donguyu baslatir
```



Kuzey-Güney yönünde ışıklar sarı, Doğu-Batı yönünde ışıklar sarı-kırmızı yanarken kullandığımız kod satırları ve uygulama görüntüsü.

### 3.5

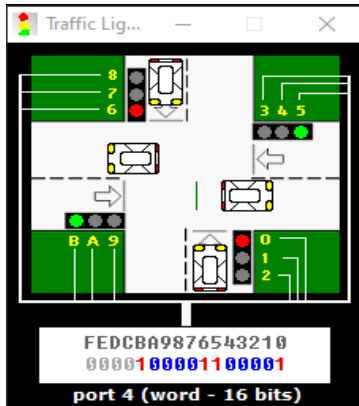
```
MOV CX, 100          ; Sure belirlemek icin kullaniyoruz (CX kaydedicisini sayac olarak kullaniyoruz)
LoopStart5:
MOV AX, 0000011001011001B      ; kuzey-guney kirmizi / dogu-bati sarı-kirmizi
OUT 4, AX      ; AX kaydedicisindeki verilerin 4. cikis portuna gonderilmesi icin kullanilir. (Cikis portuna belirtilen bit degerlerini atar)
LOOP LoopStart5      ; Donguyu baslatir
```



Kuzey-Güney yönünde ışıklar kırmızı, Doğu-Batı yönünde ışıklar sarı-kırmızı yanarken kullandığımız kod satırları ve uygulama görüntüsü.

### 3.6

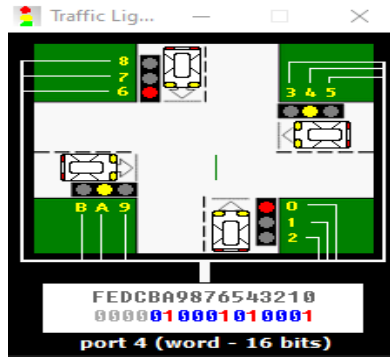
```
MOV CX, 3750          ; Sure belirlemek icin kullaniyoruz (CX kaydedicisini sayac olarak kullaniyoruz)
LoopStart6:
MOV AX, 0000100001100001B      ; kuzey-guney kirmizi / dogu-bati yesil
OUT 4, AX      ; AX kaydedicisindeki verilerin 4. cikis portuna gonderilmesi icin kullanilir. (Cikis portuna belirtilen bit degerlerini atar)
LOOP LoopStart6      ; Donguyu baslatir
```



Kuzey-Güney yönünde ışıklar kırmızı, Doğu-Batı yönünde ışıklar yeşil yanarken kullandığımız kod satırları ve uygulama görüntüsü.

3.7

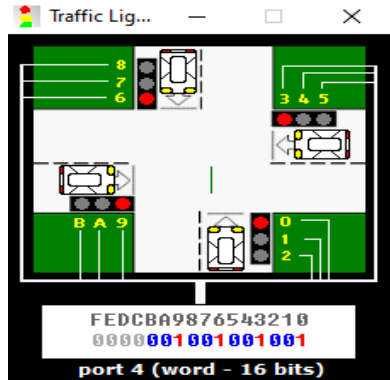
```
MOV CX, 100      ; Sure belirlemek icin kullaniyoruz (CX kaydedicisini sayac olarak kullaniyoruz)
LoopStart7:
MOV AX, 0000010001010001B      ;kuzey-guney kirmizi / dogu-bati sari
OUT 4, AX      ; AX kaydedicisindeki verilerin 4. cikis portuna gonderilmesi icin kullanilir.(Cikis portuna belirtilen bit degerlerini atar)
LOOP LoopStart7      ; Donguyu baslatir
```



Kuzey-Güney yönünde ışıklar kırmızı, Doğu-Batı yönünde ışıklar sarı yanarken kullandığımız kod satırları ve uygulama görüntüsü.

3.8

```
MOV CX, 100      ; Sure belirlemek icin kullaniyoruz (CX kaydedicisini sayac olarak kullaniyoruz)
LoopStart8:
MOV AX, 00000010001001001B      ;kuzey-guney kirmizi / dogu-bati kirmizi
OUT 4, AX      ; AX kaydedicisindeki verilerin 4. cikis portuna gonderilmesi icin kullanilir.(Cikis portuna belirtilen bit degerlerini atar)
LOOP LoopStart8      ; Donguyu baslatir
```



Tüm ışıklar kırmızı yanarken kullandığımız kod satırları ve uygulama görüntüsü.



KOD SATIRLARI:

#start=Traffic\_Lights.exe#

ORG 100h

MOV CX, 150 ; Sure belirlemek için kullanıyoruz (CX kaydedicisini sayac olarak kullanıyoruz)

LoopStart1:

MOV AX, 0000001001001001B ;kuzey-guney kirmizi / dogu-bati kirmizi

OUT 4, AX ; AX kaydedicisindeki verilerin 4. çıkış portuna gönderilmesi için kullanılır.(Çıkış portuna belirtilen bit değerlerini atar)

LOOP LoopStart1 ; Donguyu baslatır

MOV CX, 150 ; Sure belirlemek için kullanıyoruz (CX kaydedicisini sayac olarak kullanıyoruz)

LoopStart2:

MOV AX, 0000001011001011B ;kuzey-guney sari-kirmizi / dogu-bati kirmizi

OUT 4, AX ; AX kaydedicisindeki verilerin 4. çıkış portuna gönderilmesi için kullanılır.(Çıkış portuna belirtilen bit değerlerini atar)

LOOP LoopStart2 ; Donguyu baslatır

MOV CX, 3750 ; Sure belirlemek için kullanıyoruz (CX kaydedicisini sayac olarak kullanıyoruz)

LoopStart3:

MOV AX, 0000001100001100B ;kuzey guney için yesil isik / dogu-bati kirmizi

OUT 4, AX ; AX kaydedicisindeki verilerin 4. çıkış portuna gönderilmesi için kullanılır.(Çıkış portuna belirtilen bit değerlerini atar)

LOOP LoopStart3 ; Donguyu baslatır

MOV CX, 100 ; Sure belirlemek için kullanıyoruz (CX kaydedicisini sayac olarak kullanıyoruz)

LoopStart4:

MOV AX, 0000011010011010B ;kuzey-guney sari/ dogu-bati sari-kirmizi

OUT 4, AX ; AX kaydedicisindeki verilerin 4. çıkış portuna gönderilmesi için kullanılır.(Çıkış portuna belirtilen bit değerlerini atar)

LOOP LoopStart4 ; Donguyu baslatır

MOV CX, 100 ; Sure belirlemek için kullanıyoruz (CX kaydedicisini sayac olarak kullanıyoruz)

LoopStart5:

MOV AX, 0000011001011001B ;kuzey-guney kirmizi/ dogu-bati sari-kirmizi

OUT 4, AX ; AX kaydedicisindeki verilerin 4. çıkış portuna gönderilmesi için kullanılır.(Çıkış portuna belirtilen bit değerlerini atar)

LOOP LoopStart5 ; Donguyu baslatır

MOV CX, 3750 ; Sure belirlemek için kullanıyoruz (CX kaydedicisini sayac olarak kullanıyoruz)

LoopStart6:

MOV AX, 0000100001100001B ;kuzey-guney kirmizi / dogu-bati yesil

OUT 4, AX ; AX kaydedicisindeki verilerin 4. çıkış portuna gönderilmesi için kullanılır.(Çıkış portuna belirtilen bit değerlerini atar)

LOOP LoopStart6 ; Donguyu baslatır

MOV CX, 100 ; Sure belirlemek için kullanıyoruz (CX kaydedicisini sayac olarak kullanıyoruz)

LoopStart7:

MOV AX, 0000010001010001B ;kuzey-guney kirmizi / dogu-bati sari

OUT 4, AX ; AX kaydedicisindeki verilerin 4. çıkış portuna gönderilmesi için kullanılır.(Çıkış portuna belirtilen bit değerlerini atar)

LOOP LoopStart7 ; Donguyu baslatır

```

MOV CX, 100 ; Sure belirlemek için kullanıyoruz (CX kaydedicisini sayac olarak kullanıyoruz)

LoopStart8:

MOV AX, 0000001001001001B ;kuzey-guney kirmizi / dogu-bati kirmizi

OUT 4, AX ; AX kaydedicisindeki verilerin 4. çıkis portuna gönderilmesi için kullanılır.(Çıkis
portuna belirtilen bit degerlerini atar)
LOOP LoopStart8 ; Donguyu baslatir
MOV AH, 4CH ; AH'ye programin çıkis kodunu atar.
INT 21H ; Bu islemci kesmesi, programin cikis yapmasini saglar.
end ; Programin sonunu belirtir.

```

(YEŞİL KISIMLAR İŞARETLENEN YERLER KODLARIN AÇIKLAMALARIDIR)

#### 4. FLAGLAR VE RESGISTER DEĞERLERİ

Kodumuzda bulunan döngülerde sürekli değişken kısmı CX bölümüdür. Bu sebeple flag kısmında programın başından sonuna bir değişim bulunmamaktadır. CF=0 ZF=0 SF=0 OF=0 PF=0 AF=0 IF=1 DF=0 olarak başlamakta ve sonuna kadar bu şekilde devam etmektedir.

Register değerleri için ise CX kısmı sayaç olarak kullanıldığı için herhangi bir anında değerini belirlemek pek mümkün değildir. Başlangıçta atadığımız değerden başlayarak 0000 değerine kadar hızlı bir şekilde azalarak devam etmektedir. Fakat her bir döngünün kendine ait bir AX değeri vardır. Somut olarak belirtebileceğimiz tek register değerleri bunlardır. Aynı zamanda IP değeri de CX değeri gibi sürekli değişerek devam etmektedir. Diğer register değerlerinde bir değişim olmamaktadır. Yani program başladığında alınan değerler ile sonundaki değerler aynı değerlerdir.

BX: 00 00 DX: 00 00 CS:F400 SS:0700 SP:FFFE BP:0000 SI:0000 DI:0000 DS:7000 ES:7000

AX DEĞERLERİ İÇİN;

LOOP1: 02 49,

LOOP2: 02 CB,

LOOP3: 03 0C,

LOOP4: 06 9A,

LOOP5: 06 59,

LOOP6: 08 61,

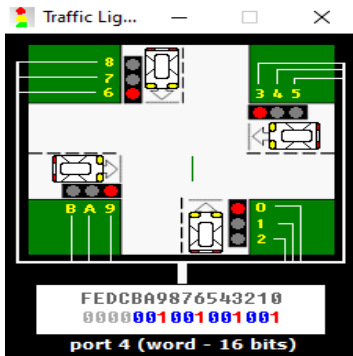
LOOP7: 04 51,

LOOP8: 02 49,

Program sonlandığında AX: 4C 49 ve SP: FFF8 olmaktadır.

## 5- PROGRAM ÇIKTILARI

5.1



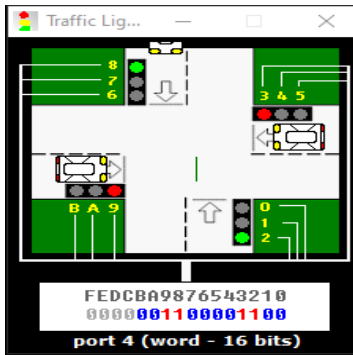
Simülasyon anında tüm ışıklar kırmızı yanarken.

5.2



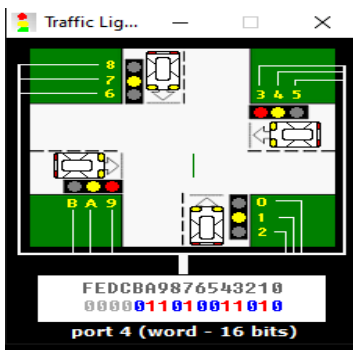
Simülasyon anında Kuzey-Güney yönündeki araçlar geçişe hazırlanırken.

5.3



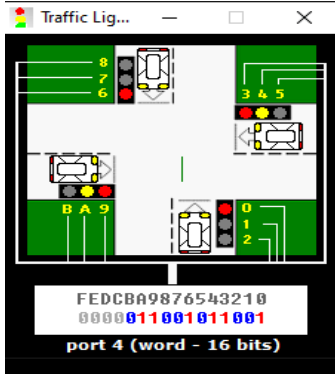
Simülasyon anında Kuzey-Güney yönündeki araçlar geçiş yaparken.

5.4



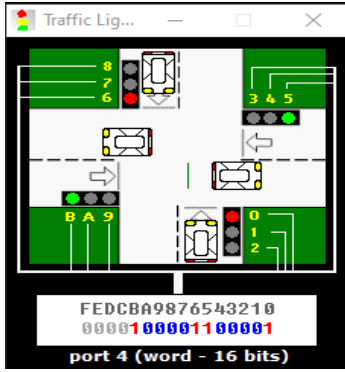
Simülasyon anında Kuzey-Güney yönündeki araçlar durmaya, Doğu-Batı yönündeki araçlar geçiş yapmaya hazırlanırken.

5.5



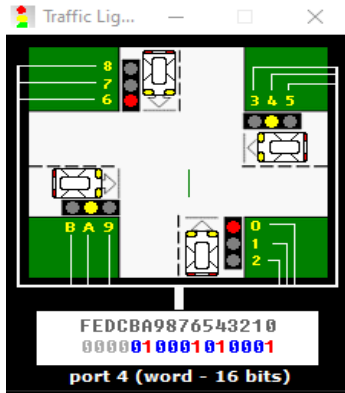
Simülasyon Anında Kuzey-Güney yönünde hareket eden araçlar için kırmızı ışık yanarken, Doğu-Batı yönündeki araçlar geçiş yapmaya hazırlanırken.

5.6



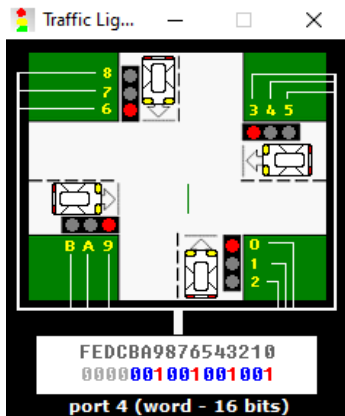
Simülasyon anında Doğu-Batı yönündeki araçlar geçiş yaparken.

5.7



Simülasyon anında Doğu-Batı yönündeki araçlar durmaya hazırlanırken.

5.8



Simülasyon anında tüm ışıklar kırmızı yanarken.

## 6. SONUÇLAR ve ÖĞRENDİKLERİMİZ

### 6.1 SONUÇLAR

6.1.1 Bir trafik lambasının çalışma mekanizmasının oluşturulması,

6.1.2 Yeşil, Sarı, Kırmızı ışıkların günlük yaşamda çalıştığı şekilde belirli bir uyum içerisinde ve düzenli aralıklar ile çalışması,

6.1.3 Yanan ışıkların belirli bir döngü içerisinde düzenli zaman aralıklarıyla çalışması ve programı sonlandırmadığımız sürece çalışmaya devam etmesi,

6.1.4 Araçların geçiş yaptıkları esnada herhangi bir kazanın oluşmaması için döngülerin ve zamanların uyumlu olarak ayarlanması,

6.1.5 Araçların geçişlerini tamamladıktan sonra trafik kurallarına ve ışık direktiflerini uygun bir şekilde durması,

6.1.6 Özellikle sarı ışığın kullanımı konusunda dikkat edilmesi ve gerçek trafik lambalarıyla birebir aynı özelliklerde olması (Araçların harekete geçmeden sarı-kırmızı ışığın aynı anda yanması ve durmadan önce sadece sarı ışığın yanması),

6.1.7 Trafik ışıklarında zamanlama ve koordinasyon gibi durumların önemi ve planlanması,

Açıklama 1: Yukarıda verilen özellikler projeyi hazırlarken dikkat ettiğimiz konuları ve projenin son halinin içeriği hakkında bilgi vermektedir.

Açıklama 2: Aynı zamanda proje, Örnek proje konularında verilen özelliklere uygun ve geçişlerin en kısa sürede tamamlanması hususunda dikkate alınarak yapılmıştır. Zaman aralığı değiştirilerek araç geçiş sayısı artırılabilir veya azaltılabilir.

### 6.2 ÖĞRENDİKLERİMİZ

6.2.1 Bu proje ile elde edilen sonuçlar oldukça başarılıdır. Gerçek dünyadaki bir trafik ışığı sisteminin işleyişini başarılı bir şekilde simüle ettik. Assembly dili kullanarak bu karmaşık bir işlemi basitleştirmeyi ve anlaşılır kılmayı başardık.

6.2.2 Assembly dilini kullanma yeteneklerini geliştirmek için önemli bir fırsat olduğunu gördük. Assembly dili, bilgisayarların daha derin seviyelerini anlamak için kritik öneme sahip bir dil olduğunu öğrendik. İçerdiği düşük seviye talimatlar sayesinde, bilgisayarların işleyişini ve programlamanın daha yüksek seviye dillerine kıyasla daha iyi anladık.

6.2.3 Zamanlama ve koordinasyonu dengelemeyi öğrenmemiz bu projede önemli bir faktör oldu. Trafik ışıklarının döngüsünün düzgün bir şekilde işlemesi, doğru zamanlamanın ve koordinasyonun anlaşılmasını gerektirir. Bu, ileride yapacağımız projelerde faydalı olacak değerli bir beceri olduğunu göstermiştir.

6.2.4 Son olarak, bu projede, trafik ışıklarının koordinasyonunun genel trafik yönetimine etkisini de gözlemledik. Özellikle kesişme noktalarında, birden fazla trafik ışığı setinin uyumlu bir şekilde çalışması, trafiğin sorunsuz bir şekilde ilerlemesini sağlar. Bu durum, kesişme noktalarında meydana gelebilecek trafik kazalarını ve sıkışıklığını önlemek için önemlidir.

PROJEYİ HAZIRLAYANLAR:

MUHAMMET ALTEN-210501042

MUHAMMED KÜLÜNK-210501008

HAMZA YİĞİT UZUNALI-210501012

ÖĞRETİM GÖREVLİSİ:

DR. ÖĞR. ÜYESİ. EMRE ÖZEREN