

Deney 2: Kod Dönüştürücü ve Kodlayıcı Tasarımı

Genel Bilgiler:

Bu deneyde, SSI (**S**mall **S**cale **I**ntegration: Küçük Ölçekte Tümlleştirme, 1-9 kapı) devreler kullanarak bir BCD den 3 fazla (Excess-3) kod dönüştürücüsü ve 8'den 3'e bir kodlayıcısının sentezi yapılacaktır.

Lojik devre sentezi yöntemleri genel olarak iki grupta toplanabilir. Birincisi sözle tanımdan doğruluk tablosuna geçmek ve bu tablodan da Quine-McCluskey yöntemi ile yada Karnaugh diyagramı uygulamasıyla minimal fonksiyonu bulmaktır. Minimal fonksiyona karşı düşen devre iki seviyeli gerçekleştirilebilir. Birinci yöntemin iki farklı uygulamasından hangisinin seçileceği bağımsız değişken sayısı ile saptanır. Değişken sayısı 4-5'e kadar olan fonksiyonlarda Karnaugh diyagramı uygulaması daha çabuk sonuca ulaştıracağından tercih edilir. İkinci yöntem ise sözle tanımdan bir algoritma çıkararak bu algoritmaya karşı düşen devreyi gerçekleştirmektir. Bu yöntem genellikle değişken sayısı fazla olan ve/veya keyfi çıkışları fazla olan fonksiyonların gerçekleştirilmesinde daha yararlıdır. Örneğin ders notlarında incelenen, karşılaştırma ve 4 bitlik toplama devreleri bu yöntemle gerçekleştirilmiştir. Birinci yöntemin uygulanmasındaki güçlük değişken sayısı ile doğruluk tablosunun üstel biçimde büyümesidir. İkinci yöntemde bu güçlük görülmez.

Bu iki genel yöntem dışında, bir de doğruluk tablosuna karşı düşen Boole fonksiyonunda lojik işlemler yapılarak bulunan eşdeğer fonksiyonların gerçekleştirilmesi yöntemi vardır. Ancak genel kurallar verilememiştir. O nedenle lojik işlemler sezgisel olarak yapılmaktadır. Diğer bir yöntem, Özel-VEYA kapıları kullanarak yapılan tasarım yönteminin genel kuralları verilmiştir ancak bu dersin kapsamı içine alınmamıştır.

Bu deneyde ilk iki yönteme ilişkin birer örnek devre incelenecektir. Birincisi BCD 'den (Binary Coded Decimal) 3 fazla koduna dönüştürücünün tasarımıdır. İkincisi ise 8 girişli 3 çıkışlı kodlayıcı tasarımıdır.

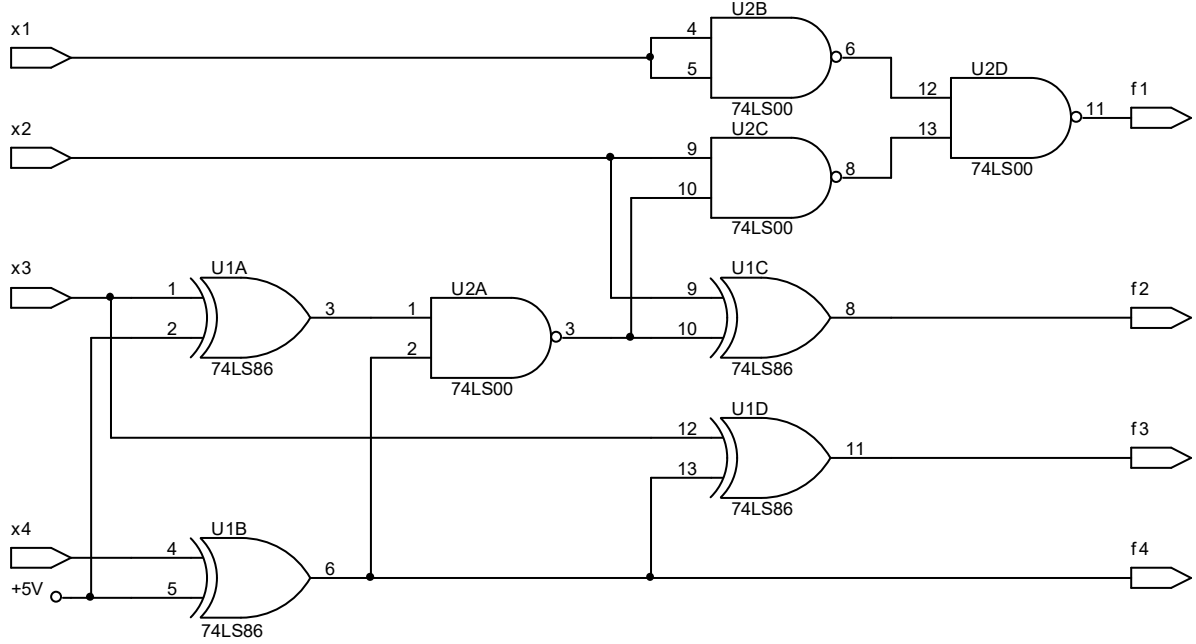
BCD-3 fazla Kod Dönüştürücü

Sözle Tanım: Bağımsız girişler BCD kodları temsil eden ($x_1x_2x_3x_4$) değişkenleridir. En ağırlıklı bit (MSB: **M**ost **S**ignificant **B**it), x_1 dir. Çıkışlar ise 3 fazla kodunu temsil eden ($f_1f_2f_3f_4$) bağımlı değişkenlerdir. MSB f_1 'dir. Bu devrenin çıkışında girişine uygulanan sayının üç fazlası görülür. Çıkış fonksiyonlarına ilişkin doğruluk tablosu aşağıda verilmiştir.

Tablo 2-1 BCD-3 fazla kod dönüştürücü doğruluk tablosu

x_1	x_2	x_3	x_4	f_1	f_2	f_3	f_4
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
10-15				k	k	k	k

$f_1f_2f_3f_4$ ün minimal fonksiyonları bulunduğundan ve lojik işlemler yapıldıktan sonra bulunan fonksiyonlara karşı düşen devre aşağıda Şekil 1-1.1 de verilmiştir.



Şekil 2.1 BCD-3 fazla kod dönüştürücü devre

8 Girişli 3 Çıkışlı Kodlayıcı

Sözle tanım: Üç bitlik kodlayıcının (encoder), $2^3=8$ bağımsız girişi ($g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7$) ve üç çıkışı vardır. Giriş değişkenlerinden bir tanesi 1, diğerleri 0 iken; 1 değerini alan giriş değişkeninin indisi çıkışta ikili kodlanmış olarak görülür. Örneğin $g_3=1$, diğer değişkenler 0 iken çıkış 011'dir. Bu sözünü ettiğimiz girişlerin sayısı giriş değişkeni sayısı kadar, yani 8 tanedir. Oysa toplam $2^8=256$ tane giriş vardır. Söz konusu girişlerin dışında 248 giriş hiçbir zaman uygulanmamaktadır, yani bu girişler için çıkışlar keyfidir.

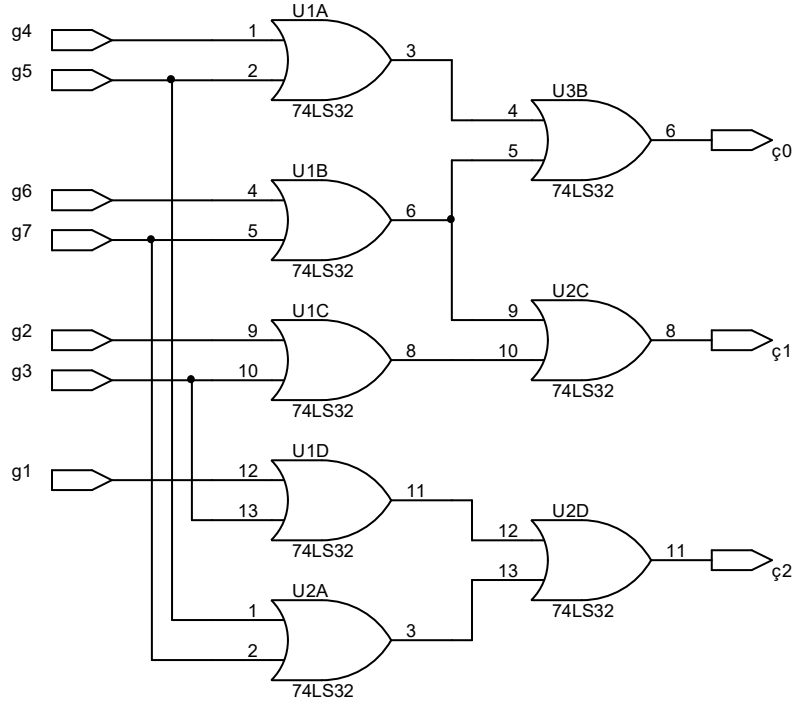
Algoritmayı bulmak için ipucu olarak; çıkış fonksiyonlarının 1 değerini alması, hangi girişlerin 1 olmasını zorunlu kılmaktadır diye düşününüz. Bu girişlerin VEYA edilmesi istenen fonksiyonu verir.

İnceleyeceğimiz devrenin doğruluk tablosu Tablo 2-2'de verilmiştir.

Tablo 2-2 3-Bit kodlayıcının doğruluk tablosu

g_0	g_1	g_2	g_3	g_4	g_5	g_6	g_7	ζ_0	ζ_1	ζ_2
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Sonuç devre Şekil 2.2'de verilmiştir.



Şekil 2.2 Bir 3-Bit kodlayıcının lojik devresi

Deney Öncesi Yapılacak İşlemler:

- 1- Genel Bilgiler kısmını ve ders notlarından ilgili kısımları çalışınız.
- 2- Kullanılan tümleşik devrelerin katalog bilgilerini inceleyiniz.
- 3- Deneyin sonunda yer alan soruları yanıtlamaya çalışınız.

Deneyde Yapılacak İşlemler:

1-

- a) Şekil 2.1'deki devreyi deney düzeneğiniz üzerine kurunuz.
 - Tümleşik devrelerin besleme ve toprak bağlantılarını yapınız.
 - Üç değişkenin tümleyenini ilave bir DEĞİL tümleşik devresi kullanmadan elde edebiliriz. ÖZEL-VEYA kapısının bir girişi 1, diğer girişi tümleyeni alınacak değişkene bağlanırsa, çıkışında girişteki değişkenin tümleyenini görür. Nedenini düşününüz.
 - ($x_1x_2x_3x_4$) girişlerine, istediğiniz değerleri verebilmek için, birer lojik anahtara bağlayınız.
 - Lojik anahtarın yanındaki +V ve 5V seçeneğini sağlayan anahtarı 5V konumuna getiriniz. Bizim deneylerimizde TTL tümleşik devreleri kullanıldığından bu seçim yapılmıştır. CMOS tümleşik devreler kullanılsaydı +V seçeneği seçilirdi.
 - $f_1f_2f_3f_4$ çıkışlarını gözleyebilmek için LED'lere (Light Emitting Diode) bağlayınız. Bu ışık veren diyotlardan kırmızısı yandığında çıkışın lojik 1 olduğunu, yeşili yandığında ise çıkışın lojik 0 olduğunu gösterir.
 - LED'lerin bulunduğu yerde siyah renkli iki anahtar vardır. (TTL-CMOS) seçenekli anahtarı TTL konumuna, (+V-5V) seçenekli anahtarı ise 5V konumuna getiriniz.
- b) Kurduğunuz devrenin dönüştürme işlemini yaptığını sağlatınız.

2-

- a) Şekil 2.2 deki devreyi deney düzeneğiniz üzerinde kurunuz.
- Tümlleşik devrelerin besleme ve toprak bağlantılarını yapmayı unutmayınız.
 - Dört girişli VEYA kapıları yerine iki girişli VEYA kapıları kullanılmıştır.
 - ($G_0, G_1, G_2, G_3, G_4, G_5, G_6, G_7$) girişlerini, istediğiniz değerleri verebilmek için lojik anahtarlara bağlayınız.
 - Devrenizin üç çıkışını gözleyebilmek için LED'lere bağlayınız.
- b) Kurduğunuz devrenin, doğruluk tablosunu oluşturarak, üçlü kodlama işlemini yaptığını sađlatınız.

Sorular:

- 1- Kod dönüştürücü için bulunan minimal fonksiyonlardan, gerçeklediğiniz eşdeğer fonksiyonlara nasıl geçildiğini gösteriniz.
- 2- Kod dönüştürücü devreyi, doğrudan minimal fonksiyonları yalnızca VEDEĞİL kapıları kullanarak gerçekleyiniz. Deneydeki devre ile bulduğunuz bu devreyi eleman sayısı, tüm devre sayısı, giriş yelpazeleri ve gecikmeler açısından karşılaştırınız.
- 3- Kod dönüştürücü devreye ilişkin fonksiyonlarda görülen keyfiler (k) bir yarar sağladı mı? Neden?
- 4- Deneyde gerçekleştirdiğiniz kod dönüştürücü devrede farklı fonksiyonlar için ortak kullandığınız kapılar oldu mu? Bu sonucu çok çıkışlı devrelerdeki bir genel kurala dönüştürebilir misiniz?
- 5- Kod dönüştürücünün yukarıda verilen doğruluk tablosu ile sizin bulduğunuz doğruluk tablosunu karşılaştırınız. Farklılıkların nedenini açıklayınız.
- 6- Kodlayıcıda 4 girişli VEYA yerine 2 girişli VEYA kullanabileceğimiz Boole cebirinde toplama işleminin hangi özelliğine dayanmaktadır. VEDEĞİL işlemi için bunu kullanabilir miyiz?
- 7- Kodlayıcıda 4 girişli VEYA yerine 2 girişli VEYA kullanmak bir zorunluluk çünkü giriş yelpazesi 4 olan VEYA kapısı katalogda yoktur. Oysa 4 girişli VEDEĞİL kapısı katalogda var. Buna göre Devreyi VEDEĞİL kapılarıyla gerçekleseydiniz tümlleşik devre sayısı bakımından bir yarar sağlar mıydınız? Neden?
- 8- Kodlayıcı devrede bir kapıyı farklı iki fonksiyon için ortak kullanmanızın bir yararı oldu mu? Neden?
- 9- Kodlayıcıda hiçbir indirgeme düşünmeden devreyi gerçekleştireseydiniz kaybınız ne olurdu?
- 10- Kodlayıcı devrenizde özel 8 girişin dışında 248 giriş için keyfi değer alan çıkışlar hangi değerleri alırlar? Neden?
- 11- En genel anlamda bir BCD Excess kodlayıcıyı öğrenci numaranızın son rakamına göre tasarlayın. (Numarası 0 ile bitenler 2, 3 ile bitenler 4 alacaktır.)
- 12- 4-bit Gray koddan 4-bit binary sayıya dönüştüren devreyi Özel-VEYA kapılarıyla tasarlayınız.

Malzeme Listesi:

- 1 adet 74LS86 Özel-VEYA tümlleşik devresi
- 1 adet 74LS00 VEDEĞİL tümlleşik devresi
- 2 adet 74LS32 VEYA tümlleşik devresi