

## 2. Sayı Sistemleri

Bilgisayar elektronik bir cihaz olduğu için elektrik akımının geçirilmesi (1) yada geçirilmemesi (0) durumlarını işleyebilir. Bu nedenle ikili sayı sistemini temel olarak veri işler ve saklarlar.

En küçük bellek birimi sadece 0 ve 1 değerlerini alabilen ikili sayı sisteminde bir basamağa denk gelen Bit'tir.

1 Bayt (Byte) = 8 Bit

1024 Byte = 1 KiloByte (KB)

1024 KB = 1 MegaByte (MB)

1024 MB = 1 GigaByte (GB)

1024 GB = 1 TeraByte (TB)

Neden 1000 değil de 1024

Bilgisayar 2'li sayı sistemine göre çalıştığı için  $10^3$  değil  $2^{10}$  sayma sistemi temel alınır.

### gelecekte kullanılacak birimler için öneriler

8 Bit	= 1 Byte
1024 Byte	= 1 Kilobyte
1024 Kilobyte KB	= 1 Megabyte
1024 Megabyte MB	= 1 Gigabyte
1024 Gigabyte GB	= 1 Terabyte
1024 Terabyte TB	= 1 Petabyte
1024 Petabyte PB	= 1 Eksabyte
1024 Eksabyte EB	= 1 Zettabyte
1024 Zettabyte ZB	= 1 Yottabyte
1024 Yottabyte YB	= 1 Brontobyte
1024 Brontobyte	= 1 Geopbyte
1024 Geopbyte	= 1 Saganbyte
1024 Saganbyte	= 1 Epicbyte

### POZİTİF VE NEGATİF GERÇEL SAYILAR

Bir byte'lık en küçük ve en büyük pozitif sayılar

00000000 (decimal 0)

11111111 (decimal 255)

Buradaki tüm sayılar, pozitiftir. Bir başka deyişle sayı işaretsizdir. Negatif sayılar söz konusu olduğunda bu sayıların yarısının pozitif, yarısının negatif olur

Örneğin 1 byte'lıksayı -127 ile +127 arasında değişecektir.

İkili sistemde negatif sayılar, çıkarma işleminin toplama aracılığıyla yapılabilmesini sağlamak amacıyla tümleyen sayılarla gösterilir.

Tümleyen sayı, verilen sayıyı, o bit sayısı için temsil edilen en büyük sayıya tamamlayan sayıdır. (Pratikte bit evirerek yapılır.)

Örneğin 00001010 ın tümleyeni 11110101 dir  $(255 - 10)_{10}$ . Bu türden tümleyene "1'e tümleyen sayı" denir. En büyük ağırlıklı (en soldaki) bit negatif sayılar için 1 olarak rezerve edilir.

### İkili Sayı Sistemi

Sayısal bilgisayarlar ikili (Binary) sayıları kullanırlar. İkili ya da iki tabanına göre olan sayı sistemi sadece 0 ve 1 sembollerinden oluşur. Bu sembolere bit (Binary digit kelimelerinden BI ve T kısımlarından türetilmiştir.) adı verilir. Bilgisayarın elektronik devrelerinde 0 bit'i düşük voltajla, 1 bit'i ise yüksek voltajla ifade edilir. 8 bitlik bellek birimine 1 byte denir.

Genel olarak herhangi bir sayı sisteminin yazılış şekli ve decimal sistemdeki karşılığı aşağıdaki gibidir.

R = radix (kök) m=magnitute (büyüklük-burada üs) a= sayı göstermek kaydıyla, R tabanlı bir sayı

$$a_m R^m + a_{m-1} R^{m-1} + \dots + a_0 R^0 + a_{-1} R^{-1} + \dots \quad (0 \leq a_m \leq 9)$$

şeklinde ifade edilir. Burada  $a_0 R^0$  'dan sonraki pozisyon (,) ondalık ifadenin başlangıç yeridir.

On tabanlı bir sistemde bir sayının değeri,

$$a_n 10^n + a_{n-1} 10^{n-1} + \dots + a_1 10^1 + a_0 10^0$$

şeklinde ifade edilir. örneğin 1327 sayısının 1 tane 1000 artı 3 tane 100 artı 2 tane 10 artı 7 tane 1'e eşit olduğu bilinmektedir. Bu durum yukarıdaki gösterim şekli kullanılarak;

$$1 \cdot 1000 + 3 \cdot 100 + 2 \cdot 10 + 7 \cdot 1 \text{ veya}$$

$$1 \cdot 10^3 + 3 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0 \text{ gibi ifade edilebilir.}$$

Benzer şekilde 2'li sistemde bir sayının değeri;

$$a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_1 2^1 + a_0 2^0$$

Şeklinde ifade edilebilir. Örneğin 1 0 0 1 ikili sayısının onlu karşılığı;

$$1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 0 + 0 + 1 = 9 \text{ 'dur.}$$

## Onaltılı (Hexadecimal) Sayılar

Tipik bir mikrobilgisayar bellek yeri (location) 1 0 0 1 1 1 1 0 ikili sayısını saklayabilir. 0 ve 1'lerden oluşan bu uzun sayıyı hatırlamak ve klavyeden girmek oldukça zordur ve girişi hatası yapma olasılığını artırmaktadır. 1 0 0 1 1 1 1 0 ikili sayısı onlu bir sayıya çevrilebilir. Çevrildiğinde bu ikili sayının 10 tabanlı 158'e eşit olduğu görülür. Birçok mikrobilgisayar sistemi hatırlama ve 1 0 0 1 1 1 1 0 gibi ikili sayıların girişini kolaylaştırmak için onaltılı (hexadecimal) gösterimi kullanır.

Onaltılı yada 16 tabanlı sayı sistemi 0'da 9'a kadar olan sayıları ve A,B,C,D,E ve F harfleri olmak üzere 16 sembol kullanır.

Tablo 2'de 16'lı sembolün 4 bit'in kombinasyonu ile temsil edildiği görülmektedir.

Buna göre 1 0 0 1 1 1 1 0 ikili sayısı 16'lı sayı sisteminde 9E olarak gösterilebilir. İlgili sayının 1 0 0 1 kısmı 16'lı sistemde 9'a, 1 1 1 0 kısmı ise E'ye eşittir.

- İkili sayı sisteminin okunabilirliğinin zor olması nedeniyle 16'lı sayı sistemi verilerin temsil edilmesinde sıklıkla kullanılır. 16'lı sayı sistemindeki her basamak 4 biti ifade eder ( $2^4 = 16$ ).

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

Bir ikili sayıyı 16'lık sayı sistemine çevirmek için en sağdan başlayarak sayı 4'er bitten oluşan gruplara ayrılır. Daha sonra her 4 bitlik grup 16'lı karşılığı ile değiştirilir. Örneğin 1 1 1 0 1 0 ikili sayısının 16'lı karşılığına çevirmek istediğimizi düşünelim. Bu durumda,

4 bit grup	4 bit grup
0 0 1 1	1 0 1 0
3	A

olduğundan, 1 1 1 0 1 0 ikili sayısının 16'lı karşılığı 3A'dır. Benzer şekilde 1 1 1 1 1 1 sayısının 16'lı karşılığı;

4 bit grup	4 bit grup
0 1 1 1	1 1 1 1
7	F

olduğundan 7F'dir.

## Sekizli (Octal) Sayı Sistemi

Sekizli (Octal) gösterim, onaltılı sistemde olduğu gibi ikili sayıları temsil etmek için kullanılır. Sekizli sayılar 0'dan 7'ye kadar olan sembolleri kullanır ve bu nedenle 8 tabanına göre sayı sistemi denir. Tablo 3'te onlu, sekizli ve ikili sayı sistemlerinin karşılıkları gösterilmektedir.

Verilen bir sayıyı 8'li karşılığına çevirmek için;

1. Sayının en sağından başlayarak sayıyı 3 bitlik gruplara ayırınız.
2. Her 3 bitlik grubu tablo 3'ten yararlanarak 8'li karşılığı ile değiştiriniz.

Örneğin 11111000100 ikili sayısının 8'li karşılığı;

011	111	000	100
3	7	0	4

olduğundan 3704'tür.

$$(200)_{10} = (C8)_{16} = (310)_8 = (11001000)_2$$

## B C D Sayılar (Binary Coded Decimal)

Ondalık sayı sistemindeki sayıların her basamağının 4 bit olarak kodlanmasıdır.

4859 sayısının BCD karşılığı 0100 1000 0101 1001 dir.

BCD kullanıldığında 16 bit ile temsil edilebilecek en büyük sayı 9999'dur. Normalde 16 bit ile  $2^{16}$  farklı sayının (65535'e kadar) temsil edilebileceğini biliyoruz. BCD 4 bit ile temsil edilebilecek 16 farklı durumdan 10 tanesini kullanabildiği için böyle bir kayba neden olmaktadır.

İkili sayılar çevirme kolaylığı bakımından sekizli veya onaltılı gösterim ile temsil edilir. buna karşın 10'lu sisteme çevirim oldukça zordur. Kullanıcıların 10'lu giriş ve çıkış yaptıkları hesap makineleri, oyunlar ve sayısal aletlerde onlu sayıları göstermek için özel bir kod sistemi olan B C D (Binary Coded Decimal= İkili Kodlanmış Onlu) kodu kullanılır.

Onlu	Sekizli	İkili		
		$2^2$	$2^1$	$2^0$
0	0	0	0	0
1	1	0	0	1
2	2	0	1	0
3	3	0	1	1
4	4	1	0	0
5	5	1	0	1
6	6	1	1	0
7	7	1	1	1

Tablodaki 8421, BCD kodundaki 4 bitin basamak değerini ( $2^3=8$ ,  $2^2=4$ ,  $2^1=2$ ,  $2^0=1$ ) göstermektedir. Başka BCD kodlar da kullanılmaktadır. Ancak onlara burada değinilmeyecektir.

3691 onlu sayısının 8421 B C D 'deki karşılına çevirme işlemi aşağıdaki örnekte gösterilmiştir.

**Örnek 1:**

Onlu	3	6	9	1
BCD	0011	0110	1001	0001

Yukarıdaki örnekten görülebileceği gibi her onlu basamağı, doğrudan 4 bitlik B C D karşılığı ile değiştirilmiştir. Bu örnekten 3691 ondalık sayısının 0011 0110 1001 0001 BCD sayısına eşit olduğu görülmektedir.

Örnek 2’de 1000 0000 0111 0010 BCD sayısının onlu karşılığına çevirme işlemi gösterilmektedir.

**Örnek 2: BCD – Ondalık sayı çevrimi**

BCD	1000	0000	0111	0010
Ondalık	8	0	7	2

  

Onlu	BCD			
	$2^3$	$2^2$	$2^1$	$2^0$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Örnekten de görüldüğü gibi her 4 bitlik grup doğrudan onlu karşılıkları ile değiştirilmiş ve bu ikili sayının onlu karşılığı 8072 olarak elde edilmiştir.

$$(73443.38)_{10} = 7 \times 10^4 + 3 \times 10^3 + 4 \times 10^2 + 4 \times 10^1 + 3 \times 10^0 + 3 \times 10^{-1} + 8 \times 10^{-2}$$

$$(11101)_2 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (29)_{10}$$

$$(11011.1011)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$$

$$= 16 + 8 + 0 + 2 + 1 + \frac{1}{2} + 0 + \frac{1}{8} + \frac{1}{16} = (27.6875)_{10}$$

**ASCII (American Standard Code for Information Interchange)**

- 1 bayt (8 bit) veri  $2^8$  yani 256 farklı değer alabilir

$$(00000000)_2 = (0)_{10} \quad (11111111)_2 = (255)_{10}$$

Bilgisayarda metin türü verilerin işlenebilmesi veya saklanabilmesi için büyük ve küçük harflerin, noktalama işaretlerinin ve klavyedeki diğer yardımcı karakterlerin ikili kodlar ile ifade edilmesi gereklidir.

50'li yıllarda her üretici firma farklı bir kodlama kullanırken, 1963 yılında yayınlanan ASCII standardı ile tüm bilgisayarların aynı kodlamayı kullanması sağlanmıştır. İlk etapta oluşturulan 7 bitlik kodlara sahip (128 farklı kod içeren) ASCII tablosuna, sonradan 1 bit daha ilave edilerek 256 kodluk «genişletilmiş ASCII tablosu» oluşturulmuştur

- 128 kodun 33 tanesi ekrana basılmayan, 95 tanesi basılabilen karakterden oluşur.
- 0-31 arasındaki sayılar, yazıcı gibi bazı çevresel aygıtlar için kullanılan denetim karakterlerine atanmıştır.
- Örn: 12 (hex C) form besleme/yeni sayfa (FF: Form Feed) işlevine ayrılmıştır. Bu komut yazıcıya bir sonraki sayfanın başına atlama bilgisi verir.
- Örn: ENTER tuşuna bastığımızda 10 (hex A) ve 13 (hex D) kodları üretilir. 10 satır başı (LF: Line Feed) 13 ise alt satır (CR: Carriage Return) karşılığıdır. İkisi de ekranda görünmez, alt satırın başına gitme işini gerçekleştirir.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	SOH (start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	STX (start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	ETX (end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	EOT (end of transmission)	36	24	044	&#36;	\$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	ENQ (enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	ACK (acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	BEL (bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	BS (backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	VT (vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	CR (carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	SO (shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	SI (shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	DLE (data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	DC1 (device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	DC2 (device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	DC3 (device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	DC4 (device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	CAN (cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	EM (end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	SUB (substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	ESC (escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	FS (file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	
29	1D	035	GS (group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	RS (record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	US (unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

Source: [www.LookupTables.com](http://www.LookupTables.com)

## Unicode

256 karakterlik ASCII kod tablosunda Yunan, Kiril, Arap ve Japon alfabeti gibi farklı alfabelere ait karakterler bulunmamaktadır. Türkçedeki Ş, ş, Ğ, ğ, İ ve ı karakterleri de yoktur. 1990'lı yıllara kadar bu alfabelere özgü farklı yazı tipleri (font) geliştirilerek bu soruna çözüm bulunmuş, 1991 yılında yayınlanan ve her elemanın 16-bit bir kod ile temsil edildiği 65536 farklı eleman içerebilen Unicode (Universal Code: Evrensel Kod) ile tüm dillere ait karakterleri, matematiksel sembolleri, vs. içeren ortak bir kod tablosu oluşturulmuştur.

n bit ile temsil edilebilecek en büyük sayı =  $2^n - 1$ 'dir. Diğer bir ifadeyle

n = 4 ise 15, n = 8 ise 255, n = 16 ise 65535, ... farklı sayı ifade edilebilir

Bilgisayarda sayılar genellikle 1 bayt, 2 bayt, 4 bayt veya 8 bayt gibi büyüklüklerde saklanır.

### Bir Sayıyı Decimal sayı sisteminden 2 li sayı sistemine dönüştürme

$$(175)_{10} = (10101111)_2$$

$$\begin{array}{r}
 175 \quad 2 \\
 -174 \quad 87 \quad 2 \\
 \quad 1 \quad -86 \quad 43 \quad 2 \\
 \quad \quad 1 \quad -42 \quad 21 \quad 2 \\
 \quad \quad \quad 1 \quad -20 \quad 10 \quad 2 \\
 \quad \quad \quad \quad 1 \quad -10 \quad 5 \quad 2 \\
 \quad \quad \quad \quad \quad 0 \quad -4 \quad 2 \quad 2 \\
 \quad \quad \quad \quad \quad \quad 1 \quad -2 \quad 1 \\
 \quad \quad \quad \quad \quad \quad \quad 0
 \end{array}$$

### ikili sayı sisteminde dört işlem

#### toplama

$$(10101)_2 \quad (21)_{10}$$

$$+ (1010)_2 \quad + (10)_{10}$$

$$(11111)_2 \quad (31)_{10}$$

1 0 1 1 1 1 ----- taşınan sayılar

$$(1 1 0 1 1 1)_2 \quad (55)_{10}$$

$$(1 0 0 0 1)_2 \quad (17)_{10}$$

$$+ (1 1 0 1 0)_2 \quad + (26)_{10}$$

$$(1 1 0 0 0 1 0)_2 \quad (98)_{10}$$

#### Çıkartma

$$(100)_2 \quad (4)_{10}$$

$$- (11)_2 \quad - (3)_{10}$$

$$(001)_2 \quad (1)_{10}$$

0 1 ----- önceki haneden alınan borç

$$(1 1 0 1 1 1)_2$$

$$- (1 1 0 1 1)_2$$

$$(1 1 1 0 0)_2$$

## 2'nin tümleyeni metodu ile çıkartma

$$D = M - S = M + (P - S) - P$$

$$P = 2^n_{\text{decimal}} \quad n = M \text{ eksilenindeki hane (digit) sayısı}$$

$$P = (100000\dots 0)_2$$

Önce P-S, M sayısına eklenir, sonra toplamının en soldaki 1 bırakarak P'yi çıkart

Örnek : n = 8      1 0 1 1 1 1 0 1      işlemi yerine

$$S = \underline{\underline{1\ 1\ 0\ 1\ 1\ 1\ 0}} \quad \text{-----} \rightarrow S = 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0$$

$$1\ 0\ 0\ 1\ 1\ 1\ 1 \quad P - S = 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1 + 1 = 1\ 0\ 1\ 0\ 0\ 1\ 0$$

$$1\ 0\ 1\ 1\ 1\ 1\ 0\ 1$$

$$+ \underline{\underline{1\ 0\ 0\ 1\ 0\ 0\ 1\ 0}}$$

$$1\ 1\ 0\ 0\ 1\ 1\ 1\ 1$$

$$n = 6 \quad P = 1\ 0\ 0\ 0\ 0\ 0\ 0 = 1\ 1\ 1\ 1\ 1\ 1 + 1$$

$$S = \underline{\underline{0\ 1\ 1\ 0\ 1\ 1}}$$

$$P - S = 1\ 0\ 0\ 1\ 0\ 0 + 1 = 1\ 0\ 0\ 1\ 0\ 1$$

$$n = 4 \quad P = 1\ 0\ 0\ 0\ 0 = 1\ 1\ 1\ 1 + 1$$

$$S = \underline{\underline{1\ 1\ 0\ 0}}$$

$$P - S = 0\ 0\ 1\ 1 + 1 = 0\ 1\ 0\ 0$$

İkiye tümleyen sayıda tek bir 0 vardır. Sayılar  $-2^{\text{bitsayısı}-1} \dots 0 \dots (2^{\text{bitsayısı}-1} - 1)$  arasında değişir. Örneğin 8 bit için  $-128 \dots 0 \dots 127$ .

İkiye tümleyen sayı aşağıdaki şekilde elde edilir.

$$2^{\text{bitsayısı}} - \text{sayı}$$

Örnek :  $-20$   $-(0001\ 0100)$  sayısını 8 bitlik ikili sayı ile ifade edelim  
 $2^8 - 20 = 256 - 20 = 236 = 1110\ 1100$

Pratik yöntem sayıyı evirip 1 eklemektir.

0001 0100

$$1110\ 1011 + 1 = 1110\ 1100$$

### Çıkarma işlemi

Örneğin  $40 - 20$  yi 2'ye tümleyenle hesaplayalım

$$\begin{array}{r} 0010\ 1000 \\ + 1110\ 1100 \\ \hline \end{array}$$

$$0001\ 0100 = (20 \text{ dec})$$

### Çarpma

$$(101)_2 \quad (5)_{10}$$

$$\times (11)_2 \quad (3)_{10}$$

$$101$$

$$+101$$

$$(1111)_2 \quad (15)_{10}$$

### Bilgisayardaki çarpma işlemi

$$1\ 1\ 0\ 1\ 1\ 0\ 1 = (m) \text{ çarpılan}$$

$$\times \underline{1\ 1\ 1\ 0\ 1} = \text{çarpan}$$

$$1\ 1\ 0\ 1\ 1\ 0\ 1 = m$$

$$+ \underline{1\ 1\ 0\ 1\ 1\ 0\ 1} = m \text{ sayısını 2 hane sola kaydır}$$

$$1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1 = 1. \text{ Kısmi toplam}$$

$$+ \underline{1\ 1\ 0\ 1\ 1\ 0\ 1} = m \text{ sayısını bir hane sola kaydır}$$

$$1\ 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1 = 2. \text{ Kısmi toplam}$$

$$+ \underline{1\ 1\ 0\ 1\ 1\ 0\ 1} = m \text{ sayısını bir hane sola kaydır}$$

$$1\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 1 = \text{tamamlanmış çarpım}$$

### Bölme

$(10111)_2 \div (1100)_2$  (bölünenin ilk 4 rakamı bölenden küçük olduğu için aramayı ilk 5'de yap

$$\underline{-1100} \quad 11 \text{ (bölüm)}$$

$$(46)_{10} \quad \underline{(12)_{10}}$$

10110 0

(3)<sub>10</sub> (bölüm)

- 1100

(1010)<sub>2</sub> (kalan)

(10)<sub>10</sub> (kalan)

1 0 1 0 1 1 1 1 1 . 0	<u>1 1 0 1 0</u>
- <u>1 1 0 1 0</u>	1 1 0 1 . 1
1 0 0 0 1 1	
- <u>1 1 0 1 0</u>	
1 0 0 1 1 1	
- <u>1 1 0 1 0</u>	
1 1 0 1 0	
- <u>1 1 0 1 0</u>	

### 10 tabanlı bir sayıyı 2 tabanına çevirme

SAYI BÖLEN TAM SAYI BÖLME KALANI

46 ÷ 2	0	↑	(46) <sub>10</sub> = (101110) <sub>2</sub>
23	2	1	→
11	2	1	
5	2	1	
2	2	0	
1	2	1	

SAYI ÇARPAN TAM SAYI KALAN KISMI

0,8125 X 2		(0,8125) <sub>10</sub> = (0,1101) <sub>2</sub>
1,6250 X 2	1	→
1,25 X 2	1	

$$0,50 \times 2 = 0$$

$$1,0 \times 2 = 1 \downarrow$$

Bu işlemlerdeki benzer sistematik, tüm sayı sistemleri ile decimal arasındaki dönüşümlerde (R) kök çarpan veya bölen sayısı değiştirilerek kullanılabilir

### **Onaltılı sistem**

$$\begin{aligned}(6AFD)_{16} &= 6 \times 16^3 + A \times 16^2 + F \times 16^2 + D \times 16 \\ &= 6 \times 16^3 + 10 \times 16^2 + 15 \times 16^2 + 13 \times 16 \\ &= (27389)_{10}\end{aligned}$$