



Sayısal Devreler (Lojik Devreleri)

SAYISAL DEVRELER

Doç.Dr. Feza BUZLUCA
İstanbul Teknik Üniversitesi
Bilgisayar Mühendisliği Bölümü

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

 Sayısal Devreler Ders Notlarının Creative Commons lisansı Feza BUZLUCA'ya aittir.
Lisans: <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.tr>

<http://akademi.itu.edu.tr/buzluca>  2000-2019 Feza BUZLUCA 1.1


Sayısal Devreler (Lojik Devreleri)

İTÜ Bilgisayar Mühendisliği Bölümündeki donanım derslerinin bağlantıları

Sayısal devreler bölümündeki diğer donanım dersinin temelini oluşturmaktadır.

```

    graph TD
      EG[Elektronige Giriş] --> SED[Sayısal Elektronik Devreleri]
      SED <--> SD[Sayısal Devreler]
      SD <--> LD[Lojik Devre Lab.]
      SD --> MS[Mikroişlemci Sistemleri ve Lab.]
      SD --> BO[Bilgisayar Organizasyonu]
      MS --> BM[Bilgisayar Mimarisi]
      BO --> BM
  
```

<http://akademi.itu.edu.tr/buzluca>  2000-2019 Feza BUZLUCA 1.2


Sayısal Devreler (Lojik Devreleri)


Sayısal devrelerin kullanım yerleri:

Günümüzde çevremizde gördüğümüz neredeyse tüm elektronik cihazlar sayısal devreler içerirler.

Örnekler:

- Merkezi işlem birimi (CPU): Eş zamanlı ardışıl devredir. Bu tür sayısal devreleri dersin ikinci bölümünde inceleyeceğiz.
- Bilgisayar bellekleri: Belleklerin yapı taşlarını oluşturan "flip-flop" ve tutucu (latch) adlı sayısal devre elemanlarını bu ders kapsamında göreceğiz.
- Ev elektroniği: TV, çamaşır makinesi kontrol birimi, ses ve görüntü cihazları
- Otomobiller: ABS, motor ateşleme denetimi
- Cep telefonları



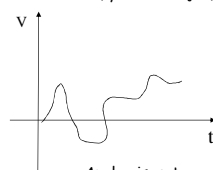
<http://akademi.itu.edu.tr/buzluca>  2000-2019 Feza BUZLUCA 1.3

Sayısal Devreler (Lojik Devreleri)

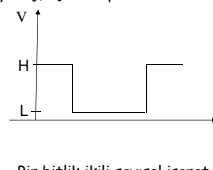
Analog - Sayısal (Dijital) İşaretler:

Gerçek dünyada karşılaştığımız bir çok fiziksel büyüklüğün (akım, gerilim, sıcaklık, ışık şiddeti vb.) değeri sürekli bir aralık içinde değişmektedir. Sınırlar arasındaki her türlü olası değeri alabilen bu tür işaretlere **analog** işaretler denir.

İkili (binary) sayısal işaretler ise belli bir anda sadece olası iki değerden birini alabilirler: 0 - 1, yüksek - alçak, doğru - yanlış, açık - kapalı.




Analog işaret



Bir bitlik ikili sayısal işaret

Bir analog işareti temsil edebilmek için bir bitten daha fazla ikili sayısal işarete gerek duyulur.

<http://akademi.itu.edu.tr/buzluca>  2000-2019 Feza BUZLUCA 1.4

Sayısal Devreler (Lojik Devreleri)


Sayısal Sistemlerin Avantajları:

Eskiden analog sistemlerin kullanıldığı bir çok alanda günümüzde daha avantajlı olduğundan sayısal sistemler kullanılmaktadır.

Örnekler: Fotoğrafçılık, video, ses kayıtları, otomobil motorları, telefon sistemleri vb.

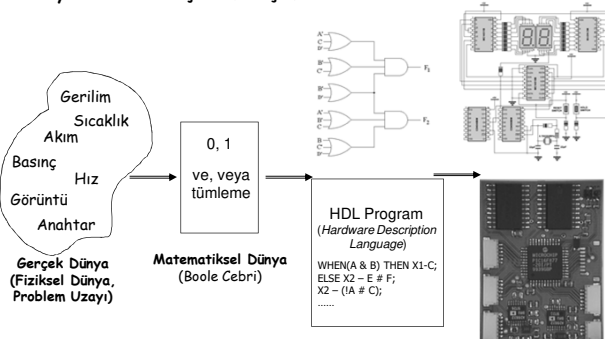
Sayısal Sistemlerin Avantajları:

- Bir sayısal sisteme aynı giriş kümesi defalarca uygulandığında hep aynı çıkış kümesi elde edilir.
- Burada aynı giriş kümesinin uygulanması demek her defasında aynı değer dizisinin aynı sırada uygulanması demektir. Analog sistemler ise çevre koşullarından daha çok etkilenirler ve çıkışları değişkenlik gösterebilir.
- Sayısal tasarım (lojik tasarım) dayandığı matematiksel temeller açısından daha kolaydır. Ayrıca sayısal sistemleri test etmek ve hatalardan arındırmak da analog sistemlere göre daha kolaydır.
- Esneklik ve programlanabilirlik. Günümüzde sayısal sistemleri programlanabilir bilgisayarlar şeklinde gerçekleştirmek mümkündür. Bu sayede aynı tasarım yeni gereksinimlere göre yeniden programlanarak tekrar kullanılabilir.
- Sayısal verileri bilgisayar ortamında saklamak ve işlemek mümkündür.
- Sayısal sistemler daha hızlı çalışmaktadır.
- Sayısal sistemler küçülmekte ve ucuzlanmaktadır.
- Sayısal sistemler gelişmeye devam ediyor.

<http://akademi.itu.edu.tr/buzluca>  2000-2019 Feza BUZLUCA 1.5

Sayısal Devreler (Lojik Devreleri)

Sayısal Devre Gerçekleme Aşamaları:




Gerçek Dünya (Fiziksel Dünya, Problem Uzayı): Gerilim, Akım, Sıcaklık, Basınç, Hız, Görüntü, Anahtar

Matematiksel Dünya (Boole Cebri): 0, 1 ve, veya tümleme

HDL Program (Hardware Description Language): WHEN(A & B) THEN X1=C; ELSE X2 = E # F; X2 = (A # C);

Gerçekleme (Kodlama)

<http://akademi.itu.edu.tr/buzluca>  2000-2019 Feza BUZLUCA 1.6

Sayısal Devreler (Lojik Devreleri) Lisans: <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.tr/>

İkili Sayısal Kodlama (Binary Digital Coding):

Sayısal sistemler ikili sayısal işaretler üzerinde işlemler yaptıklarından sadece iki farklı değeri (binary) işleyebilirler: AÇIK-KAPALI, ALÇAK-YÜKSEK, 0-1.

Bu nedenle sayısal devreler yardımıyla üzerinde işlem yapılacak olan fiziksel büyüklüklere (gerilim, sıcaklık vs.) ve her türlü veriye (harf, sayı, renk, ses) ikili sayılar karşı düşürülür.

Gerçek dünya → Kodlama → Kod sözcükleri

01101101
00100110
11110011
00000011
...

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.7

Sayısal Devreler (Lojik Devreleri)

Sayısal Kodlama (devamı):

n basamaklı (n bitlik "Binary digit") bir ikili sayı kullanarak 2^n tane farklı "şey" ifade edebiliriz.

n bit → 2^n farklı "şey"

Örneğin 8 basamaklı (8 bitlik) bir ikili sayı kullanarak 2^8 tane (256) farklı "şey" ifade edebiliriz.

Bunlar 256 farklı renk, 256 sembol, 0 ile 255 arası tamsayılar, 1 ile 256 arası tamsayılar, -128 ile +127 arası tamsayılar olabilir.

0000000, 00000001, 00000010, ... , 11111101, 11111110, 11111111.

Farklı türde veriler için farklı kodlama sistemleri (yöntemleri) kullanılır.

Bir ikili değer (Örneğin 10001101) ne anlama geldiğine o değeri kullanacak olan sistem (donanım ya da yazılım sistemi olabilir) ya da kişi belirler.

Bu değer bir sayı da olabilir, bir renk de.

Özellikle sayıların kodlanması büyük önem taşır.

Bu nedenle derste sayıların kodlanmasına ilişkin temel bilgiler verilecektir.

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.8

Sayısal Devreler (Lojik Devreleri)

BCD (Binary Coded Decimal) İkili kodlanmış onlu sayılar:

0-9 arasındaki rakamlara 4 bitlik bir ikili kod karşı düşürülür.

Doğal BCD:

Sayı:	Kod:	Sayı:	Kod:	Örnek:
0:	0000	5:	0101	Sayı: 805
1:	0001	6:	0110	Kod:1000 0000 0101
2:	0010	7:	0111	
3:	0011	8:	1000	
4:	0100	9:	1001	

Artıklı (redundant) kodlamadır. Çünkü 4 bit ile 16 farklı kodlama yapılabilmekte, ancak bunlardan sadece 10 tanesi kullanılmaktadır.

BCD sayılar üzerinde işlem yapmak zor olduğundan günümüz bilgisayarlarında sayıları göstermek için bu kodlama kullanılmamaktadır.

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.9

Sayısal Devreler (Lojik Devreleri)

Ağırlıklı Kodlama:

Bitlerin konumlarına birer ağırlık verilir.

Doğal ikili kodlama: Sayıların ağırlıklı kodlama ile 2 tabanında gösterilmesidir.

Örneğin: $11010 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 26$

Soldaki ilk basamağa yüksek anlamlı bit (Most Significant Bit - MSB)

Sondaki basamağa düşük anlamlı bit (Least Significant Bit - LSB) denir.

Bilgisayarlarda sayıları göstermek için doğal ikili kodlama kullanılır.

Hamming uzaklığı: n uzunluğundaki iki kod sözcüğünün arasındaki Hamming uzaklığı, o sözcüklerdeki aynı sırada olup değerleri farklı olan bileşenlerin sayısıdır.

Örneğin: 011 ile 101 arasındaki uzaklık 2 dir.

Richard Wesley Hamming (1915-1998) Matematikçi, ABD

Bitişik kodlar: Bir birini izleyen sayılara karşı gelen kodlar arasındaki uzaklık 1 ise o kodlama bitişiktir.

Ayrıca son sayı ile ilk sayı arasındaki uzaklık da 1 olursa kod çevrimlidir.

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.10

Sayısal Devreler (Lojik Devreleri)

Örnek: Çevrimli bir BCD kodu (doğal BCD'den farklı)

Sayı:	Kod:	Sayı:	Kod:
0:	0000	5:	1110
1:	0001	6:	1010
2:	0011	7:	1000
3:	0010	8:	1100
4:	0110	9:	0100

Gray Kodu: 2^n elemanlı bir küme için 2 tabanında artıksız ve çevrimli bir kodlama yapılırsa gray kodu elde edilir.

Örnek: 2 bitlik bir Gray kodu:

Sayı:	Kod:
0:	00
1:	01
2:	11
3:	10

Gray Kodunun patenti 1953'te Bell Laboratuvarında çalışan fizikçi Frank Gray tarafından alınmıştır.

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.11

Sayısal Devreler (Lojik Devreleri)

Sayıların Bilgisayarda (Sayısal Sistemlerde) Gösterilmesi

Bu derste tamsayıların gösterilimine ilişkin bilgiler verilecektir.

Kayan noktalı (floating point) sayıların gösterilmesi Bilgisayar Mimarisi dersinde ele alınmaktadır (Bkz. <http://www.buzluca.info/dersler.html>).

Sayılar kodlanmadan önce işaretli ya da işaretli sayılarla çalışılacağı belirlenmelidir. Çünkü işaretli ve işaretli sayıların kodlanmasında farklı yöntemler kullanılmaktadır.

İşaretsiz (Unsigned) Tamsayıların Kodlanması:

Bilgisayarlarda işaretli tamsayıların ifade edilmesinde "doğal ağırlıklı ikili kodlama" kullanılır.

Örnek: $215_{10} = (11010111)_2 = 1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$

$215/2 = 107$ kalan 1 (düşük anlamlı bit "Least Significant Bit - LSB") son basamak

$107/2 = 53$ kalan 1

$53/2 = 26$ kalan 1

$26/2 = 13$ kalan 0

$13/2 = 6$ kalan 1

$6/2 = 3$ kalan 0

$3/2 = 1$ kalan 1

$1/2 = 0$ kalan 1 (yüksek anlamlı bit "Most Significant Bit - MSB") ilk basamak

8 bit ile ifade edilebilecek en büyük işaretli tamsayı: $11111111_2 = 255_{10}$

8 bit ile ifade edilebilecek en küçük işaretli tamsayı: $00000000_2 = 0_{10}$

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.12

Sayısal Devreler (Lojik Devreleri) Lisans: <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.tr/>

İşaretili (Signed) Tamsayıların Kodlanması:


Pozitif ve negatif sayıları ayırt etmek için ikili sayının ilk basamaktaki en yüksek anlamlı bitine bakılır.

- "0" ile başlayan sayıların **pozitif**,
- "1" ile başlayan sayıların **negatif** olduğu kabul edilir.

Pozitif tamsayılar:
Pozitif sayıların kodlanmasında (işaretsiz sayılarda olduğu gibi) "doğal ağırlıklı ikili kodlama" kullanılır.
Dikkat edilmesi gereken nokta sayının 0 ile başlamasıdır.
Buna göre 8 bit ile temsil edilebilecek pozitif işaretili sayılar:
0000 0000 ile 0111 1111 arasında (yani 0 ile +127 arasında) değişecektir.

Pozitif Tamsayı Örnekleri:

8 bit +5 ₁₀	: 0000 0101
8 bit +100 ₁₀	: 0110 0100
4 bit +5 ₁₀	: 0101
4 bit +7 ₁₀	: 0111

<http://akademi.itu.edu.tr/buzluca> <http://www.buzluca.info>  2000-2019 Feza BUZLUCA 1.13

Sayısal Devreler (Lojik Devreleri)

Negatif Tamsayılar:

Negatif sayıların kodlanmasında **2'ye tümeleme (2's complement)** yöntemi kullanılmaktadır.
Bu yöntemde pozitif bir sayının 2'ye tümeleyeni hesaplandığında o sayının negatif gösterilimi elde edilmiş olur.
Bir sayının **2'ye tümeleyeni elde etmek için**


- Önce sayı 1'e tümlenir, yani 0'lar 1, 1'ler 0 yapılır,
- 1'e tümlenmiş sayıya 1 eklenir.

$$2'ye\ tümeleyen(A) = \bar{A} + 1$$

\bar{A} , 1'e tümeleyeni göstermektedir.
2'ye tümeleme yöntemi aritmetik işlemlerde kolaylık sağladığı için tercih edilmektedir.

Negatif Sayılara Örnekler:

8 bitlik +5 ₁₀	: 0000 0101	4 bitlik +7 ₁₀	: 0111
1'e tümeleme	: 1111 1010	1'e tümeleme	: 1000
1 ekleme	: +_____1	1 ekleme	: +_____1
Sonuç -5 ₁₀	: 1111 1011	Sonuç -7 ₁₀	: 1001

<http://akademi.itu.edu.tr/buzluca> <http://www.buzluca.info>  2000-2019 Feza BUZLUCA 1.14

Sayısal Devreler (Lojik Devreleri)


2'ye tümeleme yöntemi (devamı):

2'ye tümeleme işlemi bir sayının **işaretini değiştirir**.
Bir negatif sayıya 2'ye tümeleme işlemi uygulandığında o sayının pozitif değeri elde edilmiş olur.
2'ye tümeleme işlemi:

pozitif → negatif
negatif → pozitif

Örnek: Negatif bir sayının pozitif yapılması:

8 bitlik -5 ₁₀	: 1111 1011
1'e tümeleme	: 0000 0100
1 ekleme	: +_____1
Sonuç +5 ₁₀	: 0000 0101


<http://akademi.itu.edu.tr/buzluca> <http://www.buzluca.info>  2000-2019 Feza BUZLUCA 1.15

Sayısal Devreler (Lojik Devreleri)

İşaretili sayılar bir grafik üzerinde gösterilebilir.
Aşağıda 4 bitlik sayılar gösterilmiştir.

4 bit ile ifade edilebilecek mutlak değeri en büyük negatif tamsayı
1000 = -8
Mutlak değeri en küçük negatif tamsayı 1111 = -1

8 bit mutlak değeri en büyük negatif tamsayı 1000 0000 = -128
Mutlak değeri en küçük negatif tamsayı 1111 1111 = -1

<http://akademi.itu.edu.tr/buzluca> <http://www.buzluca.info>  2000-2019 Feza BUZLUCA 1.16

Sayısal Devreler (Lojik Devreleri)


İkili Sayıların Uzatılması (Sign Extension)

Sayısal sistemlerde ikili sayılar için belli uzunlukta yerler (bellek gözleri) ayrılır.
Bazı durumlarda daha az bit ile ifade edilebilen bir sayıyı daha büyük bir yere yazmak ya da daha uzun bir sayı ile işleme sokmak gerekebilir.
Bu durumda kısa olan sayı uzatılır.
Örneğin 4 bitten 8 bite veya 8 bitten 16 bite uzatma.
Uzatma işleminde sayının işaretsiz ya da işaretili olmasına göre farklı yollar izlenir.

İşaretsiz Sayılar: Sayının başına (yüksek anlamlı kısmına) gerektiği kadar sıfır '0' eklenir.
Örnek: 4 bitlik 3₁₀: 0011 8 bitlik 3₁₀: 0000 0011
Örnek: 4 bitlik 9₁₀: 1001 8 bitlik 9₁₀: 0000 1001

İşaretili Sayılar: Sayının başına (yüksek anlamlı kısmına) sayının işareti gerektiği kadar eklenir. Buna **işaret uzatma (sign extension)** denir.

Örnek: 4 bitlik +3 ₁₀	= 0011	8 bitlik +3 ₁₀	= 0000 0011
Örnek: 4 bitlik -7 ₁₀	= 1001	8 bitlik -7 ₁₀	= 1111 1001
Örnek: 4 bitlik -1 ₁₀	= 1111	8 bitlik -1 ₁₀	= 1111 1111

<http://akademi.itu.edu.tr/buzluca> <http://www.buzluca.info>  2000-2019 Feza BUZLUCA 1.17

Sayısal Devreler (Lojik Devreleri)

16 Tabanın Kullanılması


Sayısal devrelerin yapıları 2'li sayıların kullanılmasını zorunlu hale getirmiştir.
Ancak 2'li sayıların yazılması ve okunması uzunlukları nedeniyle zor olmaktadır.
Bu nedenle kağıt üstündeki gösterimlerde kolaylık sağladığı için 16 tabanında (hexadecimal) sayılar kullanılmaktadır.
2'li - 16'lı Dönüşüm:

- 2'li sayı 4 bitlik gruplar halinde yazılır,
- Her dörtlü için 16'lık karşılığı yazılır.

Örnek:
01011101₂ = 0101 1101 (İkili - Binary)
= 5 D (Onaltılı - Hexadecimal)

10 tabanına dönüşüm:
5D₁₆ = (5 x 16) + 13 = 93
Sonuç: 01011101₂ = 5D₁₆ = 93₁₀
16 tabanındaki sayıları göstermek için genellikle \$ ve h simgeleri kullanılır.
Örnek: \$5D veya 5Dh.

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

<http://akademi.itu.edu.tr/buzluca> <http://www.buzluca.info>  2000-2019 Feza BUZLUCA 1.18

Sayısal Devreler (Lojik Devreleri)

Sayısal Sistemlerde Toplama ve Çıkarma İşlemleri

Bilgisayarlarda tamsayı aritmetik işlemleri Aritmetik/Lojik Birim (ALB) tarafından yapılır (*Arithmetic Logic Unit - ALU*).

Tamsayı toplama ve çıkarma işlemleri işaretli ve işaretli sayılar üzerinde aynı şekilde yapılır.

Ancak çıkan sonucun yorumlanması işaretli ve işaretli sayılarda farklı olmaktadır. Farklı uzunluklarda sayılar ile işlem yaparken kısa sayının uzatılması gerekir. Uzatma işlemi işaretli ve işaretli sayılarda farklı olur. (Bkz. 1.15)

Toplama:

İşaretsiz Tamsayılar:

n bitlik iki işaretli sayının toplanması sonucu n+1 bitlik bir sayı oluşabilir. (n+1). bit **elde** (*carry*) adını alır.

a	b	Elde	Sonuc
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Örnekler: 8 bitlik işaretli sayıların toplanması

$$\begin{array}{r} 01110101 : 117 \\ + 01100011 : 99 \\ \hline 11011000 : 216 \end{array}$$

Sadece sağlama için

$$\begin{array}{r} 11111111 : 255 \\ + 00000001 : 1 \\ \hline 100000000 : 256 \end{array}$$

Sadece sağlama için

Elde oluşmadı. (Elde=0)

Elde oluştu. (Elde=1)

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.19

Sayısal Devreler (Lojik Devreleri) Lisans: <https://creativecommons.org/licenses/by-nc-nd/4.0/deed.tr/>

Toplama:

İşaretili Tamsayılar:

- İşlem işaretli sayılarda olduğu gibi yapılır.
- Ancak sonucun yorumlanması farklıdır.
- n bitlik işaretli iki sayının toplanması sonucu (n+1). bit oluşursa bu bit göz ardı edilir.

Örnek: 8 bitlik işaretli sayıların toplanması

$$\begin{array}{r} 11111111 : -1 \\ + 00000001 : +1 \\ \hline 100000000 : 0 \end{array}$$

Göz ardı edilir. İşaret (+)

$$\begin{array}{r} 11111111 : -1 \\ + 11111111 : -1 \\ \hline 111111110 : -2 \end{array}$$

Göz ardı edilir. İşaret (-)

Dikkat:

Eğer n bitlik sayılarla çalışılıyorsa işaret her zaman en yüksek anlamlı bit (sağdan sola doğru sayıldığında) n. bittir (n+1. değil).

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.20

Sayısal Devreler (Lojik Devreleri)

Taşma (Overflow) (işaretili tamsayılar):

İşaretili sayılarda toplama sonucu oluşan değer n bit ile gösterilemeyebilir. Örneğin 8 bit ile gösterilebilecek sayılar -128, +127 arasındadır. Oluşan sonuç bu aralığın dışına çıkıyorsa **taşma** oluşur.

Toplama sonucunda taşma oluştuğu toplanan sayıların ve sonucun işaretinden anlaşılır.

Toplamada iki durumda taşma oluşabilir:

poz + poz → neg ve neg + neg → poz

Örnek:

$$\begin{array}{r} 01111111 : +127 \\ + 00000010 : +2 \\ \hline 10000001 : Gösterilemiyor \end{array}$$

İki pozitif sayı toplandı. Sonuç negatif çıktı. **Taşma** vardır. Not: n+1. bit oluşmadı. Bu bit göz ardı edilir.

$$\begin{array}{r} 10000000 : -128 \\ + 11111111 : -1 \\ \hline 101111111 : Gösterilemiyor \end{array}$$

İki negatif sayı toplandı. Sonuç pozitif çıktı. **Taşma** vardır. Not: n+1. bit oluştu. Bu bit göz ardı edilir.

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.21

Sayısal Devreler (Lojik Devreleri)

Çıkarma:

- Bilgisayarlar, çıkarma işlemi için 2'ye tümele yönteminden yararlanırlar.
- Çıkartılacak olan sayının işareti değiştirilip (2'ye tümlenip) birinci sayı ile toplanır.

$$\begin{aligned} A - B &= A + (-B) \\ &= A + 2'ye \text{ tümele } (B) \\ &= A + \bar{B} + 1 \end{aligned}$$

Böylece tek bir toplama devresi ile hem toplama hem çıkarma yapmak mümkün olur.

Toplama işleminde olduğu gibi, çıkarma işlemleri de işaretli ve işaretli sayılar üzerinde aynı şekilde yapılır.

Ancak çıkan sonucun yorumlanması işaretli ve işaretli sayılarda farklı olmaktadır.

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.22

Sayısal Devreler (Lojik Devreleri)

Çıkarma (devamı):

İşaretsiz Tamsayılar:

n bitlik iki işaretli sayı arasında 2'ye tümlen yöntemine göre çıkarma yapıldığında n+1. bit oluşursa sonuç geçerlidir, borç (*borrow*) yoktur.

Eğer n+1. bit oluşmazsa (sıfır) birinci sayı ikinciden küçüktür, **borç** vardır.

Örnekler: 8 bitlik işaretli sayılar ile çıkarma

$$\begin{array}{r} 00000101 : 5 \\ - 00000001 : 1 \\ \hline 100000000 : 4 \end{array}$$

2'ye tümele → Elde oluştu: **Borç Yok**

$$\begin{array}{r} 00000001 : 1 \\ - 00000101 : 5 \\ \hline 011111100 : Borç (ilk sayı küçüktür) \end{array}$$

2'ye tümele → Elde oluşmadı: **Borç Var**

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.23

Sayısal Devreler (Lojik Devreleri)

Çıkarma (devamı):

İşaretili Tamsayılar:

İşaretili tamsayılar arasındaki çıkarma da, işaretli sayılarda olduğu gibi 2'ye tümlen yöntemine göre yapılır.

Elde biti göz ardı edilir.

Toplamada olduğu gibi işaretli sayıların çıkarılmasında da taşma olabilir.

Çıkarmada iki durumda taşma oluşabilir:

poz - neg → neg ve neg - poz → poz

Örnek: 8 bitlik işaretli sayılar ile çıkarma

$$\begin{array}{r} 00000101 : 5 \\ - 00001100 : 12 \\ \hline 111100010 : -12 \end{array}$$

2'ye tümele → İşaret 1, sonuç **negatif**

$$\begin{array}{r} 11111001 : -3 \\ - 01111111 : 127 \\ \hline 101111110 : Gösterilemiyor \end{array}$$

2'ye tümele → İşaret biti 0, Sonuç **pozitif**

Neg - poz = poz. **Taşma** vardır.

<http://akademi.itu.edu.tr/buzluca>
<http://www.buzluca.info>

2000-2019 Feza BUZLUCA 1.24

Tamsayıların karşılaştırılması:

Tamsayıların karşılaştırılması için çıkarma işlemi kullanılır

$S = A - B$, çıkarma işleminden sonra ilgili bayraklar (durum bitleri) kontrol edilir.

İşaretsiz Tamsayılar:

Borç	Sonuç (S)	Karşılaştırma
X (önemli değil)	=0	A=B
YOK (elde = 1)	≠0	A>B
VAR (elde = 0)	≠0	A<B

İşaretlili Tamsayılar :

Taşma	Sonuç (S)	Karşılaştırma
X (önemli değil)	=0	A=B
YOK	Pozitif, ≠0	A>B
YOK	Negatif	A<B
VAR	Pozitif	A<B
VAR	Negatif	A>B

Elde (Carry), Borç (Borrow), Taşma (Overflow) Kavramlarının Özeti

Elde: İşaretsiz sayıların toplanmasında oluşabilir. Sonucun n bite sığmadığını, (n+1). bitin gerekli olduğunu gösterir.

Borç: İşaretsiz sayıların çıkartılmasında oluşabilir. Birinci sayının ikinciden küçük olduğunu, sonucun negatif çıktığını gösterir. 2'ye tımleyen yöntemine göre yapılan çıkarmada n+1. bit oluşursa borç yoktur.

Taşma: Sadece işaretlili sayılar üzerinde yapılan toplama ve çıkarma işlemlerinde oluşur. Sonucun, ayrılan bit sayısı ile ifade edilemediğini gösterir.

Taşma olduğu, işleme giren sayıların ve sonucun işareti incelenerek anlaşılır.

Aşağıdaki durumlar oluştuğunda taşma var demektir:

poz + poz → neg poz – neg → neg

neg + neg → poz neg – poz → poz