LOJİK DEVRE BENZETİMİ

Eyyüb Ahmet Yıldırım, Alparslan Beraat Özdemir
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
Kocaeli, Türkçe
170202014@kocaeli.edu.tr, alparslanozdemir@outlook.com.tr

ÖZET

Programımız kullanıcıdan girilen komutlar yardımıyla text dosyalarındaki verilerden yararlanarak oluşturulan devreyi kullanıcıdan tarafından girilen komutlarla giriş uçları değiştirilerek simüle etmektedir. Bu işlemleri yaparken programımız log.txt dosyasına yapılan tüm işlemleri kaydetmektedir.

I. GİRİŞ

Lojik Devre Benzetimi, içerisinde main fonksiyonu ile birlikte 23 fonksiyon ve 2 struct yapısı içerir.

Bu fonksiyonlardan bir kısmı kapıların özelliklerini, bir kısmı ise konsoldan girilecek değerleri temsil eder. Struct yapıları ise devredeki tüm girişleri, çıkışları ve kapı değerlerini tutan ve işlemlerin üzerinde yapıldığı yer olarak tasarlanmıştır.

Toplam algılanabilir 9 farklı komut bulunmaktadır. Bu 9 komuttan help komutu, diğer 8 komutun nasıl kullanılacağının anlatıldığı bir çıktı vererek kullanıcıyı bilgilendirici komuttur. Program komutlarda case-sensitive(büyük-küçük harf duyarlı) değildir.

Kullanıcı lojik devre benzetimi yapmak için programa belirtilen formatta yazılarak devre tanımı yapılmış bir metin belgesini dosya adı belirterek yükleyebilir. Devre dosyası yükledikten sonra devrede bulunan giriş çıkış uçları için farklı bir değer dosyası yükleyerek uçlara değer ataması beklenir.

Programda belli komutlar diğerlerinden önce çalıştırılmalıdır. Örneğin devre yüklemesi yapılmadan değer yüklemesi yapılamaz.

II. TEMEL BİLGİLER

Proje gelişiminde;

Tümleşik geliştirme ortamı olarak Code::Blocks 17.12 kullanılmıştır.

Program Windows 10 İşletim sisteminde test edilmiştir. Program C dilinde geliştirilmiştir.

III. TASARIM

Lojik devre modellemesinin programlanma aşamaları altta belirtilen başlıklar altında açıklanmıştır.

A. Fonksiyonlar

void log(char komut[20],char cikti[200]) :

Program çalışırken log kaydını tutarken kullandığımız fonksiyon. Log kaydı komutu kullancağımız yerlere komut ve cikti stringlerini fonksiyona görerek kullandığımız fonksiyon.

int AND(int kapi_i) ,int OR(int kapi_i),

int NAND(int kapi_i),int XOR(int kapi_i), int NOR(int kapi_i)

Fonksiyon ismi olan olan kapının sonucunu return eden fonksiyonlar.

int kapi_sonuc(int kapi_i):

Program çalışırken hangi kapıda işlem yapacağını bulan fonksiyon. Kapıyı bulunca yukardaki kapı fonksiyonları çağırıyor kendi içinde.

void Y(char x[20]):

Kullanıcının konsola Y <"devre.txt"> girmesi sonucu çağırılan fonksiyondur.Fonksiyona gönderilen x değeri kullanıcın konsola girdiği veridir. Bu fonksiyon girilen text dosyası içerinde tanımlanmış devreyi programımıza atayıp programda devrenin oluşmasını sağlar.

void baskadosya_ekle(char baskadosya[10]):

Yukarıdaki Y fonksiyonun içinde çağırılan bu fonksiyonumuz eğerki Y de çağırılan text dosyasının içinde .include <"baskadevre.txt"> şeklinde bir başka devre ekleme işlemi varsa çalışır. "baskadevre.txt" içindeki kapıyı devreye ekler.

void I(char x[20]):

Kullanıcın konsola I <deger.txt> komutunu girmesi sonucu çalışır. "deger.txt" içindeki bilinmeyenler ait başlangıç değerlerinin verilmesini sağlar. Bu fonksiyonu Y fonksiyonunu kullanmadan kullanamazsınız.

int intyap(char x):

I fonksiyonundaki char veriyi atoi() fonksiyonu ile int çevirmek istediğimizde aldığımız hatadan dolayı bu fonksiyonu yazdık. Sadece char olan 0 ve 1 sayılarını int olan 0 ve 1 e çevirmektedir.

void H(char x[20]):

Kullanıcın konsola - H a – komutu girmesi ile eğer a değeri 0 ise 1 yapan fonksiyon. Fonksiyon a'nın değerini değiştirdikten sonra hesapla(char girilen) fonksiyonun içinde devre nin yeni değerleri belirlenir.

void L(char x[20]):

H fonksiyonun aynısıdır. Sadece girilen değeri 1 ise 0 yapar.

void hesapla(char girilen):

H ve L fonksiyonlarına gönderilen değerlerin değişmesi sonrası devreyi bir nevi çalıştırır ve değişen değerleri uygular.

void islem_tekar(int indis):

hesapla fonksiyonun içinde tekrar eden işlemler olduğu için kodun uzamaması için bu fonksiyona tekrar eden kısmı yazıp hesapla fonksiyonunda bu fonksiyonu çağırdık.

void G(char girilen[20]):

G* yada G a komutları yazıldığında çalışan fonksiyonumuz. G* yazılmışsa devredeki tüm giriş çıkış ve ara noktalarının lojik değerlerini ekrana basar. G a şeklinde giriş olursa a nın lojik değerini ekrana yazar.

void S(char x[20]:

H ve L fonksiyonlarındaki değişimler sonrası olanları simüle eden fonksiyonumuz.

void A():

Bizden istediğiniz komutlar dışında bir komut bu. Devredeki kapıları giriş ,çıkış,giriş sayısı,gecikme süresi gibi tüm bilgileriyle ekrana yazan fonksiyonumuz.

void HELP():

Konsoldan girilecek komutların işlevlerini ekrana basan fonksiyonumuz.

void ara(char x[20]):

Konsola girilen komutları kendi içinde arayıp ilgili fonksiyonu çağıran fonksiyonumuz.

void K(char x[20]):

K <"komut.txt"> girilmesi ile çalışır. Kullanıcının "komut.txt" içine daha önce yazmış olduğu komutları icra eder.

B. Algoritma

Algoritma, bir sonraki sayfada bulunan akış şeması ile gösterilmiştir.

IV. SONUÇLAR VE EKRAN ÇIKTILARI

Help komutu kullanıldığında program çıktısı şekil 1'de görünmektedir.

```
>help
komut                                   icrasi
.....................................................................................
Y <devre.txt>           | "devre.txt" dosyasindan devreyi yukler.
I <deger.txt>           | "deger.txt" icindeki degerlerlerle devre ilklendirilir.
H <giris ucundan biri>  |           ilgili uc u lojik-1 yapar
L <giris ucundan biri>  |           ilgili uc n lojik-0 yapar
G <giris ucundan biri>  | ilgili uc un seviyesini (0 veya 1) konsolda gosterir
G*                      | tum uclarin seviyesini (0 veya 1) konsolda gosterir
A                       |           tum kapilari gosterir
C                       |           benzetimden cikis yapar
H dan sonra bosluk birakmak zorundasiniz..
>
```

Şekil 1.

G komutunun ekran görüntüsü Şekil 2'dedir.

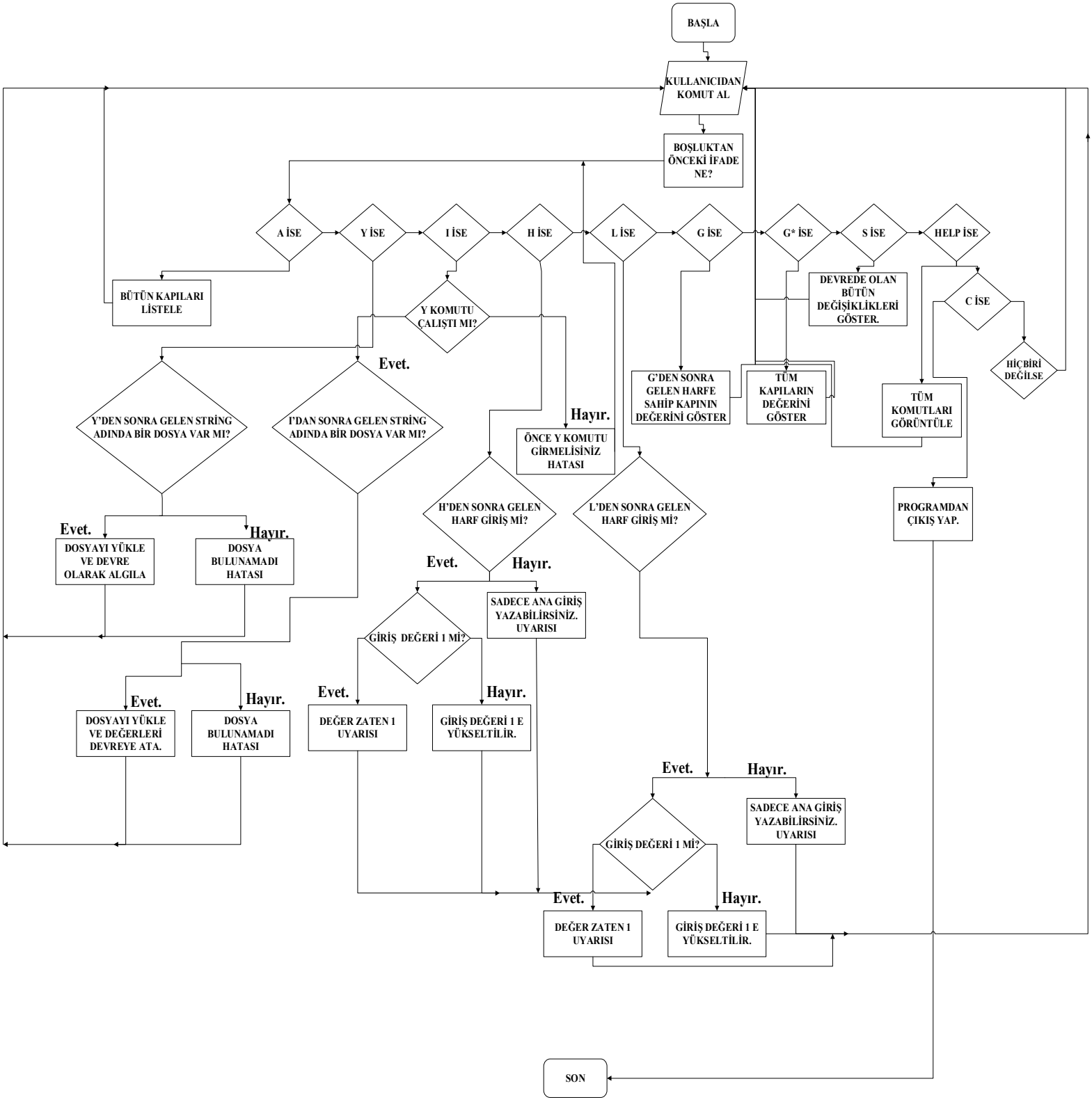
```
>G*
a = 0
b = 0
d = 1
c = 1
e = 1
f = 0
```

Şekil 2.

KAYNAKÇA

[1] http://www.baskent.edu.tr/~tkaracay/etudio/ders/prg/c/kutuk.htm

[2] http://bilgisayarkavramlari.sadievrenseker.com/2009/01/01/c-ile-zaman-islemleri/


BAŞLA
KULLANICIDAN KOMUT AL
BOŞLUKTAN ÖNCEKİ İFADE NE?
A İSE
Y İSE
I İSE
H İSE
L İSE
G İSE
G* İSE
S İSE
HELP İSE
BÜTÜN KAPILARI LİSTELE
Y KOMUTU ÇALIŞTI MI?
DEVREDE OLAN BÜTÜN DEĞİŞİKLİKLERİ GÖSTER.
C İSE
HİÇBİRİ DEĞİLSE
Evet.
Hayır.
Y'DEN SONRA GELEN STRİNG ADINDA BİR DOSYA VAR MI?
I'DAN SONRA GELEN STRİNG ADINDA BİR DOSYA VAR MI?
ÖNCE Y KOMUTU GİRMELİSİNİZ HATASI
G'DEN SONRA GELEN HARFE SAHİP KAPININ DEĞERİNİ GÖSTER
TÜM KAPILARIN DEĞERİNİ GÖSTER
TÜM KOMUTLARI GÖRÜNTÜLE
PROGRAMDAN ÇIKIŞ YAP.
H'DEN SONRA GELEN HARF GİRİŞ Mİ?
L'DEN SONRA GELEN HARF GİRİŞ Mİ?
DOSYAYI YÜKLE VE DEVRE OLARAK ALGILA
DOSYA BULUNAMADI HATASI
SADECE ANA GİRİŞ YAZABİLİRSİNİZ. UYARISI
GİRİŞ DEĞERİ 1 Mİ?
DOSYAYI YÜKLE VE DEĞERLERİ DEVREYE ATA.
DOSYA BULUNAMADI HATASI
DEĞER ZATEN 1 UYARISI
GİRİŞ DEĞERİ 1 E YÜKSELTİLİR.
SADECE ANA GİRİŞ YAZABİLİRSİNİZ. UYARISI
GİRİŞ DEĞERİ 1 Mİ?
DEĞER ZATEN 1 UYARISI
GİRİŞ DEĞERİ 1 E YÜKSELTİLİR.
SON