

DENEY 4

SAYISAL ARİTMETİK

Deneyin Amacı

Bu deneyde işaretli ve işaretsiz sayılar için ikili sayı (Binary) sistemindeki toplama işleminin anlaşılması, işlem performansını artırabilmek için iki tabanındaki sayıların ikili tamlayan (complement) biçiminin anlaşılması, ikili sistemdeki sayıların çarpılıp bölünmesi işlemlerinin anlaşılması hedeflenmiştir.

Ayrıca 16 (Hexa decimal) tabanlı sayı sistemi ve BCD kodundaki sayılar ile aritmetik işlemlerin yapılması üzerinde durulacaktır.

Önbilgi

N bitlik bir sayı $0 - 2^N - 1$ arasındaki sayıları temsil edebilir. Bu şekildeki ifade tarzı işaretsiz tamsayı veya kesirler için geçerlidir. Ancak sayısal sistemlerde işaretli sayılarında kullanılması zorunludur. Bu yüzden işaretli sayılarında ifade edilmesi gerekmektedir. İşaretli sayılar iki şekilde ifade edilmektedir (Kodlanmaktadır).

1-İşaret biti, mutlak değer şeklinde kodlama :

Bu şekilde gösterimde sayının mutlak değeri ikili sistemde ifade edilir ve bu sözcüğe (en ağırlıklı bit olarak) işaret biti eklenir. Sayı pozitif ise işaret biti 0, sayı negatif ise işaret biti 1 olur. Örnek ;

01010111 (+87)

11010111(-87)

Bu örneklerdeki 8 bitlik sözcükler sayının mutlak değerini ve işaretini belirtir.

Toplama ve çıkarma işlemlerinde, işaret bitinde işleme katmak için bazı kurallara ihtiyaç vardır. Bu kurallar aşağıda özetlenir.

- Aynı işaretli olanlar aynen toplanır işaret biti değiştirilmez.
- İşaret biti farklı olanlar ise işaret bitine bakılmaksızın büyük olandan küçük olan çıkarılır. Büyük olanın işareti yazılır. Aşağıdaki örnekleri inceleyiniz.
-

001011	+ 01011	001011	+01011
+ 001110	+ 01110	+ 100110	- 00110
	+ 11001 (011001)		+00101 (000101)

001011	+01011	001011	+ 01011
+110110	-10110	- 001001	- 01001
	-01011 (01011)		+00010 (000010)

Örnekler iyi incelenirse bu şekilde kodlanmış sayıları çıkartmak için çıkartma devrelerine gerek olduğu görülür (Bu konunun öğrenci tarafından araştırılması gerekir.)

2- Sayının tümleyen metoduna göre kodlanması:

İkilik sayıların kodlanmasının özel bir şeklide ikili sayı sistemleri için kullanılan tamamlayıcı (tümleyici) gösterimidir. Bu gösterim en yaygın kullanım haline gelmiştir. Bunun sebebi 2’li sayı sisteminin tamamlayıcı formu sayısal makinelerde kolayca gösterilebilir ve işlem yapılabilir. Tümleyen aritmetikte sayının negatifliği sayının kendi içinde tutulur. Bundan dolayı, sonuçta işaret kavramından vazgeçilir. Bunun anlamı tümleyen metoduna göre yazılmış sayılarda çıkartma işleminin de toplayıcılar ile yapılmasıdır. Tümleme işlemi 2 şekilde gerçekleştirilir. **Taban’a tümleme** (2’ye tamamlama), **Taban –1’e** (1’e tamamlama) **tümleme**.

Sayı	01010101	11011011
1’e tümleyen	00101010	10100100
2’e tümleyen	00101011	10100101

Soldaki sayı +85’dir. Sağdaki sayı ise –91’dir. En soldaki bit yani işaret biti tamamlama işleminde değişmez. 1’li tamamlama işaret biti değişmeden diğer bitlerin tamlayanının (1’se 0, 0’sa 1) alınmasıdır. Bunun anlamı şudur; tüm büyüklük bitleri için 1’li tamlamaya dönüşüm donanımında kolayca sağlanır. 2’li tamlama ise 1’li tamlayanla 1’in toplanmasıdır. Bu EXOR kapısı ve bazı toplayıcı devrelerle gerçekleştirilir. İkili sayı sisteminde çıkarma işlemi için çıkan sayının ikili tümleyen ile çıkarılan sayı toplanır. Aşağıda buna bir örnek verilmiştir.

$$\begin{array}{r}
 14 \\
 - 5 \\
 \hline
 \end{array}
 \qquad
 \begin{array}{r}
 00001110 \\
 + 11111011 \\
 \hline
 100001001
 \end{array}$$

Dikkat edilirse çıkarmanın sonucunun bir taşma veya taşma bitine sahip olduğu görülür. Sonuç aynı zamanda negatif sayıların ikili tamlayan gösterimi ile işaretli bir büyüklük şeklindedir. Bu örnekte taşma biti kullanılmamaktadır bu yüzden atılmaktadır (Tamamlayıcı toplama yoluyla çıkartma işlemlerine göz gezdirerek deneye gelinmelidir.)

İkili sayı sisteminde çarpma ise on tabanlı sayı sistemindeki kurallara benzer olarak yapılmaktadır. İkili tabanda çarpma için kurallar aşağıdaki gibidir:

- ikili tabandaki bir sayı 1 ile çarpılırsa sonuç sayının kendisidir.
- ikili tabandaki bir sayı 0 ile çarpılırsa sonuç 0’dır.

$$\begin{array}{r}
 \text{Bir örnek} \quad 1101 \text{ çarpılan} \\
 \quad \times 101 \text{ çarpan} \\
 \hline
 \quad 1101 \text{ kısmi sonuçlar} \\
 \quad 0000 \\
 + 1101 \\
 \hline
 1000001 \text{ gerçek sonuç}
 \end{array}$$

Bu metodun kullanımı kolaydır ve doğru sonucu verir. İkili tabanda sayıların çarpımı için tek yol bu değildir.

Başka bir yöntemde aynı ara sonuçlar bir önceki yöntemdeki gibi yapılır. Ve soldan sağa doğru kaydırılarak diğer yöntemdeki gibi sonuçlar başarılı bir şekilde elde edilir. Bu metoda bir örnek aşağıda verilmiştir.

$$\begin{array}{r}
 1101 \text{ çarpılan} \\
 \times 101 \text{ çarpan} \\
 \hline
 1101 \text{ kısmi sonuçlar} \\
 00000
 \end{array}$$

+ 001101

1000001 toplam sonuç

bu yöntemde diğer yöntem gibi doğru sonuç verir. Bu metod diğer metoda göre bir tane daha az register'e ihtiyaç duyar ve donanımını gerçekleştirmek biraz daha kolaydır.

İkilik sistemde, bir sayıyı 2 ve 2'nin kuvvetleri ile çarpmanın, o sayıyı birkaç kez sola ötelemekten ibaret olduğunu da hatırlayınız. Sayısal sistemde iki sayının çarpılması işlemi oldukça zaman alan bir işlemdir. Bunun için değişik çarpma algoritmaları mevcuttur ve bu konu araştırılmaya devam edilmektedir.

İkili tabanda bölme ise; ikili tabandaki çıkarmanın başarılı bir şekilde yapılmasıdır. Bu şekilde bölme yapmak doğru sonucu verir fakat özellikle büyük sayılarda yavaş ve biçimsizdir.

1000001/1101 bölümünün sonucu bu yöntemle bulunmak istenirse;
1000001- 1101-1101-1101-110-1101=0

(001)(010) (011) (100) (101) O zaman bu bölmenin sonucu $(101)_2$ 'dir.

Ayrıca ikilik sitemde kodlanmış bir sayının 2 ve 2'nin kuvvetlerine bölünmesi işleminin o sayının sağa doğru kaydırılması ile yapılabileceğini de hatırlayınız.

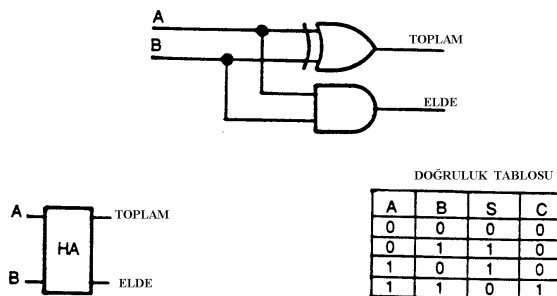
Hexadecimal Aritmetik: İkili taban aritmetiğinde özellikle büyük sayılarla işlem yapıldığında bunu 16'lı sisteme dönüştürerek işlem yapmak ikili taban aritmetiğine göre çok daha iyi bir performans sağlar. BCD kodlama, toplama sonuçların displayde 10'lu sisteme göre gösterilmesi gereken sistemlerde sıkça kullanılır. Hesap makineleri buna güzel bir örnektir. BCD sistemlerinde hexadecimaldeki 6 sayının kullanılmamasından dolayı bazı ilginç olaylar meydana gelir. Aşağıda bunla ilgili bazı örnekler verilmiştir.

DECİMAL	BCD	DECİMAL	BCD	DECİMAL	BCD
3	0011	6	0110	8	1000
+ 4	+ 1011	+ 8	+1000	+ 9	+ 1001
7	0111 Doğru	14	1110 Yanlış	17	00010001 Yanlış

Bu problemin meydana gelme sebebi 10'lu tabanda 9'dan büyük toplamalar için elde biti meydana gelir fakat BCD'de 15den büyük toplamalar için elde biti oluşur. Bunu düzeltmek için 9'dan büyük tüm toplamalara 6 daha eklenir. Bu işlemleri yapmak için toplamalardan hangisinin 9'dan büyük olduğunu bulan ve bulduklarına 6 ekleyen bir ek sisteme ihtiyaç vardır. Toplamı 9'dan az olanlar BCD aritmetiğinde doğru sonucu verirler. (Bu konunun öğrenci tarafından gözden geçirilmesi tavsiye olunur.) Oysa Hexadecimal sayılar, ikilik sistemde tam kodlama şeklinde kodlandığından toplama işlemi her zaman daha basit ve doğrudur.

Yarım Toplayıcı

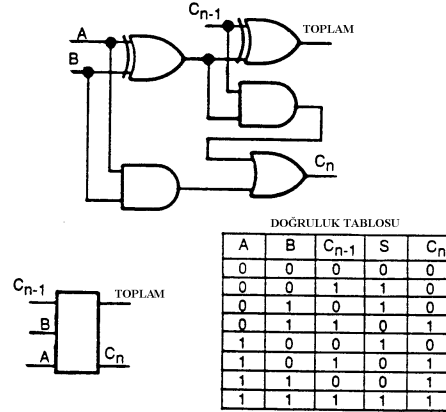
Yukarıda özet olarak anlatılan algoritmaları gerçekleştiren donanımın çekirdeği yarım toplayıcılarıdır. Yarım toplayıcının şeması ve doğruluk tablosu aşağıda gösterilmiştir. Şemadan da anlaşılacağı üzere yarım toplayıcı , birer bitlik iki sayıyı toplayan iki girişli iki çıkışlı bir devredir. Çıkışlardan birisi sonuç diğeri ise elde biti çıkışıdır.



Dikkat edilirse devrenin iki girişi ve iki çıkışı vardır. EXOR toplamayı gerçekleştirir. AND ise iki giriş birden 1 olduğu zaman elde bitini oluşturur. Buna yarım toplayıcı denmiştir çünkü bir önceki basamaktan gelen elde bitini işleme alamamaktadır. Yani en sağdaki basamakta toplama işlemi yapabilirler.

Tam toplayıcı:

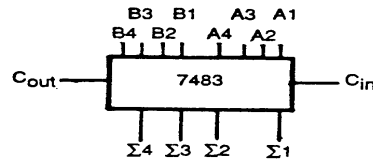
Tam toplayıcı üç girişe ve bir çıkışa sahiptir. Girişlerden ikisi toplanacak olan iki bit, diğer giriş ise bir önceki toplamdan gelen elde bitidir. Tam toplayıcının çıkışları ise toplam ve elde bitidir. Tam toplayıcının şeması ve doğruluk tablosu aşağıda verilmiştir. Tam toplayıcı her basamakta toplama işlemini başarabilir.



Paralel Toplayıcılar:

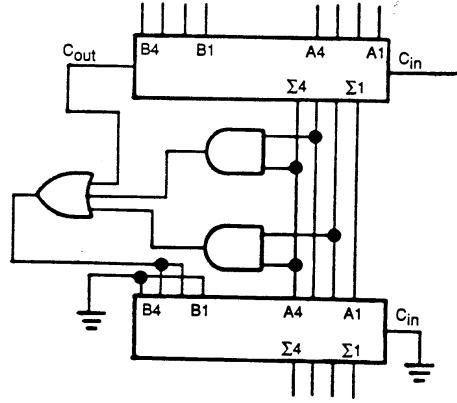
Bu tip toplayıcılar çok bitli ikili tabandaki sayıların toplanmasında iyi performans sağlarlar. Bu fonksiyonu gerçekleştiren TTL lojik ailesinden olan 7483 entegresidir. Bu devrenin tartışılması gereken bazı özellikleri vardır.

7483 entegresi hızlı elde bitini elde eden 4 bitlik binary toplayıcıdır. Elde bitinin hızlı bir şekilde elde edilmesi look ahead (ileriye gören) denilen elde devresi tarafından mümkün kılınır. Bu devre tüm tek toplayıcıların çıkışlarından örnek alarak bunları kaydederken tüm toplayıcıların eldesini gösteren ripple için bir zaman gerektirir. 7483 aynı zamanda doğru lojikselsel duyarlılıkta üstün bir matematiksel performans sağlar. Bunun anlamı şudur tüm çıkışlar doğru olacaktır. 1'e tümlleme aritmetiği için bunun anlamı şudur; son çevrim eldesi direkt işleme girer. İki farklı işaretli sayının toplamı pozitifse bu son çevrim elde bitine ihtiyaç vardır. 7483'ün blok diyagramı aşağıda gösterilmiştir.



BCD Toplayıcı :

BCD toplayıcı iki tane dört bitlik toplayıcı ve bazı ek devrelerden oluşur. BCD toplayıcının şeması aşağıda gösterilmiştir. Dikkat edilirse ilk toplayıcının basitce toplama yaptığını ikinci toplayıcının ise toplamı 9 veya daha büyük olanlara 6 eklediği görülür. İkinci toplayıcı, birinci toplayıcının çıkışın 9 veya daha büyük olması durumunda AND ve OR bağlaçları tarafından kontrol edilir.



Binary Çarpıcılar

TTL lojik familyasında mantıklı değişik entegre çarpıcılar vardır. Bu çarpıcılar hızın önemli olduğu uygulamalarda kullanılır. Bu tip devreye bir örnekte iki bitlik bir sayı ile 4 bitlik bir sayıyı çarpabilen 5 bitlik bir çıkışı 26 nS. de verebilen 74LS261 entegresidir. Bu devreler çarpma fonksiyonu ve lojik fonksiyonlarını ve diğer aritmetik fonksiyonların performansını artırmak için kullanılır.

74LS181 entegresi ise; ALU' (Aritmetic Logic Unit) ya bir örnektir. Bu iki tane 4 bitlik binary sayılardaki aritmetik ve lojik işlemlerini yapar (Öğrencinin ALU ünitesinin fonksiyonlarını araştırıp öğrenerek gelmesi gerekir.)

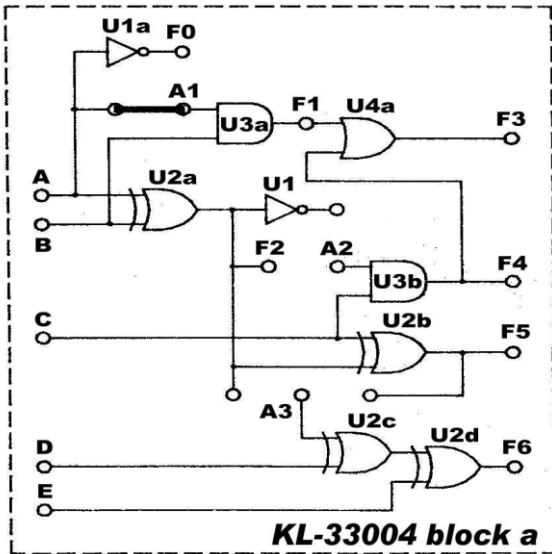
Ön Hazırlık Soruları

- 1- (101) ve (011)'in toplamı kaçtır.
- 2 - (011) ve (010) 'in farkını hesaplayınız.
- 3- 120'yi BCD'ye çeviriniz
- 4- (111001)/(10010)=?
- 5- (101)x(110) =?
- 6- 16'lık tabandaki FF'i binary'ye çevirin?
- 7- 16'lık tabandaki FF'i desimale çevirin?
- 8- ALU ve yaptığı iş nedir?
- 9-120 nin birli ve ikili tamamlayanını yazınız

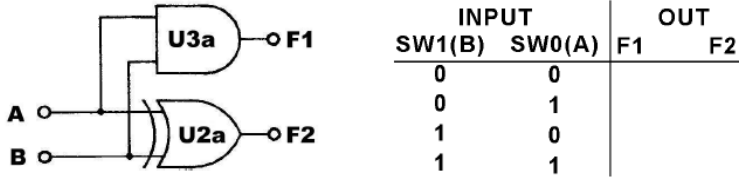
Deney.1 BİNARY TOPLAYICILAR

Bu deneyde binary toplayıcıların çalışması incelenecektir.

1. Bağlantı klipslerini aşağıdaki şekle göre yerleştirin. Şekildeki yarım toplayıcı devresini kurmak için U2a ve U3a kapıları kullanılacaktır. Vcc'yi +5V'a bağlayın.

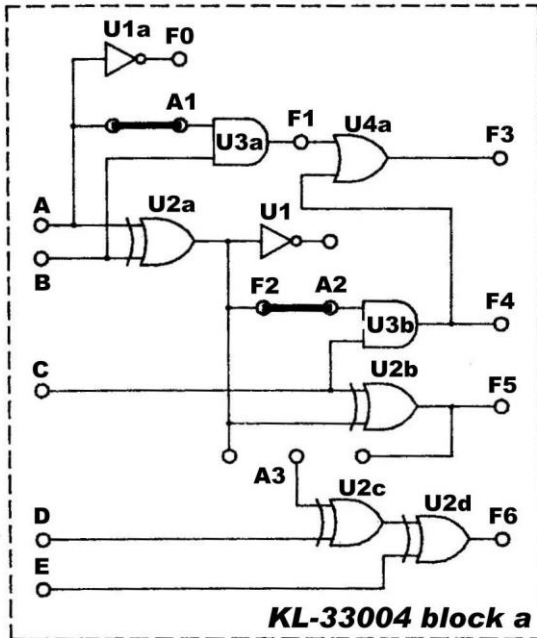
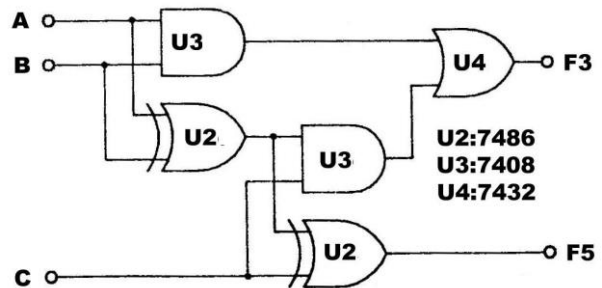


2. A ve B girişlerini SW0, SW1 veri anahtarlarına, F1, F2 çıkışlarını L1, L2 lojik göstergelerine bağlayın. A ve B girişleri için aşağıdaki tabloda verilen giriş sırasını takip ederek çıkış durumlarını kaydedin. Hangi çıkışın toplam, hangi çıkışın elde çıkışı olduğunu belirleyin.



3. Tam toplayıcıyı kurmak için devreyi aşağıdaki şekle göre tekrar düzenleyin. A, B, C girişlerini SW0, SW1, SW2 veri anahtarlarına bağlayın. A ve B toplananlar, C ise önceki elde girişleridir. F3, F5 çıkışlarını L1, L2 lojik göstergelerine bağlayın. Aşağıdaki tabloda verilen giriş sırasını takip ederek çıkış durumlarını kaydedin. Hangi çıkışın toplam, hangi çıkışın elde çıkışı olduğunu belirleyin.

OUTPUT			OUT	
SW3(C)	SW2(B)	SW1(A)	F3	F5
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		



Deney.2 TÜMDEVRE İLE TAM TOPLAYICI GERÇEKLEŞTİRMESİ

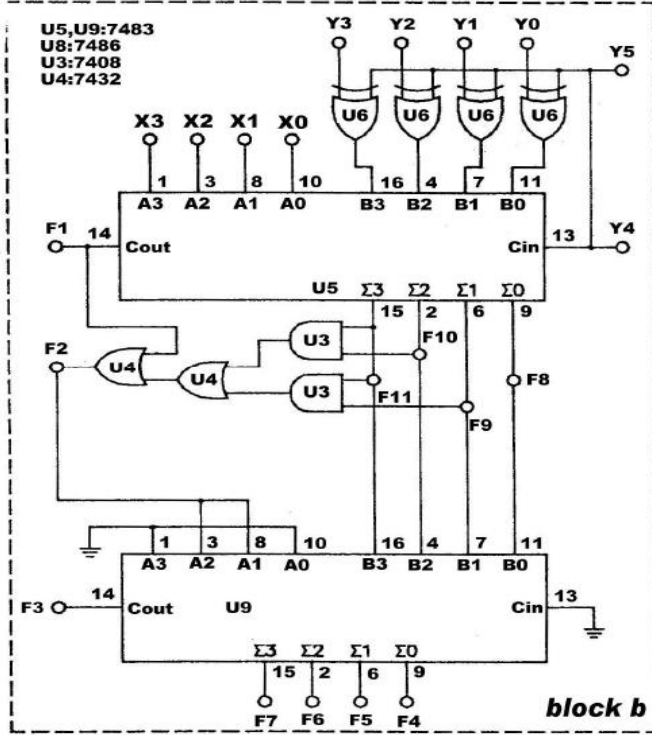
KL-330004 modülünün b bloğundaki U5 tüm devresi 4-bitlik toplayıcı olarak kullanılacaktır. Y5 girişini "0" değerine getirerek Y0~Y3 girişlerine bağlanmış olan U6a~U6d ÖZEL VEYA

kapılarının tampon gibi davranmalarını sağlayın. X0~X3 ve Y0~Y3 girişlerini sırasıyla DIP2.0~2.3 ve DIP1.0~1.3 lojik anahtarlara bağlayın. F1, Ç 0, Ç1, Ç2, Ç3 çıkışlarını ise L1~L5 lojik göstergelerine bağlayın. Aşağıdaki tabloda verilen giriş sırasını takip ederek F1 ve Ç için çıkış durumlarını 16 tabanına göre kaydedin.

X = X3 X2 X1 X0

Y = Y3 Y2 Y1 Y0

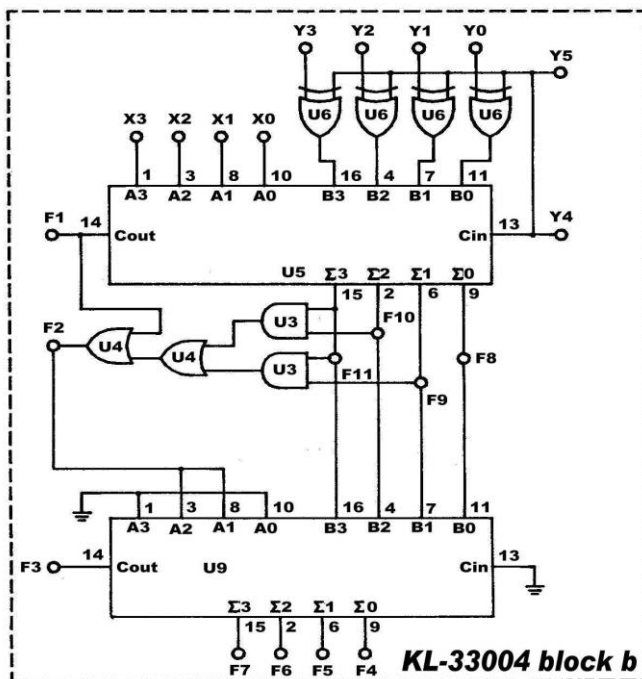
$\Sigma = \Sigma3 \Sigma2 \Sigma1 \Sigma0$



INPUT		OUTPUT	
Y	X	Σ	F1(CARRY)
0	0		
0	1		
0	6		
0	9		
0	F		
1	3		
1	6		
1	8		
3	6		
4	8		
4	F		
8	7		
9	9		
A	B		
C	E		
F	F		

Deney . 3 BCD TOPLAYICI

1. Aşağıdaki şekilde gösterilen devre BCD kodunda toplayıcı olarak görev yapar.



2. X0~X3 ve Y0~Y3 girişlerini sırasıyla DIP1.0~1.3 ve DIP2.0~2.3 lojik anahtarlara bağlayın.

U5 ve U9 tüm devreleri 7483 4-bitlik look-ahead (Öngörülü) toplayıcılardır. U5 tüm devresinin F8~F11 çıkışlarını 7 kollu göstergelerden birinin girişlerine bağlayın. F8~F11 çıkışlarını aynı zamanda L1~L4 lojik göstergelerine, F1, F2 çıkışlarını ise L5, L6 lojik göstergelerine bağlayın.

F4~F7 çıkışlarını diğer 7 kollu göstergeye bağlayın. F8~F11 çıkışlarını aynı zamanda L1~L4 lojik göstergelerine de bağlayın.

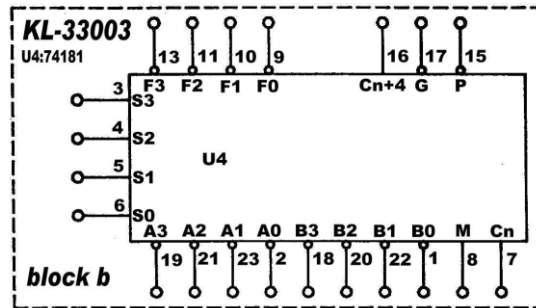
3. F8~F11 çıkışları, X0~X3 ve Y0~Y3 girişlerinin toplamı, F1 ise eldedir. X0~X3 ve Y0~Y3 girişleri için aşağıdaki tabloda verilen giriş sırasını takip ederek çıkış durumlarını kaydedin.

GİRİŞ								ÇIKIŞ(U5)					SON(U9)					
X3	X2	X1	X0	Y3	Y2	Y1	Y0	F1	F11	F10	F9	F8	F2	F3	F7	F6	F5	F4
0	0	0	0	0	0	0	0											
0	0	0	1	0	0	1	1											
0	0	1	1	0	1	0	0											
0	0	1	0	0	0	1	0											
0	0	1	0	1	0	0	0											
0	0	1	1	0	1	1	0											
0	1	0	0	0	0	1	0											
0	1	0	0	0	1	0	1											
0	1	0	0	0	1	1	0											
0	1	0	1	0	1	1	0											
0	1	1	0	0	1	1	1											
0	1	1	1	1	0	0	0											
0	1	1	1	1	0	0	1											
1	0	0	0	1	0	0	1											
1	0	0	1	1	0	0	1											
1	0	1	0	1	0	1	0											
1	0	1	0	1	0	1	1											
1	0	1	0	1	1	0	0											
1	0	1	1	1	1	1	0											
1	1	1	1	1	1	1	1											

Deney.4 ALU (ARİTHMETİC LOGİC UNIT)

Bu deneyin amacı Aritmetik lojik birimin (ALU) işlevlerini ve uygulamalarını anlamak.

1. Fonksiyon seçme girişlerini (S0~S3), SW3~SW0 veri anahtarlarına, M girişini ise DIP1.6 lojik anahtarına bağlayın. M="0" iken aritmetik işlemler, M="1" iken lojik işlemler gerçekleştirilecektir.



A0~A3 ve B3~B0 girişlerini sırasıyla DIP1.3~1.0 ve DIP2.3~2.0 lojik anahtarlarına, Cn'yi DIP2.6 lojik anahtarına, F3~F0 çıkışlarını L4~L1 lojik göstergelerine, Cn+4 çıkışını L8 lojik göstergesine bağlayın. A0~A3, B3~B0 2-77 ve F3~F0 "0" değeriyle tetiklenmektedir. Düşük seviye durumu "1" iken yüksek seviye durumu "0"dır.

2. Aşağıdaki lojik işlemleri gerçekleştirmek için M girişini "1" durumuna getirin.

(1) S3S2S1S0=0000 iken

A3A2A1A0=0000 ve B3B2B1B0=1111 iken çıkış ne olur?

A3A2A1A0=1100 ve B3B2B1B0=1010 iken çıkış ne olur?

(2) S3S2S1S0=1001 iken

A3A2A1A0=1100 ve B3B2B1B0=0110 iken çıkış ne olur?

Giriş ve çıkışların arasında nasıl bir lojik ilişki vardır?

3. Aşağıdaki aritmetik işlemleri gerçekleştirmek için M girişini “0” durumuna getirin. Cn girişine “0” uygulayarak önceki eldelere önem vermeyin.

(A) S3S2S1S0=1001 iken

(a) A3A2A1A0=B3B2B1B0=0100 iken çıkışlar ne olur?

F3F2F1F0 = ; Cn+4 = .

(b) A3A2A1A0=1000 ve B3B2B1B0=1100 iken çıkışlar ne olur?

F3F2F1F0 = ; Cn+4 = .

(B) S3S2S1S0=0011 iken

(a) A3A2A1A0=0100 ve B3B2B1B0=0010 iken çıkışlar ne olur?

F3F2F1F0 = ; Cn+4 = .

(b) A3A2A1A0=1010 ve B3B2B1B0=1000 iken çıkışlar ne olur?

F3F2F1F0 = ; Cn+4 = .