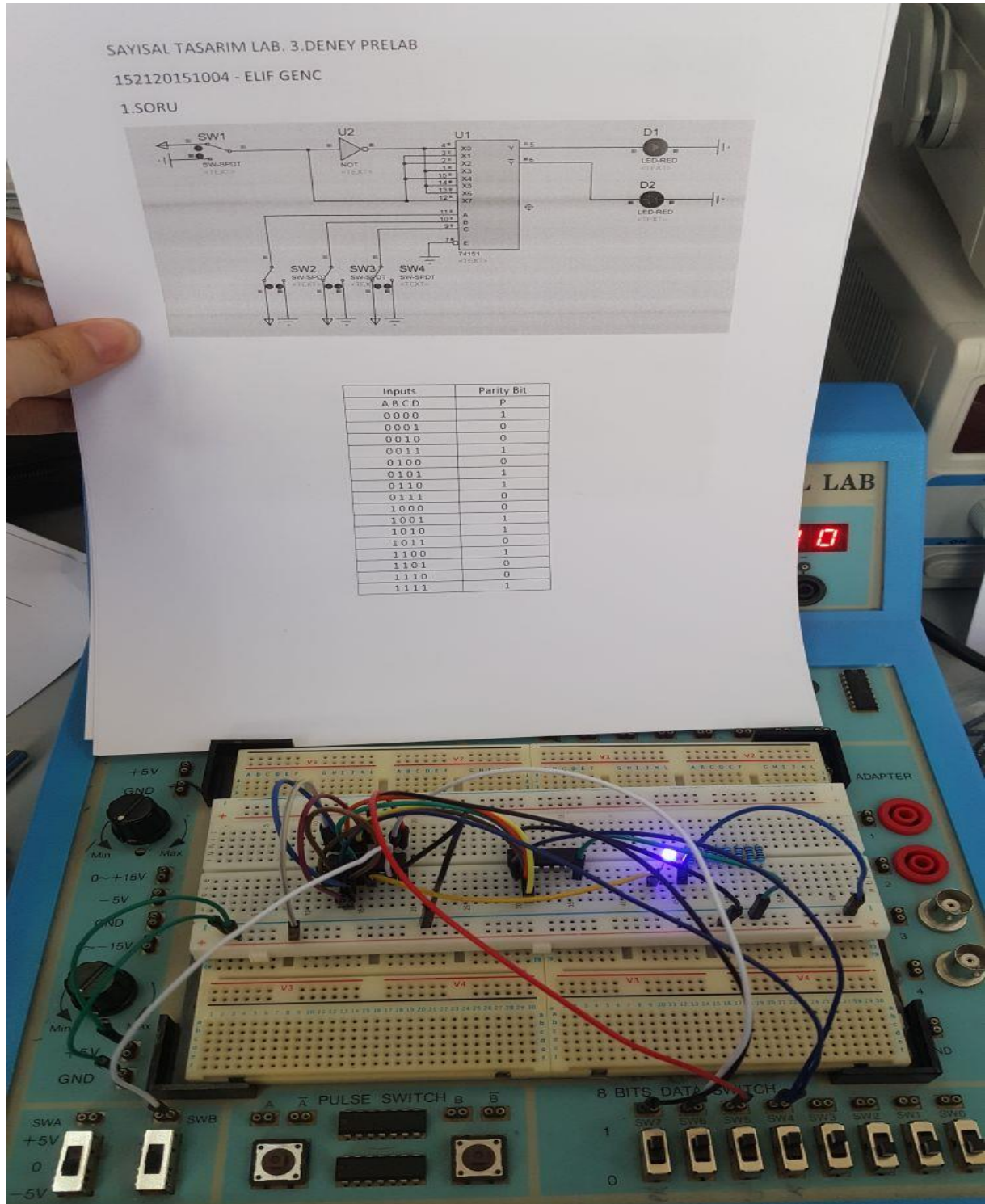


SAYISAL TASARIM LAB. - 3.DENEY RAPORU

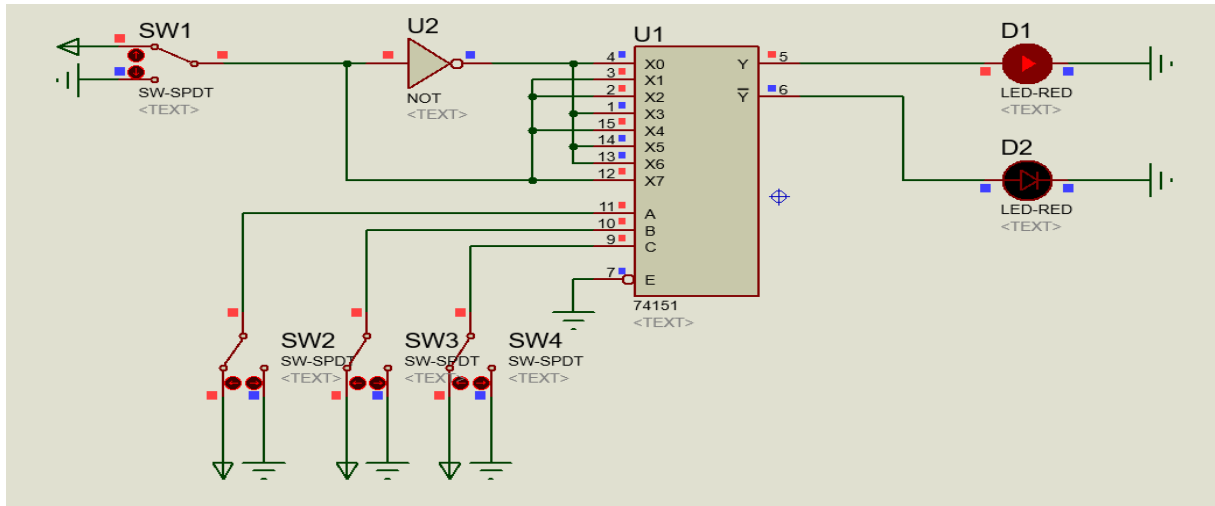
152120151004 - ELIF GENÇ

1.SORU

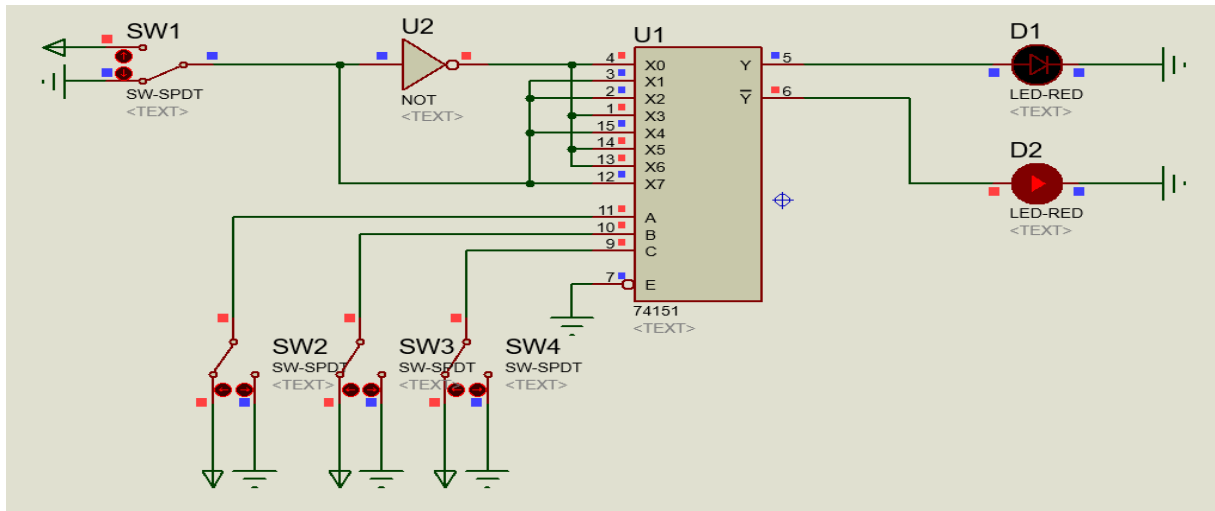


Parity Bit (Eşlik Biti): Parity seri sinyal iletişimlerinde alınan datanın doğru gönderilip gönderilmediğini anlamak için kullanılır. İkili tabandaki veriler içerisindeki bit değerlerinin toplamının tek veya çift olması esasına göre kontrol amaçlı olarak kullanılan bittir. İki çeşit parity vardır; ODD (tek) ve EVEN (çift).

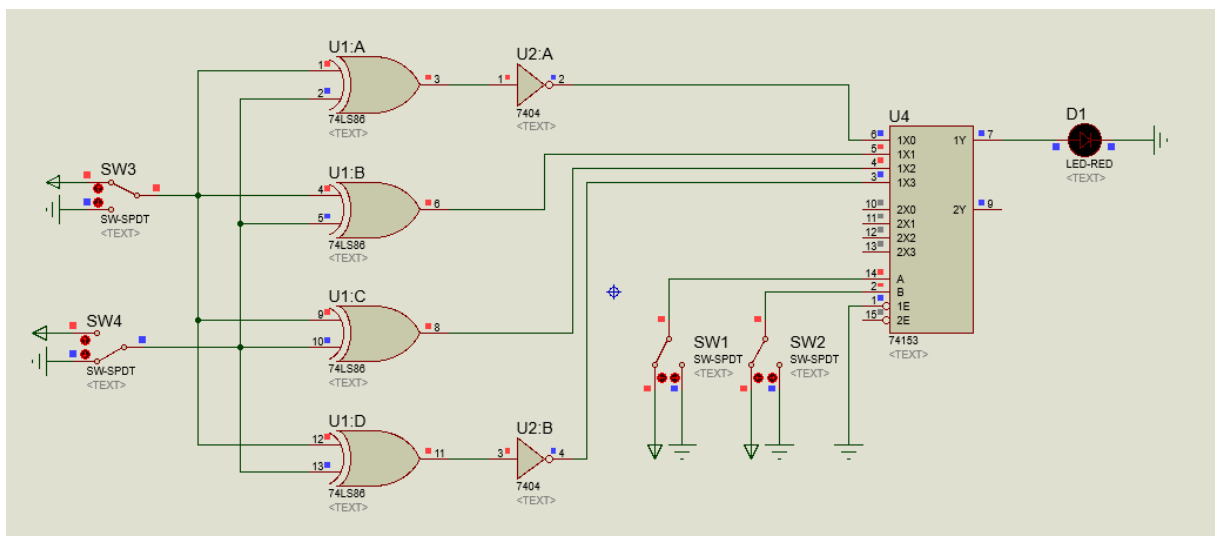
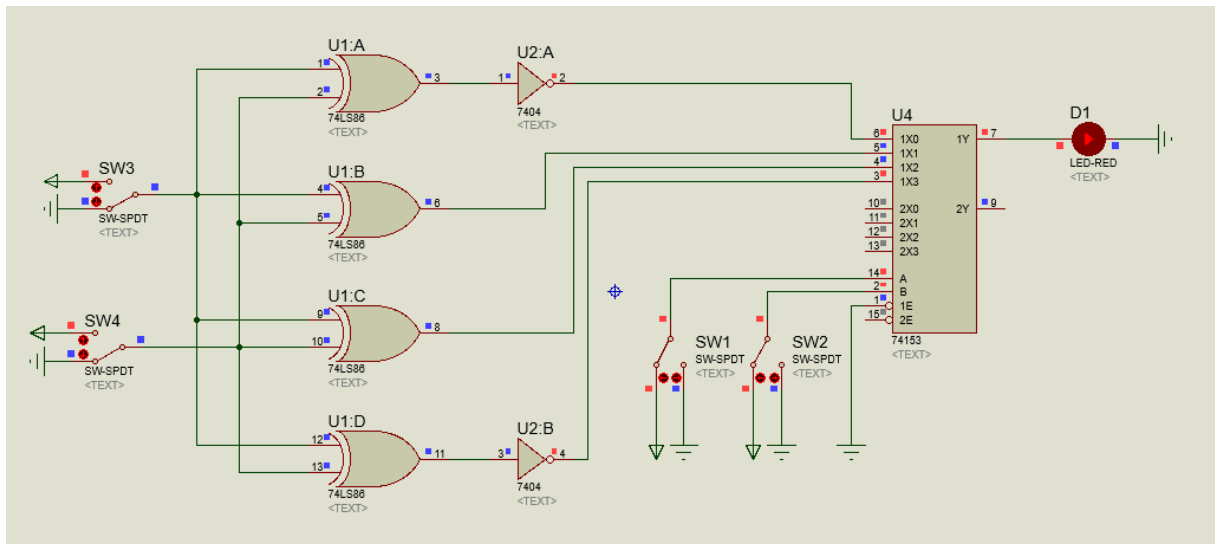
Biz bu devre tasarımı odd (tek) parity bit devresi tasarladık. 74151 nolu entegre (8 x1 multiplexer) kullanarak 0 ile 7 sayıları arasındaki data setlerinin odd parity olması için parity biti 0 yada 1 olarak led üzerinde gösteriyoruz. Verilen data setin odd (tek) parity bit olması gereken durumlarda D1 led' i yanıp, D2 led' i sönecektir. Eğer data set zaten odd (tek) parity bit özelliği taşıyorsa D1 led' i sönüp, D2 led' i yanacaktır. Ben bu koşulları sağlayabilmek için alttaki devreyi tasarladım ve doğruluğunu kontrol ettim.



Inputs	Parity Bit
A B C D	P
0 0 0 0	1
0 0 0 1	0
0 0 1 0	0
0 0 1 1	1
0 1 0 0	0
0 1 0 1	1
0 1 1 0	1
0 1 1 1	0
1 0 0 0	0
1 0 0 1	1
1 0 1 0	1
1 0 1 1	0
1 1 0 0	1
1 1 0 1	0
1 1 1 0	0
1 1 1 1	1



2.SORU



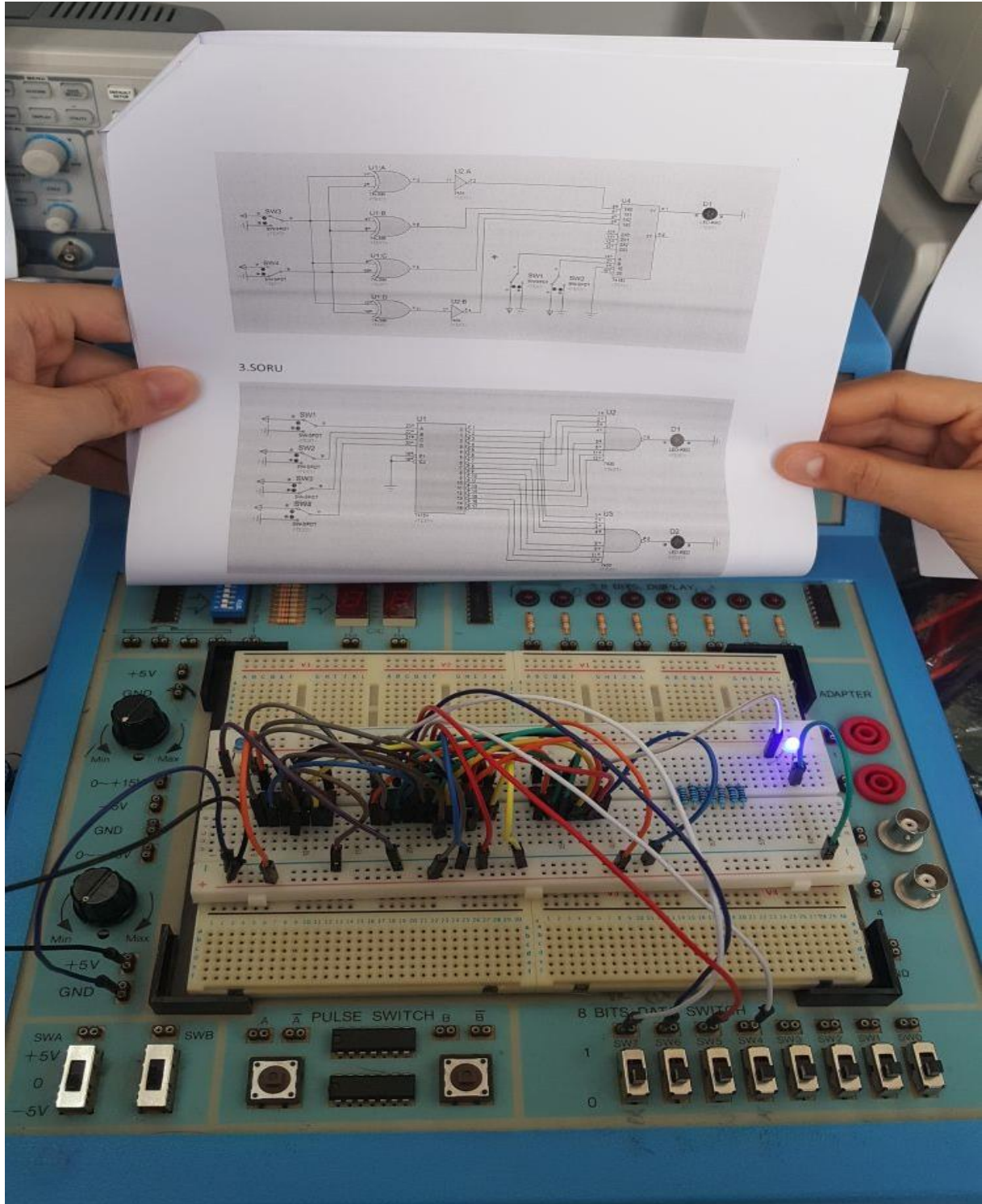
Bu devrede tasarlayacağımız konu birinci sorudaki konu ile aynıdır. Bu iki devre arasındaki temel farklılıklar entegremizin farklı olması ve bu entegreye bağlı olarak girişlerin değişmesidir.

Bu devre tasarımıda 74153 (dual 4x1 multiplexer) nolu entegreyi kullanıyoruz. Bu entegrenin seçim yapan bacakları (yani selection bits) 2 girişlidir ve entegreye gelen 4 bitlik veri soruda verilen C ve D girişlerinin kombinasyonu olarak tanımlanmıştır.

A	B	C	D		
0	0	0	0		1
0	0	0	1		0
0	0	1	0		0
0	0	1	1		1
0	1	0	0	0	
0	1	0	1	1	
0	1	1	0	1	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	1	
1	0	1	0	1	
1	0	1	1	0	
1	1	0	0		1
1	1	0	1		0
1	1	1	0		0
1	1	1	1		1

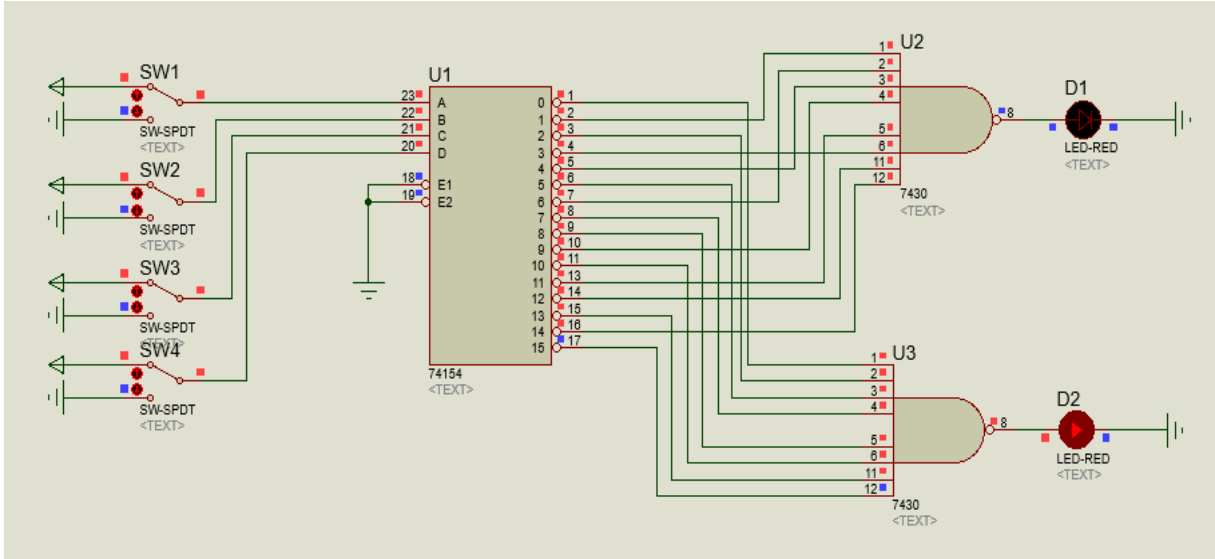
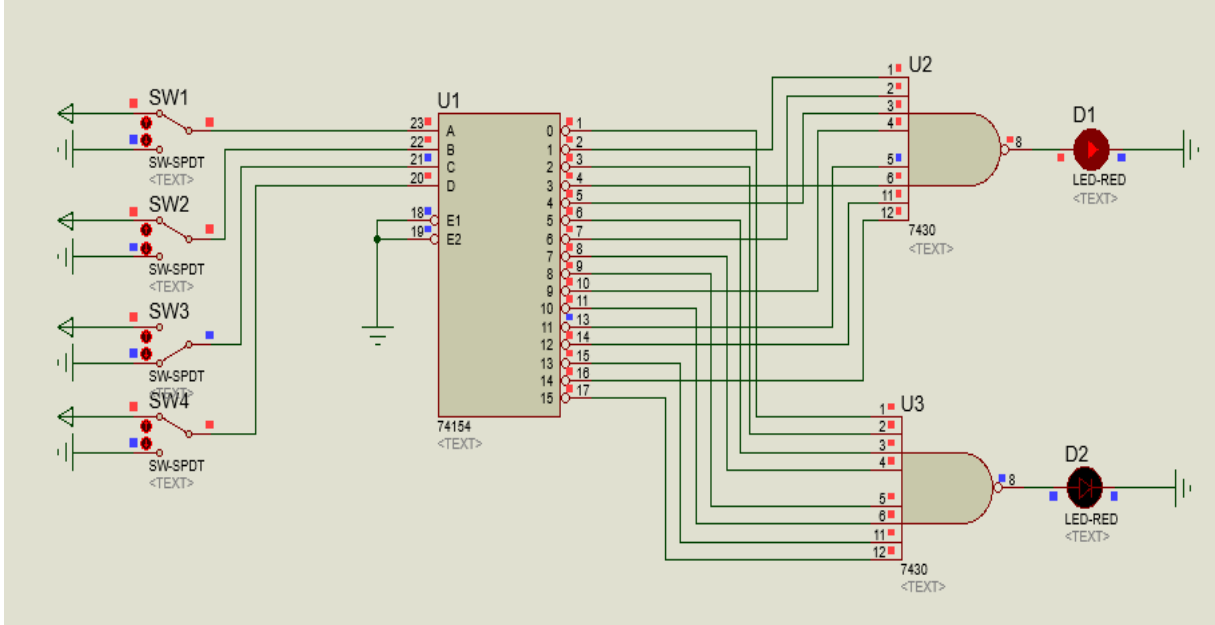
İlk 4 ve son 4 satır aynı verileri temsil ederken 4 ile 12 satırları arasındaki verilerde 4' erli olacak şekilde birbirini tekrar ediyor. Bu bilgiyi tablo yardımıyla gördükten sonra ilk 4 ve son 4 satırlar EXOR + NOT kapılarının ve 4-8 ve 8-12 satırlar arasındaki veriler ise sadece EXOR kapısının doğruluk tablosunun verileri ile aynıdır. Bu yüzden elimizdeki C ve D girişlerini EXOR ve NOT kapılarını kullanarak 4 giriş olarak çokluyoruz. Bu devrenin bize verdiği sonuç; eğer gelen veri de 1' lerin sayısı tek ise led yanmayacak ,değilse led yanıp bize sonuç hakkında bilgi verecektir.

3.SORU

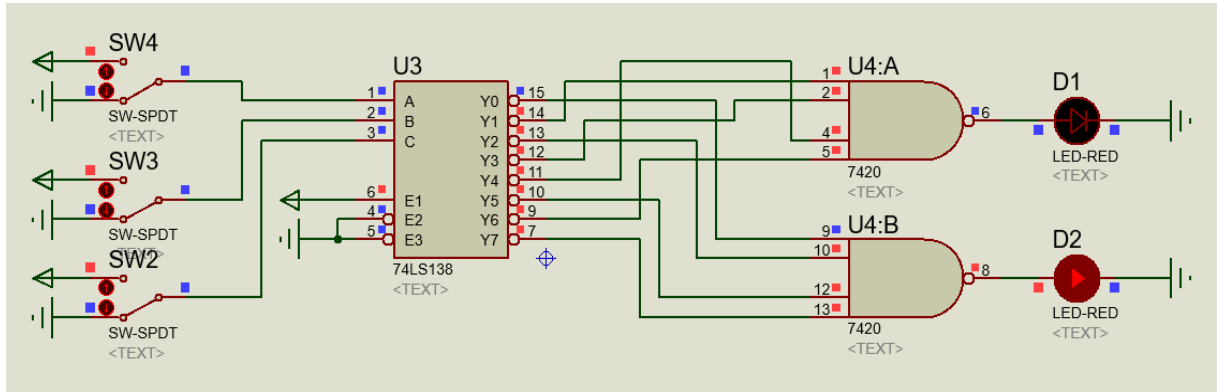
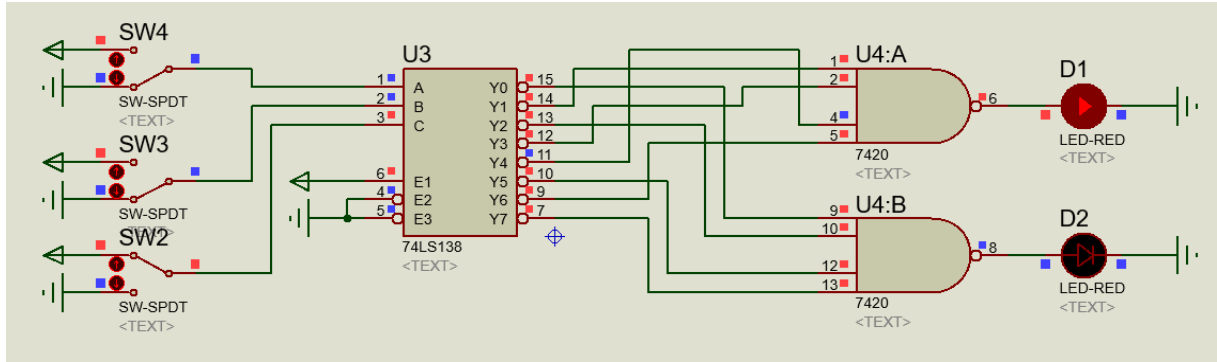


Elimizde 2 tane input vardır ve bu inputların her biri 2 bitlik veriyi temsil eder. Yani elimizde bulunan 2 bitlik 2 inputu devrede tasarlarken 74154 nolu entegrede (4x16 decoder) 4 farklı giriş olarak tasarlayacağız. Bu iki inputun toplamı tek ise D1 led' i yanıp D2 led' i yanmayacaktır ama çift ise D1 led' i sönüp D2 led' i yanacaktır.

Devreyi tasarlarken sonucu tek olan verileri 8 giriřli NAND kapısına, sonucu çift olanları ise diğeri NAND kapısının giriřleri olarak bađladık. Bu da bize dođru sonuçlar elde etmemizi sađladı.



4.SORU



A	B	C	D	Tek	Çift
0	0	0	0		Ç
0	0	0	1	T	
0	0	1	0		Ç
0	0	1	1	T	
0	1	0	0	T	
0	1	0	1		Ç
0	1	1	0	T	
0	1	1	1		Ç

1	0	0	0		Ç
1	0	0	1	T	
1	0	1	0		Ç
1	0	1	1	T	
1	1	0	0	T	
1	1	0	1		Ç
1	1	1	0	T	
1	1	1	1		Ç

Bu devre tasarımında 3 giriş kullanmamız isteniyor. Tabloda görüldüğü gibi A girişinin 0 yada 1 olması devreden çıkan sonucu etkilemiyor. O yüzden bende entegreye giren girişlerin birincisini A' nin 2 biti, diğer 3 girişi ise B inputunun 1 ve 2. biti olarak ayarladım. Bu iki tabloda da tek ve çiftlerin sıralaması değişmiyor. Bu yüzden

entegreden çıkması gereken tekleri bir NAND kapısına, çiftleri ise diğer NAND kapısına giriş olarak çizdim. Sonuçları proteus programında kontrol ettim ve sonuçlar istenilen gibi çıktı.