

Proje #2 - Lojik Devre Benzetimi

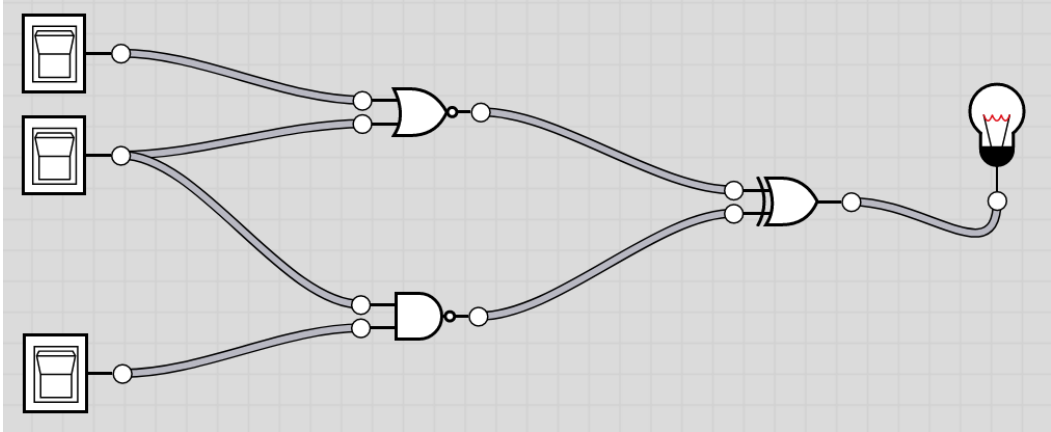
Dosya Operasyonları Üzerine Uygulama Geliştirme

Arş. Gör. Süleyman Eken & Arş. Gör. Furkan Göz

BAŞLANGIÇ: 29 Ekim 2018

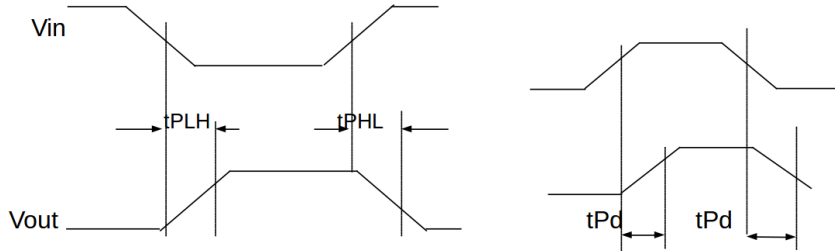
BITİŞ : 16 Kasım 2018

PROBLEM TANIMI



Bu projede, basit devre elemanlarından oluşan mantıksal bir devre kapı yayılım gecikmeleri de dikkate alınarak simule edilecektir. Entegreler (Integrated Circuit –IC); elektronik elemanların bir paket haline getirilmesidir. Sayısal entegrelerin veri sayfalarında (datasheet) bazı parametreler verilir: Lojik gerilim/akım seviyelerinin tanımlanması, Yükleme (çıkış) kapasitesi, Gürültü bağımsızlığı, Yayılım Gecikmesi ve Yayılım Hızı, Güç Harcaması, Hız-Güç Üretimi, Akım kaynağı ve Akım yutumu, Besleme gerilimi ve Çalışma Isısı gibi. Bunlardan yayılım gecikmesi (propagation delay), dijital devrenin veya lojik kapının girişindeki değişim ile buna bağlı olarak

çıkışta meydana gelecek değişme arasındaki zaman farkına karşılık gelir ve nanosaniye (ns) cinsinden ifade edilir. Şekil 1’de yayılma gecikmeleri temsil edilmiştir¹.

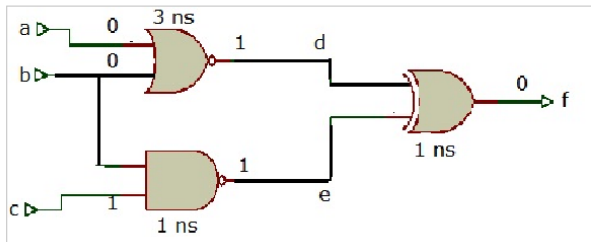


t_{pHL} → 1’den 0’a geçerken harcanan zaman, t_{pLH} → 0’dan 1’e geçerken harcanan zaman
 t_{pd} → iki değer (t_{pHL} ve t_{pLH}) ortalamasıdır.

Şekil 1: Yayılma gecikmesi gösterimi.

DEVRE TANIMLAMA FORMATI

```
.giris a b c #devrenin ana girisleri
.cikis f #devrenin ana cikisi
.kapi nor 2 d a b 3
.kapi nand 2 e b c 1
.kapi exor 2 f d e 1
.son
```



Şekil 2: Dosya içinde devre tanımlama formatı.

- Bütün devreler giriş/çıkış uçlarını bildiren .giris ve .cikis keyword (anahtar kelimeleri) ile başlamalı.
.giris <giris listesi>
.cikis <çıkış listesi>
- “devre.txt” dosyasında tanımlanan devre giriş/çıkış yapan bütün uçlar/keyword’ler/rakamlar arasında tab boşluk karakteri olmalı. Her bir

¹content.lms.sabis.sakarya.edu.tr/Uploads/69575/28659/bölüm_7.ppt

giriş/çıkış ucu ismi karakter ile başlamalı(büyük-küçük harf farketmez), içinde rakam ve underscore karakteri bulundurabilir. IN, d3, b_giris gibi.

- Bütün yorumlar diyez(#) karakteri ile başlar. Tek satırlık yorum vardır. Yorumlar program tarafından dikkate alınmamalı.
- Kapılar şu şekilde tanımlanır:
.kapi <kapı tipi> <giriş sayısı> <çıkış> <giriş listesi>
<kapının girişlere cevap verme süresi>
- Kullanılacak kapı tipleri: AND, OR, XOR, NOT, NAND, NOR
- Devre tanımlamaları **.son** anahtar kelimesi ile biter.
- Bir dosyanın içinde **.include <baska_dosya.txt>** yazılarak başka bir devre "baska_dosya.txt" dosyasından okunabilmelidir. Bu durumda ana devre tanımlamasının yapıldığı dosya içine başka bir tanımlama dahil edilmiş olur. Dikkat edilmesi gereken her iki dosyadaki giriş çıkış uçlarının aynı isimlendirilmesi (örneğin burada b girişi) aynı giriş veya çıkışa karşı geleceğidir.
- **Not:** Dosya tipi olarak düz text dosyası (.txt uzantılı) yerine YAML (YAML Ain't Markup Language)² dili gibi hiyerarşik bir dosya formatı da kullanılabilir. Dikkat edilmesi gereken husus, anahtar kelimeler ve dosya içindeki akışın .txt dosyalarında olduğu gibi kalacağıdır.

"devre.txt"

```
.include <baska_dosya.txt>
.giriş b c #devrenin ana girisleri
.cikis f #devrenin ana cikisi
.kapi nand 2 e b c 1
.kapi exor 2 f d e 1
.son
```

"baska_dosya .txt"

```
.giriş a b #devrenin ana girisleri
.cikis d #devrenin ana cikisi
.kapi nor 2 d a b 3
.son
```

Şekil 3: Dosya içinde başka devre tanımlama örneği.

²<https://en.wikipedia.org/wiki/YAML>

KULLANICI ARAYÜZÜ

- Mantık devresi benzetimi konsoldan çalışmalı. Kullanıcı programı ilk çalıştırdığında programınız "devre.txt" içindeki satırları yorumlayıp komut çalıştırabilir hale gelmelidir.
- Kullanıcı konsolda prompt(uyarı) karakterini '>' görmeli. Yani kullanıcıdan komut almaya hazır olduğunu gösteriyor.
- Kullanıcı aşağıdaki komutlardan birini yaptığında o komuta bağlı işlem icra edildikten sonra yeni komut için prompt(uyarı) karakterini '>' konsolda görmeli.(C-çıkış- komutu girilene kadar.)
- Komutların büyük harf veya küçük harf olması fark etmez.(not case sensitive)
- Bir takım komut dizisi "komut.txt" dosyası içinde alt alta yer alabilir.
- Devre yükleme (Y) ve ilklendirme (I) komutlarından sonra H, L, S, G, G*, K, C komutları aktif olmalıdır.

Tablo 1: Komut ve neyi icra edecekleri.

Komut	İcrası
Y <devre.txt>	"devre.txt" dosyasından devreyi yükler
I	"deger.txt" içindeki degerlerle devre ilklendirilir
H <giriş ucundan biri/birileri>	İlgili uç/uçları Lojik-1 yapar
L <giriş ucundan biri/birileri>	İlgili uç/uçları Lojik-0 yapar
S	Devreyi simüle eder
G <giriş/çıkış ucundan biri/birileri>	İlgili uç/uçların seviyesini (0 veya 1) konsolda gösterir
G*	Tüm uçların seviyesini (0 veya 1) konsolda gösterir
K <komut.txt>	"komut.txt" dosyası içindeki komutları konsoldan olduğu gibi icra eder
C	Benzetimden çıkış yapar

PROGRAM NASIL ÇALIŞIYOR?

Şekil 2'deki devre için "deger.txt" içindeki ilk değerler a=0, b=0, c=1, d=1, e=1, f=0 ise
>Y devre.txt
>I deger.txt
>h b komutlarından sonra arka planda aşağıdaki olaylar meydana gelir (Şekil 4'e bakınız).

- 0. nsn'de b-girişi 0'dan 1'e setlenmiş olur.
- G1(NOR) ve G2(NAND) kapıları bu olaydan etkilenir; çünkü b her iki kapı için de giriştir.
- G2 hesaplanır ve e çıkışı 1. nsn'de 1'den 0'a setlenir.
- G1'in yayılım gecikmesi G3(XOR)'ün yayılım gecikmesinden büyük olduğu için çıkışı 2. nsn'de f çıkışı 0'dan 1'e setlenir.

- G1 hesaplanır ve d çıkışı 3. nsn'de 1'den 0'a setlenir.
- G1 kapısı G3'ü tetiklediği için 4. nsn'de f çıkışı 1'den 0'a setlenir.

Zaman	Giriş/Çıkış	Değişim
0	b	0 → 1
1	e	1 → 0
2	f	0 → 1
3	d	1 → 0
4	f	1 → 0

Şekil 4: Olay akış sırası.

> S komutu ile yukarıdaki olaylar kullanıcıya gösterilmiş olur.

0ns: b 0->1

1ns: e 1->0

2ns: f 0->1

3ns: d 1->0

4ns: f 1->0

> G* komutu ile tüm girdi ve çıktıların son durumları gösterilir.

a: 0

b: 1

c: 1

d: 0

e: 0

f: 0

PUANLAMA TABLOSU

Puanlama Tablo 2'de özetlenmiştir.

Tablo 2: Puanlama Kriterleri.

İster	Puan
devre.txt içindekilere göre komutları icra etme	40 puan
devre.txt içinde baska_dosya.txt varlığında komutları icra etme	60 puan
C komutu çalıştırılana kadar konsolda yazılan komutlar ve bunlara karşılık gelen tüm çıktıların log.txt dosyasına tarih-saat-dk-sn şeklinde kaydedilmesi	20 puan
Rapor (her şey tam ise)	20 puan

Not 1: Puanlama tablosunda ilk iki satırdan biri puan olarak alınabilir. Proje C veya C++ dilleri kullanılarak gerçekleştirilecektir.

Not 2: **Proje teslimi ile ilgili hususlar:**

- Proje sunum gününde rapor (hard copy) teslim edilmesi gerekmektedir.
- Rapor IEEE formatında (önceki yıllarda verilen formatta) akış diyagramı veya yalancı kod içeren, özet, giriş, yöntem, örnek sonuçlar ve kaynakça bölümünden oluşmalıdır.

- Dersin takibi projenin teslimi dahil edestek.kocaeli.edu.tr sistemi üzerinden yapılacaktır. edestek.kocaeli.edu.tr sitesinde belirtilen tarihten sonra getirilen projeler kabul edilmeyecektir.
- Proje ile ilgili sorular edestek.kocaeli.edu.tr sitesindeki forum üzerinden Arş. Gör. Süleyman Eken veya Arş. Gör. Furkan Göz'e sorulabilir.
- Demo tarihleri daha sonra duyurulacaktır.
- Demo sırasında algoritma, geliştirdiğiniz kodun çeşitli kısımlarının ne amaçla yazıldığı ve geliştirme ortamı hakkında sorular sorulabilir. Kullandığınız herhangi bir satır kodu açıklamamız istenebilir.
- Bu proje de daha önce oluşturmuş olduğunuz grup arkadaşınızla devam edeceksiniz (eğer grup yaptığınız iseniz).