**PROGRAMLAMA LABARATUVARI 2.PROJE**

**170201043-İLKER TINKIR, 170201075-ALİ EKEN**

**ÖZET:**

Bu projede bizimle paylaşılan isterler doğrultusunda mantık kapılarıyla kurulan birkaç aşamalı bir devrenin kapı gecikmeleri de dikkate alınarak bilgisayar ortamına aktarılmasını sağlayan ve kullanıcıdan alınan komutlar doğrultusunda kapıların verdiği sonuçların simüle edilebildiği bir bilgisayar programı yazdık.

**GİRİŞ:**

Yazdığımız program kullanıcıdan çıkış bilgisi alınan kadar sürekli komut almak amacıyla sonsuz bir While döngüsüyle başlamaktadır. While döngüsü her döndüğünde kullanıcıdan bir karakter girişi istenmekte ve o girişe göre komutun bulunduğu fonksiyona yönlendirilmektedir. Kullanıcının girişleri her seferinde kontrol edilip eğer doğru giriş yapıldıysa kullanıcının komutu girdiği tarih bilgisi dosya işlemleri bilgisiyle programın çalışması esnasında açılan “log.txt” dosyası içerisine formatlı bir biçimde yazılmaktadır. Programda, bizlere belirtildiği gibi “Y, I, H, L, S, G, G\*, K, C” komutları bulunmaktadır.

Y komutu programın bulunduğu dizin içerisindeki devre.txt dosyasını okuma modunda açıp içerisindeki bilgileri daha sonra kullanmak üzere program içerisinde açılmış char dizisine kaydetmektedir. Eğer devre.txt dosyasının başlangıcında .include ifadesi varsa, program aynı klasör içerisindeki baska\_dosya.txt isimli dosyayı devre.txt dosyasıyla birleştirerek bir char dizisine atar. Böylece ayrı ayrı verilmiş iki devrenin tümleşik hale getirilmesi sağlanmış olur.

I komutu programın bulunduğu dizin içerisindeki deger.txt dosyasını okuma modunda açarak içerisindeki kapı giriş ve çıkışları program içerisinde oluşturulmuş char dizisine sırasıyla kaydeder. Bunların değerlerini ise program içerisinde oluşturulmuş bir integer dizisine sırasıyla kaydeder. Bu integer ve char dizilerinin aynı indexlerinde aynı kapı giriş veya çıkış noktasının bilgileri tutulur. Böylece bu bilgilere daha sonra ulaşmak istediğimizde kolaylık sağlanmış olur.

H komutu Y ve I komutlarıyla sanal biçimde oluşturulmuş devrenin giriş uçlarının değerlerinin “bir” yapılmasını sağlar. Bununla birlikte H komutu daha sonra S komutu için kullanılacak değişimleri içerisinde çalıştırılan fonksiyonlar yardımıyla düzenler.

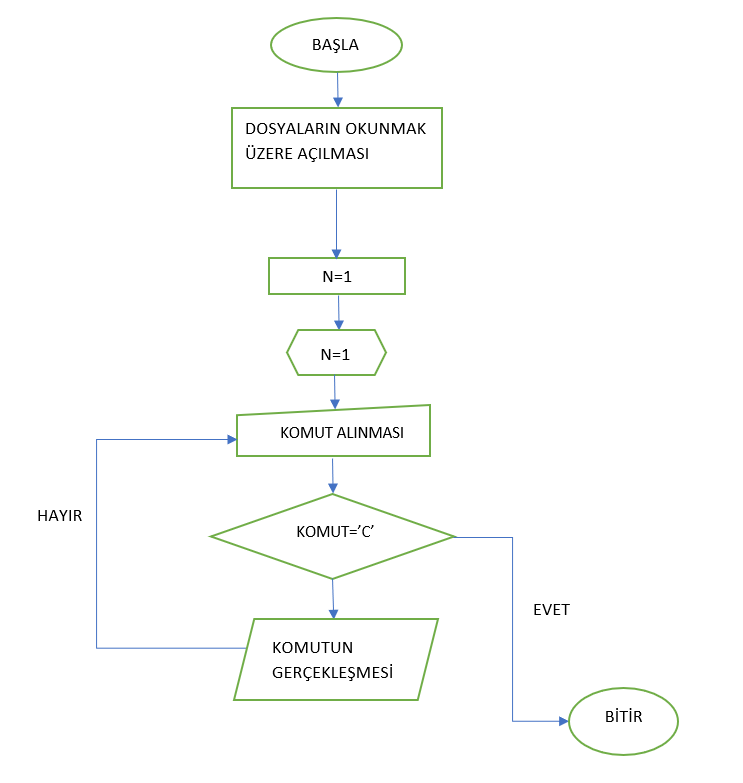
L komutu H komutuyla benzer biçimde çalışır. Komut çalıştırıldığında kullanıcının girdiği kapı girişlerini “sıfır” yapar. Daha sonra H komutunda da gerçekleşen işlemler sırasıyla gerçekleşir.

G komutu kullanıcının istediği kapı giriş çıkış uçlarınının işlemler yapıldıktan sonraki son değerlerini ekrana yazdırır. Eğer komut “G\*” biçiminde kullanılırsa devredeki tüm giriş çıkış uçlarının değerlerini sırasıyla ekrana basar.

K komutu programın bulunduğu dizindeki komut.txt dosyasını okuma modunda açarak içerisindeki komutları sırasıyla icra eder.

C komutu ise exit fonksiyonunu çağırarak programın sona erdirilmesini sağlar.

**AKIŞ ŞEMASI:**



**YÖNTEM:**

Yazdığımız programın daha basit ve dinamik bir biçimde olması için main fonksiyonunda çok uzun kodlar yazmamaya özen gösterdik. Bunun yerine her bir komut ve işlem için ayrı ayrı fonksiyonlar kullanarak programımızı oluşturduk. Kullanıcı programı sonlandırana dek sürekli komut alınması gerektiği için bir sonsuz döngü yapısı kullandık. Programda var olan diziler ve değişkenlerin farklı fonksiyonlarda ve main fonksiyonunda da kullanılması gerektiği için global olarak tanımladık. Programda komutun gerçekleştirildiği zaman bilgisinin dosyaya yazılması işlemini gerçekleştirmek için time.h kütüphanesi içerisindeki time\_t struct yapısını kullandık.

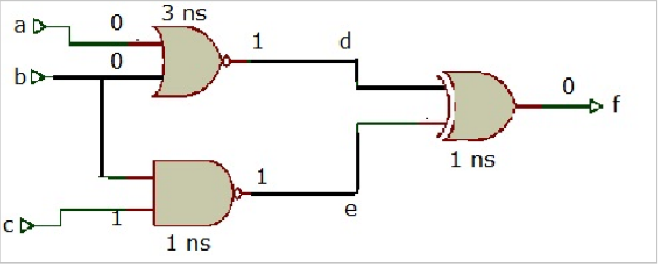
Simüle komutunun gerçekleştirilmesi için programda tanımladığımız kapılarla aynı adları taşıyan fonksiyonları kullandık. Kapıların çalışma zamanını hesaplamak için kapı gecikmelerinin, kapıların kendilerinden önceki kapıların gecikmelerine bağlı olarak dinamik bir biçimde değiştiği bir fonksiyon yapısı kullandık.

Programda varsa baska\_dosya.txt dosyasının devre.txt dosyasındaki devre ile birleştirilmesi için devre.txt içinde .include anahtar kelimesinin sorgulandığı bir yapı oluşturduk. Bu yapıyı baska\_dosya.txt içindeki kapı satırlarını programda oluşturduğumuz char dizisindeki kapı satırları arasına, giriş değişkenlerini ise char dizisindeki giriş satırına ekleyecek şekilde düzenledik.

**YAPILAN ARAŞTIRMALAR:**

Programda dosya işlemlerini kullanmamız gerektiği için gerek birinci sınıftaki Programlama 2 dersi slaytlarını gerekse internetteki diğer dosya işlemleriyle ilgili dökümanları inceledik. İncelememiz sonucu programımızda sadece “r” ve “w” biçiminde açılmış dosyalar kullanmaya karar verdik. Kapılara gelen giriş sinyallerinin çıktılarını daha kolay ve hızlı ifade edebilmek için araştırmalar yaptık ve sonucunda Bitwise operatörlerin projemizde yararlı olacağını fark ettik. Bu Bitwise operatörleri kapı fonksiyonlarında kullandık. Programımızda kullanmak üzere txt dosyaları yerine YAML dosya formatını araştırsak da bu format kullanmadık.

**ÖRNEK SONUÇLAR:**



Örnek olarak yukarıdaki devreyi inceleyelim ve b ucunu 1 yaptığımızı düşünelim. Bu durumda sırasıyla aşağıdaki işlemler gerçekleşecektir.

1. 0.ns de b ucu 0 dan 1 değerine eşitlenir.
2. Program önce ilk aşamadaki kapılardan daha küçük gecikme süresine sahip olanını tespit eder.
3. Bu devre için b ve c uçlarını giriş kabul eden 1 ns gecikmeli NAND kapısı 1.ns de çalışır ve e ucu 1 değerinden 0 değerine döner.
4. Bu aşamada program çıkış kapısının gecikme süresinin daha küçük olduğunu tespit ettiği için XOR kapısını çalıştırır ve 2.ns de f ucu 0 değerinden 1 değerine döner.
5. Bu aşamada NOR kapısının gecikme süresine ulaşıldığı için NOR kapısı çalışır ve 3.ns de d ucu 1 değerinden 0 değerine döner.
6. Bu aşamada ise çıkış kapısının gecikme süresi geldiğinde tekrar çalışması gerektiği için devredeki XOR kapısı çalışır ve 4.ns de f ucu 1 değerinden 0 değerine döner.
7. Yukarıdaki işlemler göz önüne alındığında sonuç:
8. ns b: 0->1
9. ns e: 1->0
10. ns f: 0->1
11. ns d: 1->0
12. ns f: 1->0

olarak bulunur.

**KAYNAKÇA:**

1. <https://www.programiz.com/c-programming/bitwise-operators>
2. <http://bilgisayarkavramlari.sadievrenseker.com/2008/10/22/c-ile-dosya-islemleri/>
3. <http://bilgisayarkavramlari.sadievrenseker.com/2009/01/01/c-ile-zaman-islemleri/>
4. <http://www.baskent.edu.tr/~tkaracay/etudio/ders/prg/c/time.pdf>
5. <http://embedded.kocaeli.edu.tr/>
6. <https://kodedu.com/2014/08/yaml-veri-degisim-formati-nedir/>