

Mantıksal Sistem Tasarımı – BLM 201

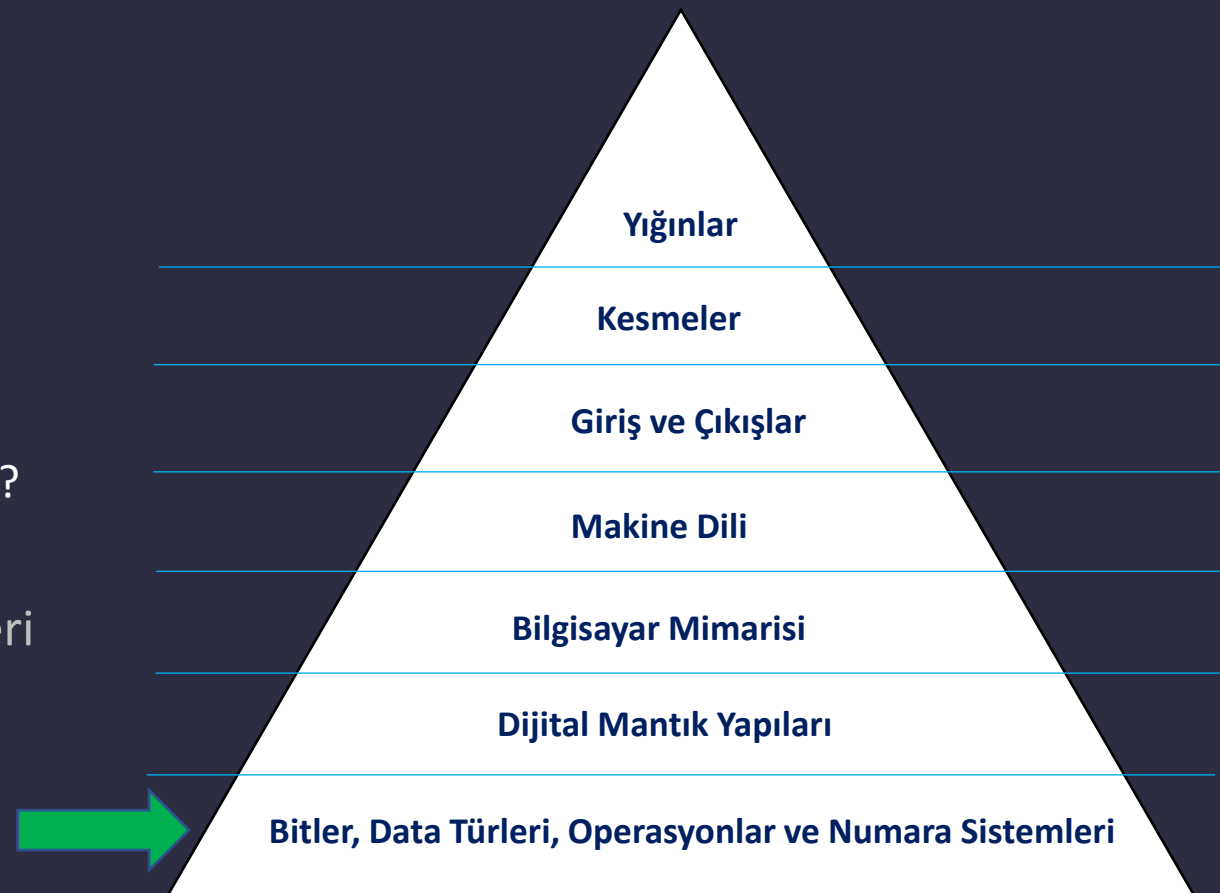
Hafta 2: Sayı Sistemleri ve Boolean Cebri Bölüm II



Fenerbahçe Üniversitesi

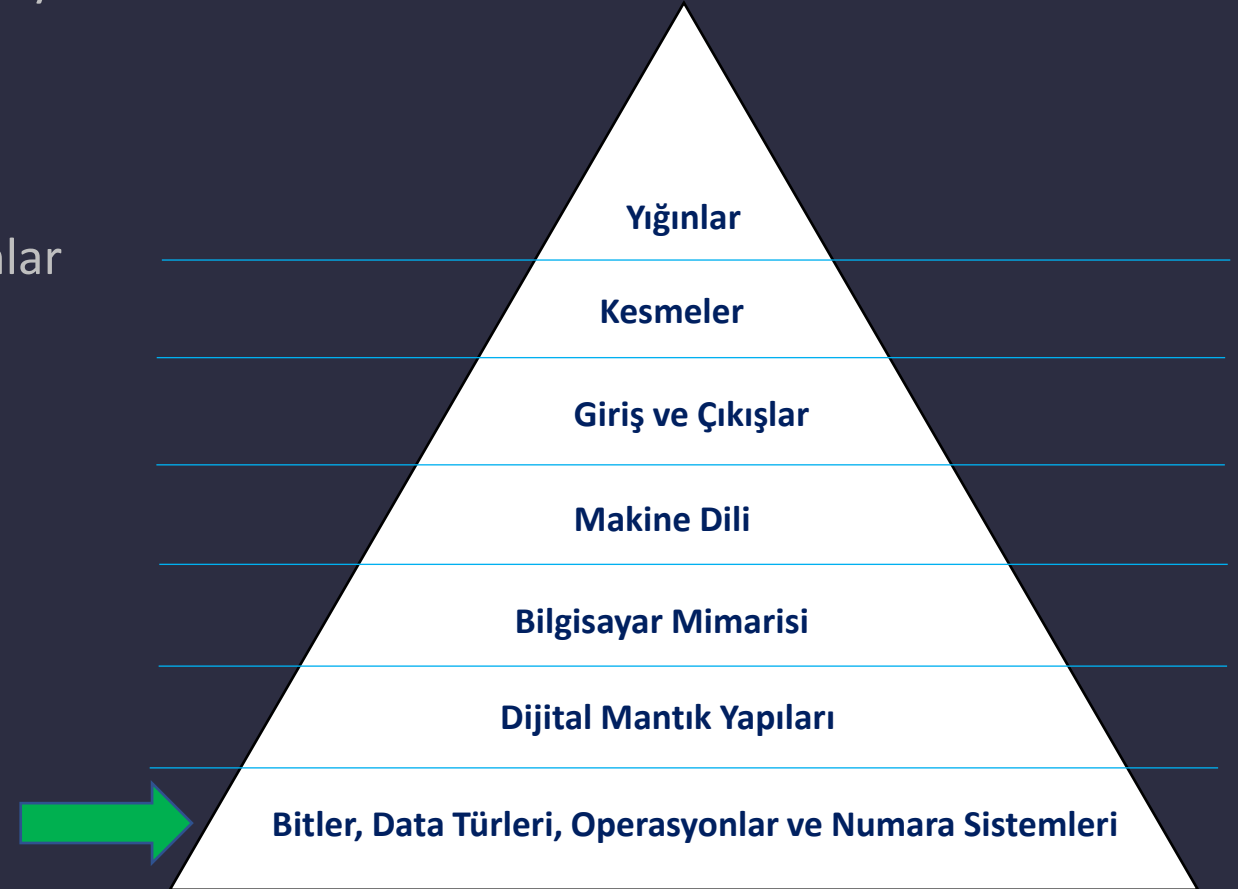
2. Hafta İçeriği

- Bitler ve Veri Türleri
 - En küçük bilgi birimi bitler
 - Veri tipleri
- Tamsayı (integer) Veri
 - İşaretsiz tamsayılar
 - İşaretsiz tamsayılar
- 2'e Tümlleyen (2's complement) Tamsayılar
 - Dijital lojik yapılar ile işlemciler nasıl yapıyor?
 - İşlemciler üzerinde komutlar nasıl çalışıyor?
- İkilik (Binary) – Ondalık (Decimal) Dönüşümleri
 - İkilik'den onluk'a dönüşüm
 - Onluk'tan ikilik'e dönüşüm



2. Hafta İçeriği

- Bitler üzerinde operasyonlar, Aritmetik Operasyonlar
 - Toplama ve çıkartma
 - İşaret uzatma
 - Taşma
- Bitler üzerinde operasyonlar, Lojik Operasyonlar
 - Ve (And)
 - Veya (Or)
 - Değil (Not)
 - Özel Veya (XOR)
- Diğer Gösterimler
 - Bit vektörü
 - Floating Point veri tipi
 - ASCII kodları
 - Hexadecimal gösterimi



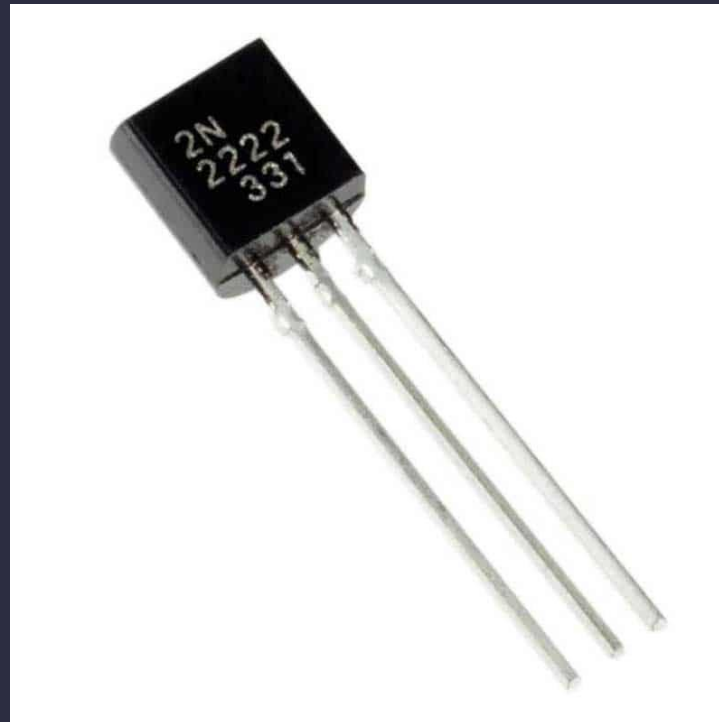
Veri Bilgisayarda Nasıl Saklanmaktadır?

- Bilgisayar bir elektronik devredir.
 - En temelde elektronların akışlarını kontrol ederek çalışır



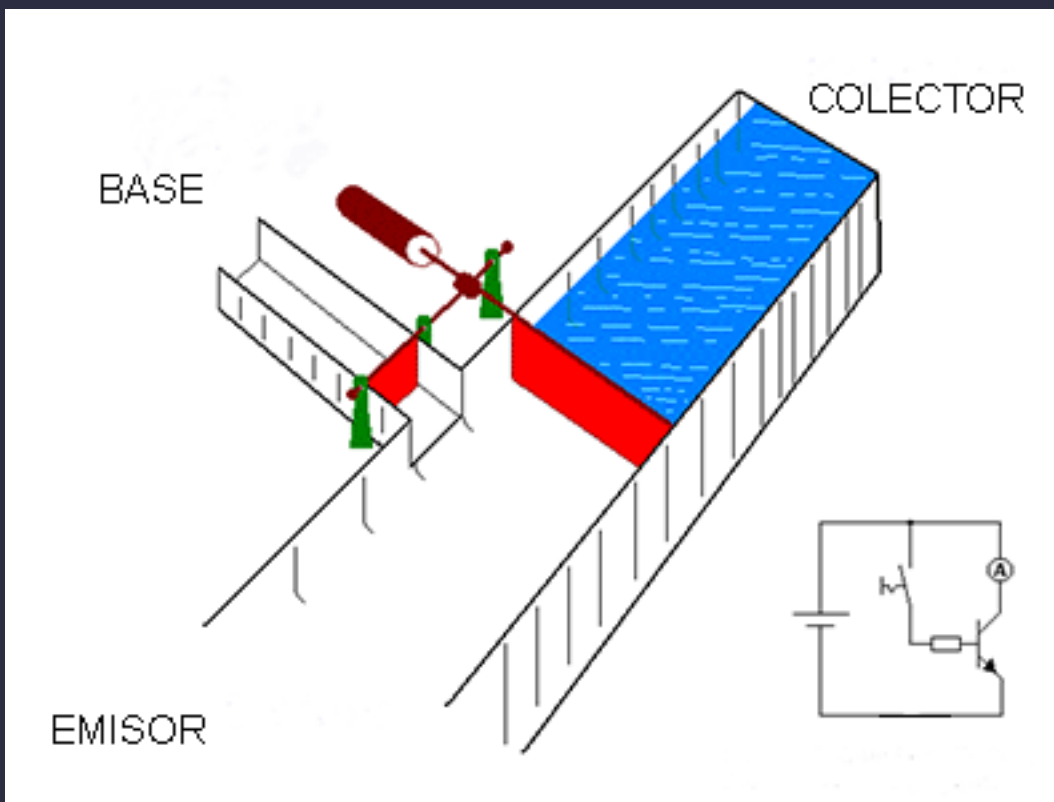
Veri Bilgisayarda Nasıl Saklanmaktadır?

- Elektronların kontrolü "Transistörler" ile yapılmaktadır.
 - En temelde elektronların akışlarını kontrol ederek çalışır



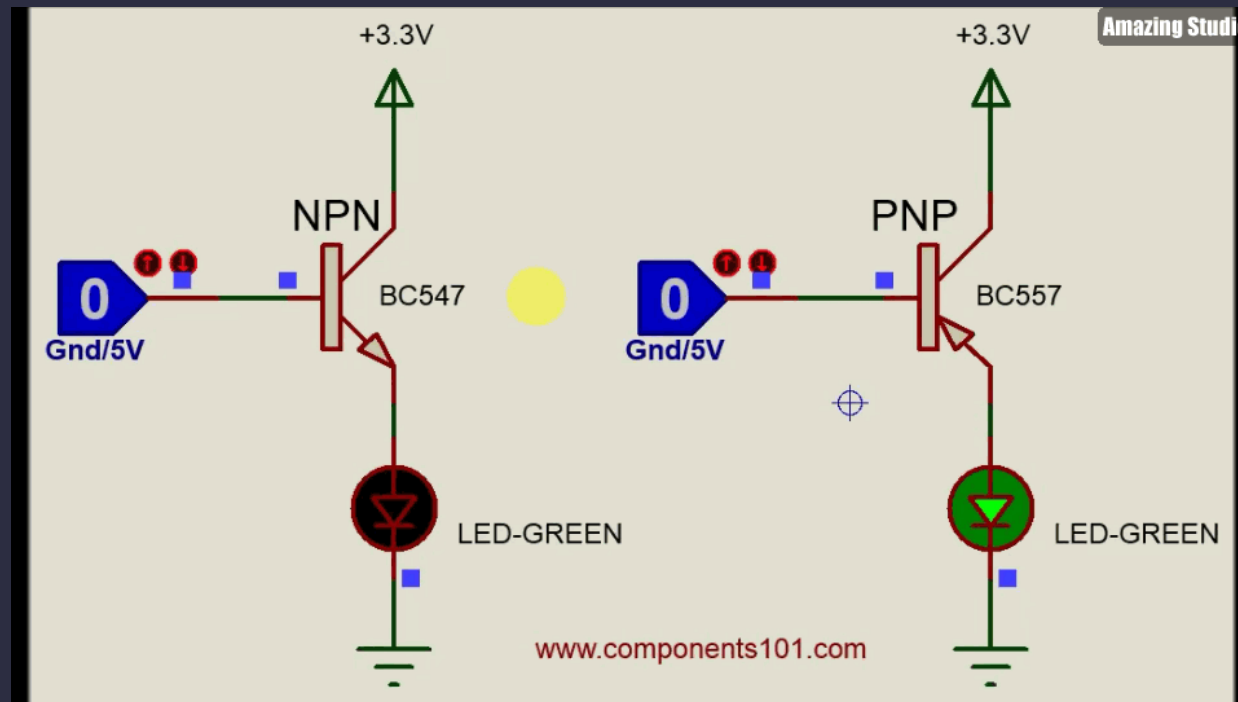
Veri Bilgisayarda Nasıl Saklanmaktadır?

- Elektronların kontrolü "Transistörler" ile yapılmaktadır.
 - En temelde elektronların akışlarını kontrol ederek çalışır



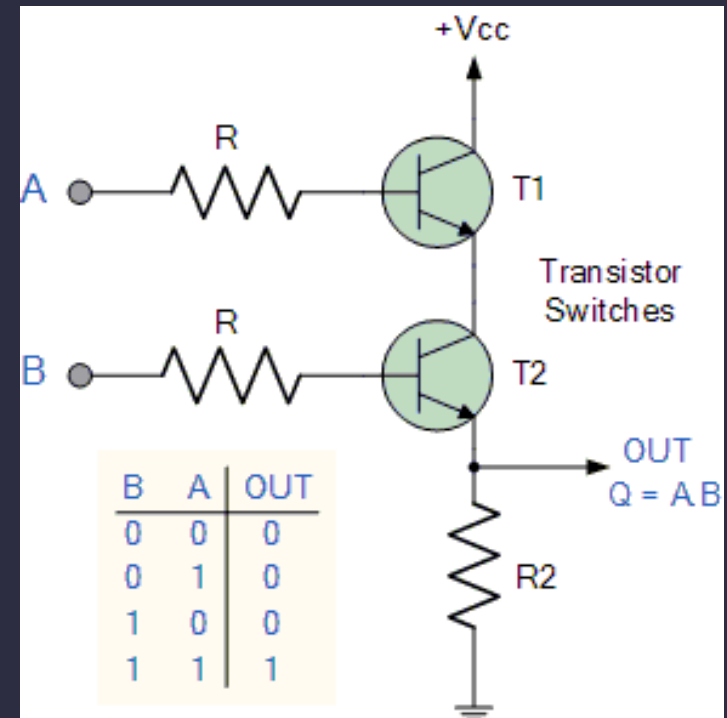
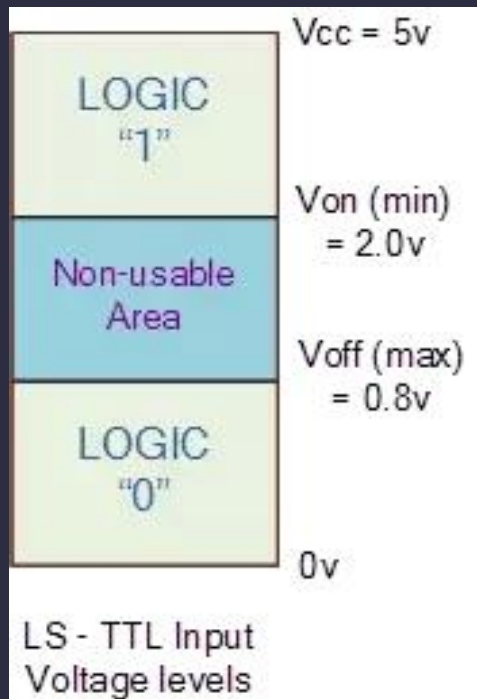
Veri Bilgisayarda Nasıl Saklanmaktadır?

- Elektronların kontrolü "Transistörler" ile yapılmaktadır.
 - En temelde elektronların akışlarını kontrol ederek çalışır



Veri Bilgisayarda Nasıl Saklanmaktadır?

- Verinin iki durumu vardır:
 1. Gerilimin (Voltage)'in var olduğu durum – Bu duruma "1" denir.
 2. Gerilimin yok olduğu durum- Bu duruma "0" denir.



Veri Bilgisayarda Nasıl Saklanmaktadır?

- Sadece gerilimin var olup olmadığına bakmak yerine, gerilim seviyelerine göre çalışan bir bilgisayar da yapmak mümkündür.
 - Ancak bu bilgisayarın kontrol devresi çok daha karmaşık olacaktır.
 - Bu nedenle günümüz modern bilgisayarları, bilgiyi ifade ederken en küçük birim olan bit kavramı ile çalışır.
 - Bir bit üzerinde 0 yada 1 tutabilen en küçük veri saklama birimidir.

Bilgisayar ikilik tabanda (binary) bir sistemdir.

İkilik (Binary) sistem:

- İki durumu bulunmaktadır: 0 ve 1
- Birden çok bitin bir araya gelmesiyle daha büyük saklama alanları elde edilir.
 - İki bit'in bir araya gelmesi ile, 4 farklı sayı ifade edilebilmektedir.

Bilgisayar ikilik tabanda (binary) bir sistemdir.

- İki bit'in bir araya gelmesi ile, 4 farklı sayı ifade edilebilmektedir.

- $00 = 0$ (onluk tabanda)
- $01 = 1$
- $10 = 2$
- $11 = 3$

Bilgisayar ikilik tabanda (binary) bir sistemdir.

- 3 bit'in biraraya gelmesi ile 8 sayı ifade edilebilmektedir:
- $000 = 0$
- $001 = 1$
- $010 = 2$
- $011 = 3$
- $100 = 4$
- $101 = 5$
- $110 = 6$
- $111 = 7$

Bilgisayar ikilik tabanda (binary) bir sistemdir.

- Özetle;
- n adet bit ile 2^n farklı sayı ifade edilebilmektedir.
- 2 bit için $2^2 = 4$ farklı sayı
- 3 bit için $2^3 = 8$ farklı sayı
- 4 bit için $2^4 = 16$ farklı sayı
- ...

ifade edilebilir.

Ne tür veriler bilgisayarda ifade edilmek / saklanmaktadır?

- Sayılar – işaretli (signed), işaretsiz (unsigned), tam sayılar (integer), ondalıklı sayılar (floating point), karmaşık sayılar (complex), rasyonel, irrasyonel, ...
 - Metinler – Karakterler (characters), metinler (string), ...
 - Resimler – piksel, resimler, ...
 - Ses
 - Mantık (logic) – doğru (true), yanlış (false)
 - Operasyonlar (Instructions)
 - ...
-
- Sayılar ile başlayalım...

İşaretsiz (Unsigned) Tamsayılar (Integers)

- İşaretsiz tamsayılar
 - Her zaman pozitif değerler taşırlar
- Örn:

$$\begin{array}{ccc} & 329 & (10' \text{luk tabanda}) \\ & / \quad | \quad \backslash \\ 10^2 & & 10^1 & & 10^0 \end{array}$$

$$3 \times 100 + 2 \times 10 + 9 \times 1 = 329$$

$$\begin{array}{ccc} & 101 & (2' \text{lik tabanda}) \\ & / \quad | \quad \backslash \\ 2^2 & & 2^1 & & 2^0 \end{array}$$

$$1 \times 4 + 0 \times 2 + 1 \times 1 = 5$$

İşaretsiz Tamsayılar

- N-bit'lik bir işaretsiz tamsayı 2^n değer taşımaktadır: 0'dan 2^n-1 'e.

2^2	2^1	2^0	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

İşaretsiz İkilik Taban Aritmetiği

- İkilik taban toplaması (10'luk taban gibi)
 - En sağdan başlayarak toplanır, elde olursa bir sonraki toplama aktarılır.

$\begin{array}{r} 10010 \\ + 1001 \\ \hline 11011 \end{array}$	$\begin{array}{r} \overset{\text{elde}}{\curvearrowright} 10010 \\ + 1011 \\ \hline 11101 \end{array}$	$\begin{array}{r} \overset{\curvearrowright}{\overset{\curvearrowright}{\overset{\curvearrowright}{\overset{\curvearrowright}{1111}}} \\ + 1 \\ \hline 10000 \end{array}$
--	--	---

İşaretli (Signed) Tamsayılar (Integers)

- N adet bit ile, 2^n farklı değer gösterebiliyoruz.
 - 2^n farklı değeri;
 - Yarısını pozitif sayılara, yarısını ise negatif sayılara atayarak işaretli tamsayılar elde edilir.
 - Pozitif sayılar 1'den 2^{n-1}
Negatif sayılar $-(2^{n-1})$ 'den -1'e
 - Örn 3 bitlik bir saklama alanımız var ise;
 - Pozitif sayılar 1'den 4'e kadar, negatif sayılar ise -4'den -1'e kadar olmaktadır.

İşaretli (Signed) Tamsayılar (Integers)

- Örn 3 bitlik bir saklama alanımız var ise;
- Pozitif sayılar 1'den 4'e kadar, negatif sayılar ise -4'den -1'e kadar olmaktadır.
- 0 sayısı da kullanılmak isteniyorsa, ya pozitif yada negatif bölümden bir sayı 0 olarak ifade edilmektedir.

İşaretili (Signed) Tamsayılar (Integers)

- Pozitif tamsayılar
 - İşaretsiz tamsayılar gibidir.
 $00101 = 5$
- Negatif tamsayılar
 - İşaret Biti ile Gösterim – İşaret biti en başa konur, Diğer bitler işaretsiz gösterimdeki gibi yazılmaktadır.
 $10101 = -5$
 - 1'e tümleyen (one's complement) – Her bir biti tersine çevrilir.
 $11010 = -5$
 - Her iki gösterimde de en büyük bit sayının işaretini gösterir:
0=pozitif, 1=negatif

İki'e tümleyen (Two's Complement)

- İşaret biti ile gösterim ve 1'e tümleyen'in sorunları
 - 0 sayısının iki gösterimi olur (+0 and -0)
- İşaret Biti ile Gösterim
 - 00000 = +0
 - 10000 = -0
- 1'e tümleyen olarak gösterim
 - 00000 = +0
 - 11111 = -0

İki'e tümleyen (Two's Complement)

- İşaret biti ile gösterim ve 1'e tümleyen'in sorunları
 - Aritmetik işlemlerin gerekli donanım devreleri çok karmaşıktır.

- İşaret biti ile gösterimli toplamadaki sorun

$$\begin{array}{r} 1011 \quad (-3) \\ + \quad 0010 \quad (2) \\ \hline 1101 \quad (-5) \rightarrow \text{Yanlış} \end{array}$$

- Çözüm için toplamadan önce hangisi büyük olduğuna bakılıp büyükten küçük çıkartılıp, büyük sayının işareti verilmesi gerekmektedir.
- Dolayısıyla gerekli toplama işlemini yapacak gerekli donanımda büyük küçük kontrolü yapacak bir devre, çıkartıcı ve işaret bitini tekrar yerleştirici bulunması gerekmektedir. Yani donanım karmaşık ve büyük olmaktadır.

İki'e tümleyen (Two's Complement)

- İfade edilmek istenen değer 0 veya pozitif ise,
 - İşaretsiz tam sayılardaki gibi yazılır, en büyük bitler 0 ile doldurulur.
- Eğer sayı negatif ise,
 - Pozitif sayı gibi yazılır
 - Her bir bitin tersi alınır (1'e tümleyen)
 - Sonuca 1 eklenir.

$$\begin{array}{r}
 \text{↻} \quad 00101 \quad (5) \\
 \quad \quad 11010 \quad (1'e \text{ tümleyen}) \\
 + \quad \quad \quad 1 \\
 \hline
 \quad \quad 11011 \quad (-5)
 \end{array}$$

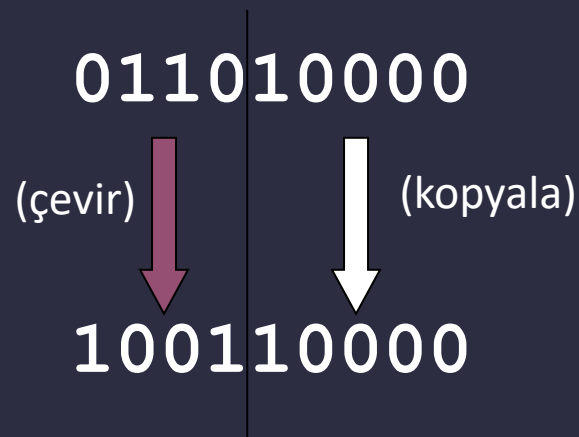
$$\begin{array}{r}
 \text{↻} \quad 01001 \quad (9) \\
 \quad \quad 10110 \quad (1'e \text{ tümleyen}) \\
 + \quad \quad \quad 1 \\
 \hline
 \quad \quad 10111 \quad (-9)
 \end{array}$$

İki'e tümleyen (Two's Complement)

- 2'e tümleyen'in bulunmasının kısayolu:
 - Sayının bitlerini sağdan sola ilk "1" i görene kadar kopyala
 - Kalan bitleri ters çevir

$$\begin{array}{r}
 011010000 \\
 100101111 \\
 + \quad \quad 1 \\
 \hline
 100110000
 \end{array}$$

(1'e tümleyen)



İki'ye tümleyen (Two's Complement)

- En büyük bit işaret biti ve ağırlığı -2^{n-1} dir.
- N adet bit ile -2^{n-1} den $2^{n-1} - 1$ sayısına kadar ifade edilebilir.
 - En küçük negatif sayının (-2^{n-1}) pozitif karşılığı yoktur.

-2^3	2^2	2^1	2^0		-2^3	2^2	2^1	2^0	
0	0	0	0	0	1	0	0	0	-8
0	0	0	1	1	1	0	0	1	-7
0	0	1	0	2	1	0	1	0	-6
0	0	1	1	3	1	0	1	1	-5
0	1	0	0	4	1	1	0	0	-4
0	1	0	1	5	1	1	0	1	-3
0	1	1	0	6	1	1	1	0	-2
0	1	1	1	7	1	1	1	1	-1

İkilik tümleyen'i 10'luk tabana çevirme

1. En büyük bit (en soldaki) 1 ise, sayının ikilik bütünleyenini alıp pozitif değerini bulun.
2. En sağ bittten başlayarak 2'nin kuvvetleri ile çarparak değerleri toplayın.
3. Eğer işleme başlarken sayı negatif ise (yani en soldaki biti 1 ise) çıkan 10'luk tabandaki sayıya – işareti koyun

n	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

$$\begin{aligned} X &= 01101000_{\text{ikilik}} \\ &= 2^6 + 2^5 + 2^3 = 64 + 32 + 8 \\ &= 104_{\text{onluk}} \end{aligned}$$

İkilik tümleyen'i 10'luk tabana çevirme

$$\begin{aligned} X &= 00100111_{\text{ikilik}} \\ &= 2^5 + 2^2 + 2^1 + 2^0 = 32 + 4 + 2 + 1 \\ &= 39_{\text{onluk}} \end{aligned}$$

$$\begin{aligned} X &= 11100110_{\text{ikilik}} \\ -X &= 00011010 \\ &= 2^4 + 2^3 + 2^1 = 16 + 8 + 2 \\ &= 26_{\text{onluk}} \\ X &= -26_{\text{onluk}} \end{aligned}$$

n	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

Onluk Tabandan İkilik Tömleyen'e Dönüşüm

- Birinci Yöntem: *Bölüm*

1. Onluk tabandaki sayının mutlak değerini al. (Her zaman pozitif olmalı.)
2. İki'e böl – kalan en küçük bit olmaktadır.
3. 0 bulana kadar bölme işlemine devam et, bölümlerden elde edilen kalanları sağdan sola olacak şekilde yaz.
4. Sayının bit genişliği kadar en sağa 0 ekle. (aşağıdaki örnekte sayının 8 bitlik olduğu varsayılmıştır)
Onluk tabandaki sayı negatif ise, çıkan sayının ikilik bütünleyenini al.

$$X = 104_{\text{onluk}}$$

$$104/2 = 52 \text{ k}0 \quad \textit{bit 0}$$

$$52/2 = 26 \text{ k}0 \quad \textit{bit 1}$$

$$26/2 = 13 \text{ k}0 \quad \textit{bit 2}$$

$$13/2 = 6 \text{ k}1 \quad \textit{bit 3}$$

$$6/2 = 3 \text{ k}0 \quad \textit{bit 4}$$

$$3/2 = 1 \text{ k}1 \quad \textit{bit 5}$$

$$X = 01101000_{\text{ikilik}}$$

$$1/2 = 0 \text{ k}1 \quad \textit{bit 6}$$

Onluk Tabandan İkilik Tümleyen'e Dönüşüm

- İkinci Yöntem: *2'nin kuvvetlerini çıkartmak*

1. Onluk tabandaki sayının mutlak değerini al.
2. Sayıdan, 2'nin kuvvetlerinden sayıya eşit veya daha küçük olan sayıyı çıkart.
3. İlgili yere 1 yerleştir.
4. 0 elde edene kadar devam et.
5. En büyük biti, bit genişliği kadar 0 ile doldur;
Onluk sayı negatif ise, çıkan ikilik tabandaki sayının ikilik bütünleyenini al

n	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

Onluk Tabandan İkilik Tümlleyen'e Dönüşüm

$$X = 104_{\text{onluk}}$$

$$104 - 64 = 40 \quad \text{bit 6}$$

$$40 - 32 = 8 \quad \text{bit 5}$$

$$8 - 8 = 0 \quad \text{bit 3}$$

$$X = 01101000_{\text{ikilik}}$$

n	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

Toplama

- İkilik tümleyen sayıların toplama işlemi, işaretsiz sayıların toplama işlemi gibidir. Herhangi bir kontrol mekanizmasına ihtiyaç duyulmaz.
 - En büyük bitten elde edilecek elde biti atılır.

$$\begin{array}{r} 01101000 \quad (104) \\ + 11110000 \quad (-16) \\ \hline 01011000 \quad (88) \end{array}$$

Çıkarma

- İkinci sayının negatif halini bul ve topla.
 - İkinci sayının ikilik tümleyenini bul ve ilk sayı ile topla

$$\begin{array}{r} 01101000 \quad (104) \\ - 00010000 \quad (16) \\ \hline 01101000 \quad (104) \\ + 11110000 \quad (-16) \\ \hline 01011000 \quad (88) \end{array}$$

İşaret Uzantısı (Sign Extension)

- İki sayı toplanırken, her iki sayının aynı bit genişliğinde olması gerekmektedir.
- İki sayıyı aynı bit genişliğine getirmek için sadece en soluna 0 eklersek;

<u>4-bit</u>	<u>8-bit</u>
0100 (4)	00000100 (halen 4)
1100 (-4)	00001100 (12, -4 değil)

- Doğru hesaplama için, sayının işaret biti, genişletilecek yerlere yerleştirilir.

<u>4-bit</u>	<u>8-bit</u>
0100 (4)	00000100 (halen 4)
1100 (-4)	11111100 (halen -4)

Taşma (Overflow)

- Sayılar çok büyük olduğunda, toplam n bitlik sayılar ile ifade edilemeyecek kadar büyük çıkabilir.

$$\begin{array}{r} 01000 \quad (8) \\ + 01001 \quad (9) \\ \hline 10001 \quad (-15) \end{array} \qquad \begin{array}{r} 11000 \quad (-8) \\ + 10111 \quad (-9) \\ \hline 01111 \quad (+15) \end{array}$$

- Taşma durumu:
 - Her iki sayının işareti aynı olduğu toplama işlemlerinde olabilir.

Mantık (Logic) Operasyonlar

- Operasyonlar, Doğru (True) veya Yanlış (False) olarak hesaplanmaktadır.
 - İki durum bulunmaktadır, Doğru=1, Yanlış=0

A	B	A Ve B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A Veya B
0	0	0
0	1	1
1	0	1
1	1	1

A	Değil A
0	1
1	0

Mantık Operasyon Örnekleri

- **Ve**

- Bitleri silmek için kullanışlıdır.
 - 0 ile Ve işlemi = 0
 - 1 ile Ve işlemi = değişiklik yok

$$\begin{array}{r} 11000101 \\ \text{Ve} \quad \underline{00001111} \\ 00000101 \end{array}$$

- **Veya**

- Ayarlama için kullanışlıdır.
 - 0 ile Veya işlemi = değişiklik yok
 - 1 ile Veya işlemi = 1

$$\begin{array}{r} 11000101 \\ \text{Veya} \quad \underline{00001111} \\ 11001111 \end{array}$$

- **NOT**

- Her biti değiştirmektedir.

$$\begin{array}{r} \text{Değil} \quad \underline{11000101} \\ 00111010 \end{array}$$

Onaltılık (Hexadecimal) Gösterim

- Bilgisayar üzerinde sıklıkla kullanılan 16'lık gösterim formatıdır.
 - İkilik tabandaki bir sayının her bir 4 biti, bir onaltılık gösterimi ifade eder.
 - Uzun 0 ve 1'ler kullanmaya göre, daha az hata yapmayı sağlar.

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

İkilik Tabandan Onaltılık Tabana Geçiş

- Her bir 4 bit, bir onaltılık'a eşittir.
 - En sağdan başlayarak gruplanır.

011101010001111010011010111

↓ ↓ ↓ ↓ ↓ ↓ ↓

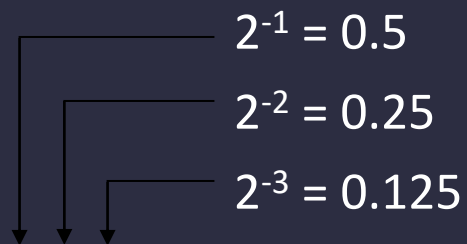
3 A 8 F 4 D 7

Ondalıklı Sayılar: Sabit Nokta (Fixed-Point) Gösterimi

- Ondalıklı sayılar nasıl ifade edilir?
 - Tam sayılar ve küsüratların ayrıldığı bir nokta seçilir.
 - İkilik tümleyen formatında sayıların toplama ve çıkarma işlemleri yapılır.

$$\begin{array}{r}
 00101000.101 \quad (40.625) \\
 + 11111110.110 \quad (-1.25) \\
 \hline
 00100111.011 \quad (39.375)
 \end{array}$$

$2^{-1} = 0.5$
 $2^{-2} = 0.25$
 $2^{-3} = 0.125$



Çok Büyük ve Çok Küçük Sayılar: Kayan Nokta (Floating-Point)

- Çok büyük sayılar: 6.023×10^{23} -- 79 bits gerektirir
- Çok küçük sayılar: 6.626×10^{-34} -- 110 bits gerektirir
- “scientific notation” denen: kesir 2^{Kuvvet}
- Kesir (*fraction*), kuvvet (*exponent*), ve işaret (i) olarak ifade edilmektedir.
- IEEE 754 Kayan Nokta Gösterimi (32-bits):




$$\text{Sayı} = (-1)^i \times 1.\text{kesir} \times 2^{\text{kuvvet}-127}, 1 \leq \text{kuvvet} \leq 254$$

$$\text{Sayı} = (-1)^i \times 0.\text{kesir} \times 2^{-126}, \text{kuvvet} = 0$$

Kayan Nokta Örneği

- IEEE kayan nokta gösterimi

- $10111111010000000000000000000000$

↑ ↑ ↑
İşaret *Kuvvet* *Kesir*

- İşaret biti 1, yani sayı negatif
- Kuvvet: $01111110 = 126$.
- Kesir: $0.100000000000... = 0.5$.

- Değer = $-1.5 \times 2^{(126-127)} = -1.5 \times 2^{-1} = -0.75$.

Metinler: ASCII Karakterler

- ASCII tablosu 8 bitlik bir tablodur. 0-255 arasında her bir sayının karşılığı bir karakter veya kontrol sinyali bulunmaktadır.

00	nul	10	dle	20	sp	30	0	40	@	50	P	60	`	70	p
01	soh	11	dc1	21	!	31	1	41	A	51	Q	61	a	71	q
02	stx	12	dc2	22	"	32	2	42	B	52	R	62	b	72	r
03	etx	13	dc3	23	#	33	3	43	C	53	S	63	c	73	s
04	eot	14	dc4	24	\$	34	4	44	D	54	T	64	d	74	t
05	enq	15	nak	25	%	35	5	45	E	55	U	65	e	75	u
06	ack	16	syn	26	&	36	6	46	F	56	V	66	f	76	v
07	bel	17	etb	27	'	37	7	47	G	57	W	67	g	77	w
08	bs	18	can	28	(38	8	48	H	58	X	68	h	78	x
09	ht	19	em	29)	39	9	49	I	59	Y	69	i	79	y
0a	nl	1a	sub	2a	*	3a	:	4a	J	5a	Z	6a	j	7a	z
0b	vt	1b	esc	2b	+	3b	;	4b	K	5b	[6b	k	7b	{
0c	np	1c	fs	2c	,	3c	<	4c	L	5c	\	6c	l	7c	
0d	cr	1d	gs	2d	-	3d	=	4d	M	5d]	6d	m	7d	}
0e	so	1e	rs	2e	.	3e	>	4e	N	5e	^	6e	n	7e	~
0f	si	1f	us	2f	/	3f	?	4f	O	5f	_	6f	o	7f	del

Diğer Veri Türleri

- **Metinler**
 - Karakterlerin ardışık olarak yazılmasıyla oluştururlar
- **Görüntü**
 - Piksellerin bir araya gelmesi ile oluşurlar
 - Siyah Beyaz: Bir bitlik (1/0 = siyah/beyaz)
 - Renkli: Kırmızı, Mavi, Yeşil (RGB) bileşenleri bulunur. Her biri 8 bitlik sayılar olarak saklanmaktadır.
- **Ses**
 - Genellikle sabit nokta gösterimli sayıların ardışık olarak kaydedilmesi olarak gösterilmektedir.