

BLM1011

BİLGİSAYAR BİLİMLERİNE

GİRİŞ

GR.2, 3(ONLINE)

2021-2022 GÜZ YARIYILI

DR.ÖĞR.ÜYESİ GÖKSEL BİRİCİK

Sayı Sistemleri

Ondalık (Decimal) Sayı Sistemi

Ondalık sayı sistemi

- Hindu Arabic, Arabic olarak ta bilinir.

10 farklı rakam kullanılır.

- 0, 1, 2, 3, 4, 5, 6, 7, 8 ve 9

Kesirli sayıların gösterimi için **nokta işareti** kullanılır.

Ondalık sayı sisteminde 543.21 sayısı

- $(5 \times 10^2) + (4 \times 10^1) + (3 \times 10^0) + (2 \times 10^{-1}) + (1 \times 10^{-2})$ şeklinde değerlendirilir.

İkili (Binary) Sayı Sistemi

İkili sayı sisteminde sadece 0 ve 1 rakamları kullanılır.

- İki sayısı «10» şeklinde ifade edilir.
- Ondalık sayı sisteminde olduğu gibi toplamın iki olması durumunda bir sonraki haneye aktarılır.

İkili sayı sistemindeki sayıların yazımı **genellikle** ondalık sisteme göre daha uzundur.

- Bunun temel nedeni ikilik sistemde her hanenin onluk sisteme göre daha az bilgi ifade edebilmesidir.
- Bundan dolayı ikilik sistemdeki hanelere **bit** adı verilir. (bit: **binary digit**)

Sekizlik (Octal) Sayı Sistemi

Sekizlik sayı sisteminde sayılar sadece 8 rakam kullanılarak ifade edilir.

- 0, 1, 2, 3, 4, 5, 6 ve 7

Sekizlik sayı sisteminde her hane ikilik sayı sistemindeki 3-biti ifade eder. ($2^3 = 8$)

Sekizli sayı sistemi 12-bit, 24-bit ve 36-bit yapısındaki çeşitli işlemcilerde kullanılmıştır.

- Örnek : PDP-8, ICL 1900 ve IBM mainframe

Onaltılık (Hexadecimal) Sayı Sistemi

Onaltılık sayı sisteminde sayıların ifade edilmesi için 16 değere ihtiyaç vardır.

- Bunun için 10 rakam ve 6 harften yararlanılır.
- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- A, B, C, D, E ve F

Hanelerin kullanımı ondalık sayı sistemi ile aynıdır.

Ondalık sayı sisteminde 256,058 sayısı

- İkilik sayı sisteminde «0011 1110 1000 0011 1010»
- Sekizlik sayı sisteminde «764 072»
- Onaltılık sayı sisteminde «3 E83A» şeklinde yazılır.

Sayı Sistemleri Arasında Geçiş

Matematiksel olarak sayı sistemleri arasındaki geçiş çarpma ve bölme işlemleri ile yapılır.

- Ondalık sayı sisteminde başka sayı sistemine geçerken bölme
- Diğer sayı sistemlerinden Ondalık sisteme geçerken çarpma

$$\begin{array}{r|l} 25 & 2 \\ \hline \underline{24} & 12 \\ 1 & 2 \\ \hline \underline{12} & 6 \\ 0 & 2 \\ \hline \underline{6} & 3 \\ 0 & 2 \\ \hline \underline{2} & 1 \\ 1 & \end{array}$$

$25 = (11001)_2$

Sayı Sistemleri Arasında Geçiş

İkilik, sekizlik ve onaltılık sayı sistemleri arasındaki geçişler daha pratik şekillerde yapılabilir.

Sekizlik sistemdeki her hane, ikilik sistemdeki üç haneye karşılık gelir.

1 000 010 111 100 011

102743

Onaltılık sistemdeki her hane, ikilik sistemde dört haneye karşılık gelir.

1000 0101 1110 0011

85E3

Decimal: 34275

Veri Depolama

Bitler ve Bit Örüntüleri

Bit: Binary Digit (0 – 1)

Bit örüntüleri ile bilgi gösterilir/depolanır.

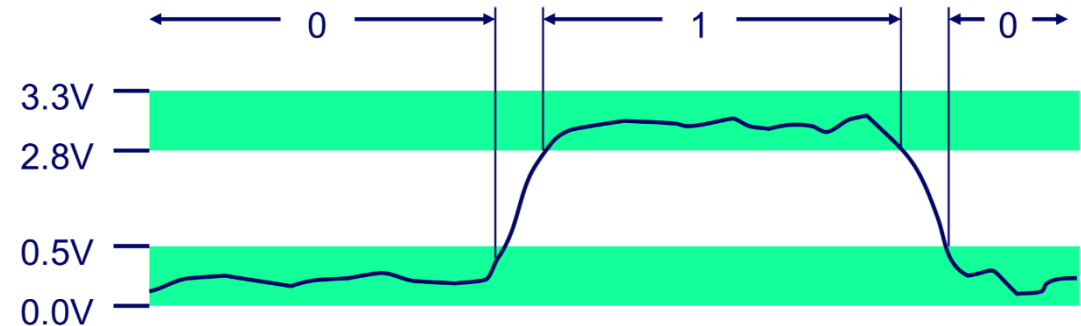
- Sayılar
 - Tamsayılar
 - Kesirli sayılar
- Metin karakterleri
- Görüntüler
- Sesler
- Diğer her şey

Neden ondalık sayı sistemi değil ?

- ENIAC ondalık sistemi kullanıyordu!
- Daha fazla sinyal seviyesi ihtiyacı hassasiyet problemi yaratır.
- Toplama, çarpma vb. işlemlerin gerçekleştirilmesi zorlaşır.

İkili sistemde bilginin aktarımı daha kolay!

- parazit, gürültülere karşı daha dayanıklı



Mantıksal (Boolean) Operatörler

0: Yok / Yanlış / False

1: Var / Doğru / True

Mantıksal AND (AND)

Mantıksal OR (OR)

Mantıksal XOR (Exclusive OR)

Mantıksal Negation (NOT)

The AND operation

$$\begin{array}{r} 0 \\ \text{AND } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \text{AND } 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \text{AND } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \text{AND } 1 \\ \hline 1 \end{array}$$

The OR operation

$$\begin{array}{r} 0 \\ \text{OR } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \text{OR } 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{OR } 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{OR } 1 \\ \hline 1 \end{array}$$

The XOR operation

$$\begin{array}{r} 0 \\ \text{XOR } 0 \\ \hline 0 \end{array}$$

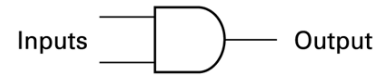
$$\begin{array}{r} 0 \\ \text{XOR } 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{XOR } 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{XOR } 1 \\ \hline 0 \end{array}$$

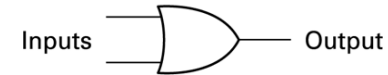
Mantıksal Operatörlerin Gerçeklenmesi

AND



Inputs		Output
0	0	0
0	1	0
1	0	0
1	1	1

OR



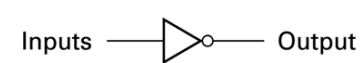
Inputs		Output
0	0	0
0	1	1
1	0	1
1	1	1

XOR



Inputs		Output
0	0	0
0	1	1
1	0	1
1	1	0

NOT

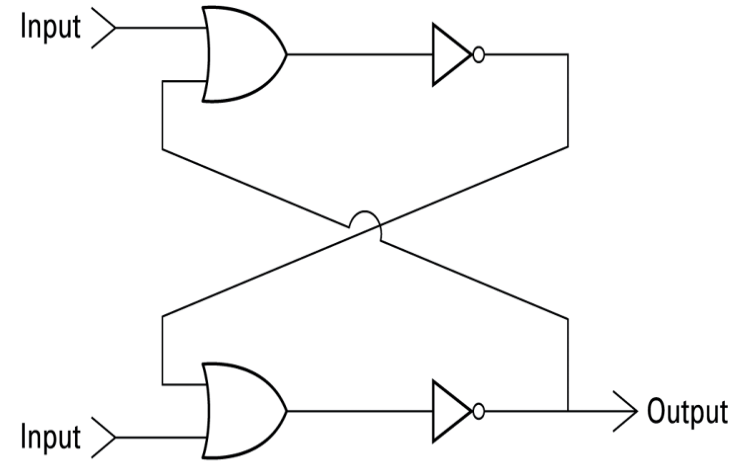
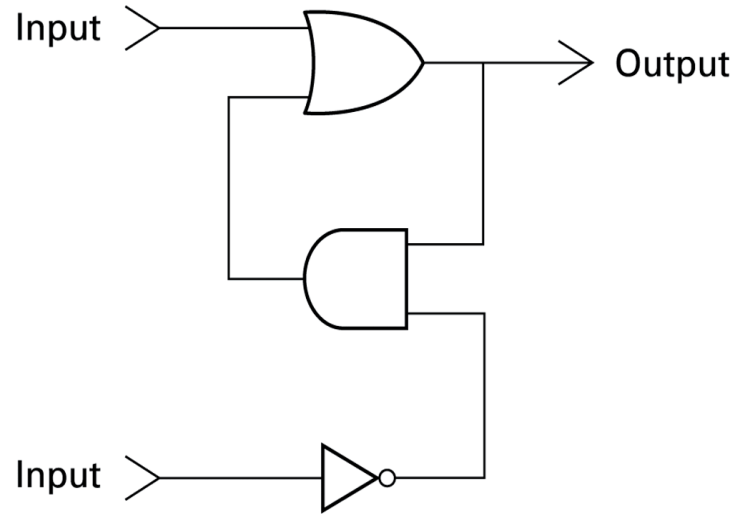


Inputs	Output
0	1
1	0

Flip-Flop

Mantıksal kapılar kullanılarak gerçekleştirilen, 1 bit veri depolayabilen mantıksal devre

- Bir giriş ile 1 depolanır
- Bir diğer giriş ile 0 depolanır
- İki giriş de 0 ise, son değer muhafaza edilir



AND – OR – XOR Örnekler

A sayısının 60, B sayısının 13 olması durumunda

- A = 0011 1100
- B = 0000 1101

A & B = ?

- **0000 1100**

A | B = ?

- **0011 1101**

A XOR B = ?

- **0011 0001**

Aritmetik Operatörler – İlişkisel Operatörler

Toplama (+)

Çıkarma (-)

Çarpma (*)

Bölme (/)

Kalan (mod) (%)

Büyüktür (>)

Küçüktür (<)

Büyük eşit (>=)

Küçük eşit (<=)

Eşit (=)

Eğit değil (≠)

Bit İşlem (Bitwise) Operatörler

AND (&)

OR (|)

Exclusive OR (XOR)

Shift

- Sol (<<)
- Sağ (>>)

Rotate

- Sol / Sağ

Shift Operatörü

A sayısı halen 60 değerine sahip 😊

- **0011 1100**

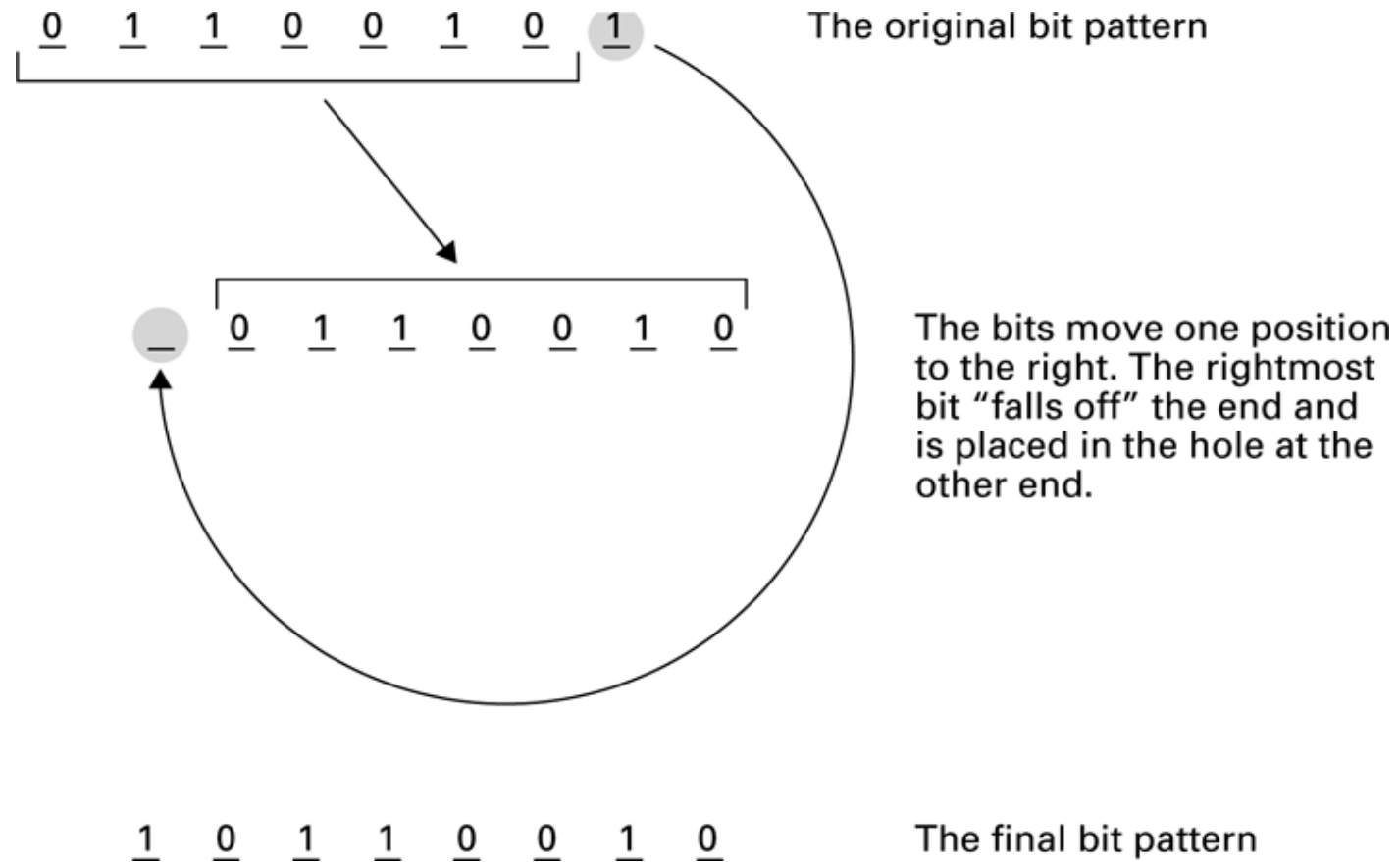
Sola doğru shift işlemi **iki kere** gerçekleştirildiğinde

- $A \ll 2$
- **240 – 1111 0000**

Sağa doğru shift işlemi **üç kere** gerçekleştirildiğinde

- $A \gg 3$
- **7 – 0000 0111**

Rotate Operatörü



Bellek Organizasyonu

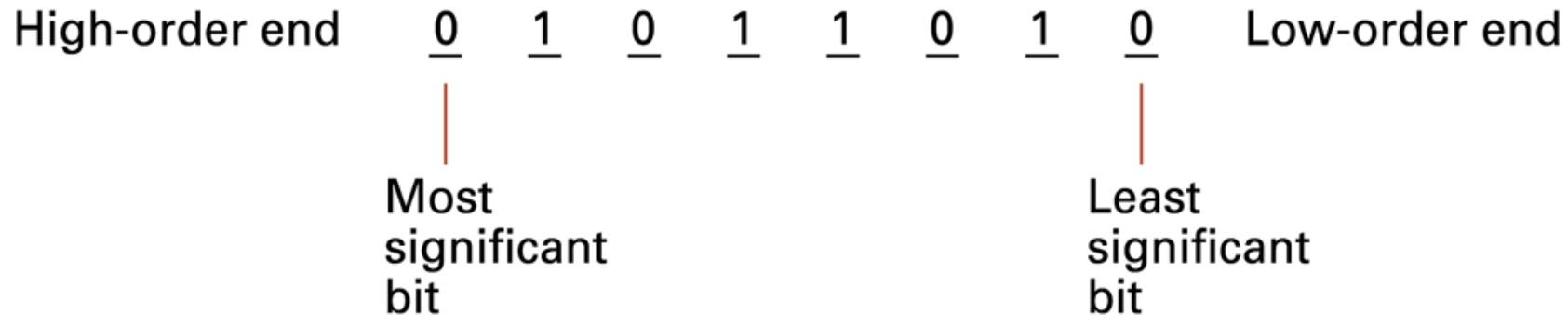
Ana Bellek Hücreleri

Hücre: Bellekteki bir birim depolama alanı.

Genellikle 8 bit: 1 byte 16 bit, 32 bit, 64 bit de olabilir.

MSB: hücredeki bitlerin kavramsal dizilişinde en soldaki (en yüksek mertebeli) bit

LSB: hücredeki bitlerin kavramsal dizilişinde en sağdaki (en düşük mertebeli) bit



Little Endian / Big Endian

Endian verinin belleğe yerleşiminin nasıl yapılacağını belirler.

0x0001020304050607

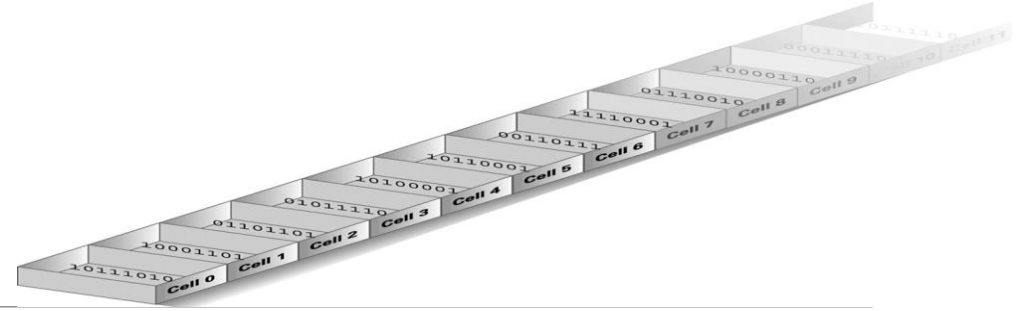
00	01	02	03	04	05	06	07
a	a+1	a+2	a+3	a+4	a+5	a+6	a+7

BIG ENDIAN

LITTLE ENDIAN

07	06	05	04	03	03	02	01
a	a+1	a+2	a+3	a+4	a+5	a+6	a+7

Ana Bellek Adresleme



Adres: Bir hücreye erişim için kullanılan tanım

- Sayılardan oluşur.
- Sıfırdan başlar

İşlemciler aritmetik lojik işlemleri farklı boyutlardaki bilgiler üzerinde gerçekleştirebilir

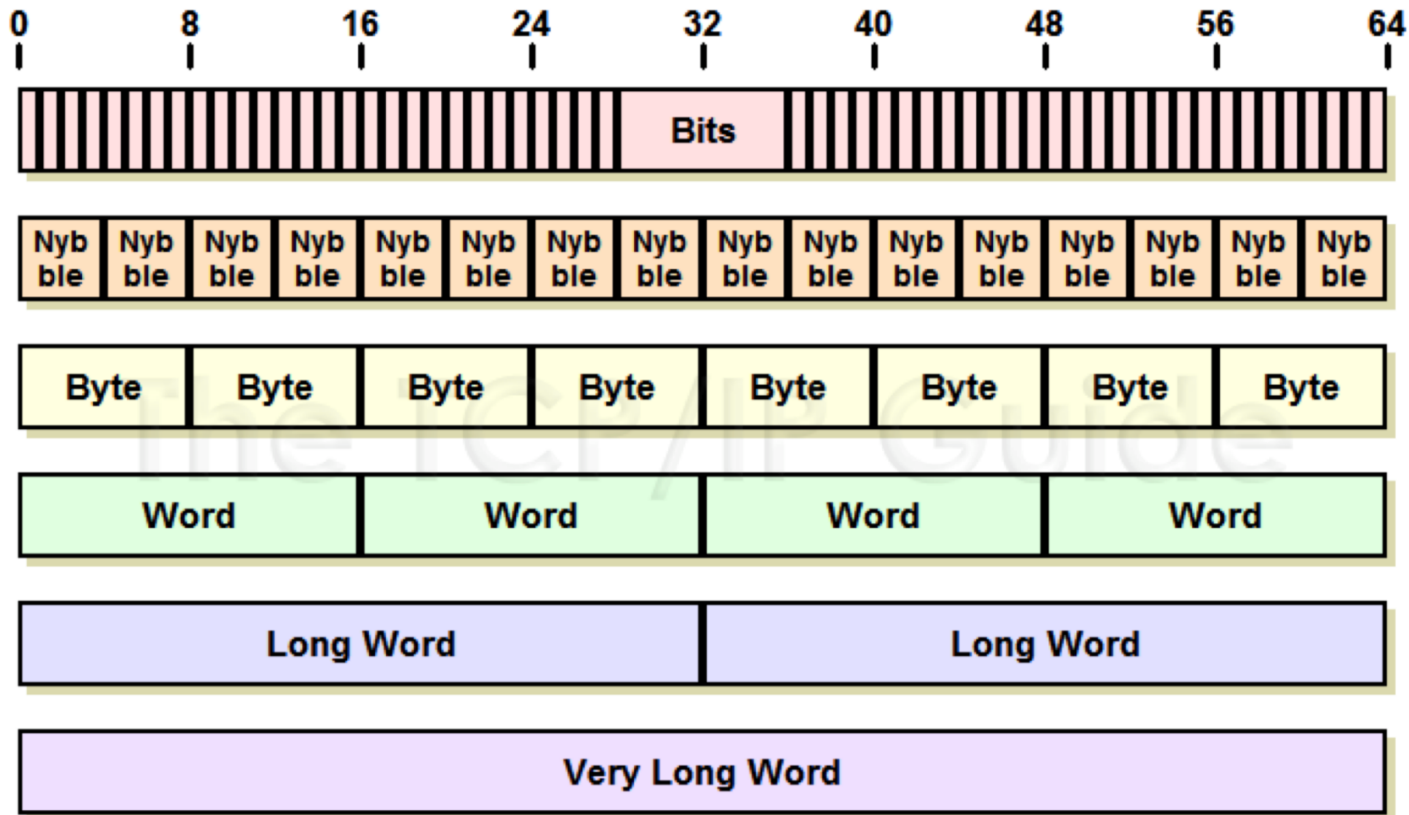
- İşlemcinin tek seferde işleyebildiği bilgi boyutu bit cinsinden ifade edilir
- 16-bit, 32-bit, 64-bit
- Zaman zaman adres yolu ile veri yolu birbirinden farklı boyutta olan işlemciler de olabilir

Günümüzde birçok kişisel bilgisayar 64-bit üzerinde işlem yapar.

- 32-bit üzerinde işlem yapan bilgisayarların bellek adresleme kapasitesi 4GB ile sınırlıdır.

32-bit words	64-bit words	bytes	addr.
Addr: 0000	Addr: 0000		0000
			0001
			0002
			0003
Addr: 0004	Addr: 0000		0004
			0005
			0006
			0007
Addr: 0008	Addr: 0008		0008
			0009
			0010
			0011
			0012
			0013
			0014
			0015
Addr: 0012			

Bit, Byte, Word, vb.



Bilginin Eşlenmesi

Bilginin Eşlenmesi – Metin

Herhangi bir veri tipinin bellekte tutulması için bir eşleme (mapping) işlemi yapılmalı

- Aynı veri tipi için farklı eşleme yapıları bulunabilir.

Her bir karakter, noktalama vs sembol belirli bir bit örüntüsü ile ifade edilir.

ASCII: – American Standard Code for Information Interchange

- 7-bit ile (128 karakter) İngilizce ve pek çok noktalama işaret kodlanmıştır.

ISO: 8-bit uzantılar ile ASCII tablosunu ana dil ailelerine göre uyarlamıştır.

UTF: Unicode Transformation Format: 16-bit örüntüler ile çok çeşitli semboller temsil edilir.

01001000	01100101	01101100	01101100	01101111	00101110
H	e	l	l	o	.

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

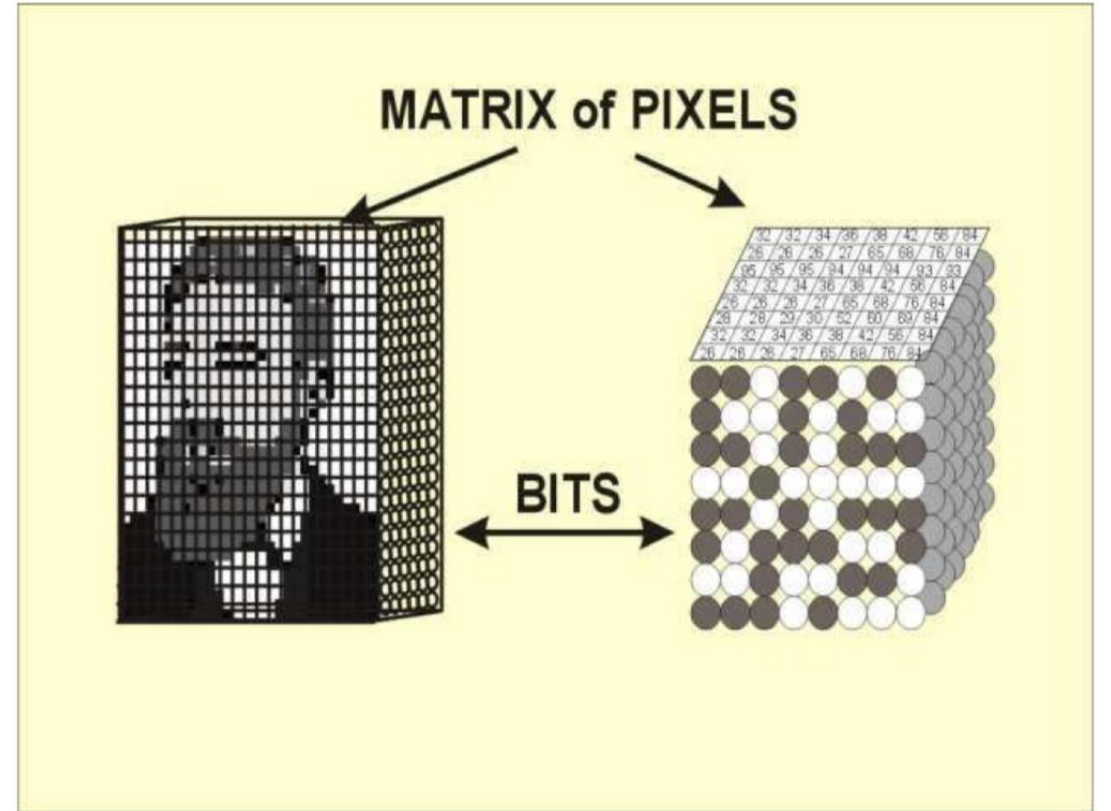
Bilginin Eşlenmesi - Görüntü

Bit eşleme teknikleri

- Piksel: «Picture Cell»
- RGB, YCbCr, CMYK, HSV, ...renk uzayları

Vektör teknikleri

- Ölçeklenebilir
- SVG, PostScript, ...



Bilginin Eşlenmesi - Sayılar

İkili gösterim (Binary Notation): İkilik tabanda bitler ile sayılar temsil edilir.

Sayısal değerlerin bilgisayarda temsilinde kısıtlar olabilir.

- Taşma (Overflow): Bilgi depolama alanına sığmadığında oluşur
- Kesme (Truncation): Bilgi tam kesinlik ile temsil edilemez

Bilginin Eşlenmesi – Negatif Sayılar

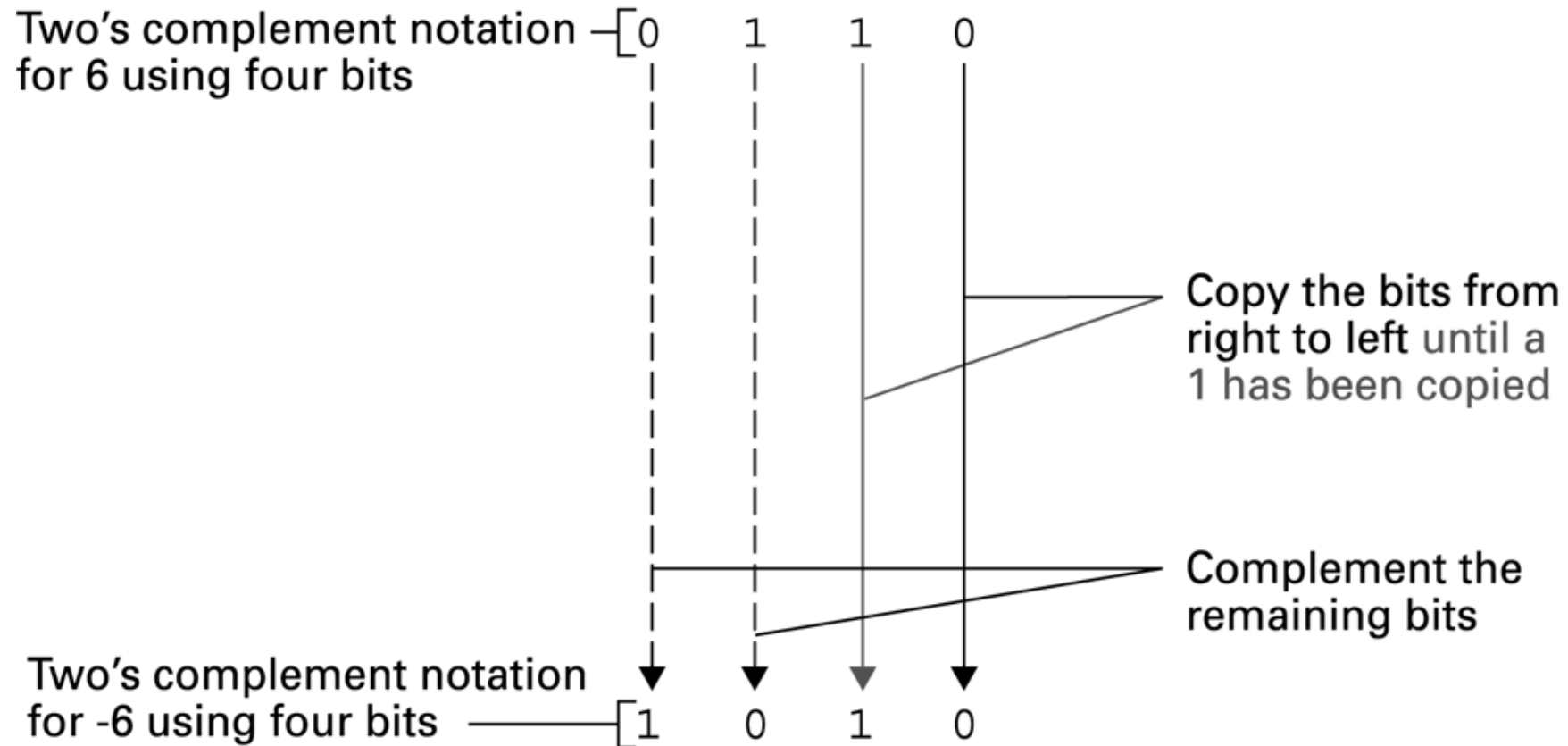
2'nin tümleyeni alınarak negatif sayılar ifade edilir. (2's complement)

- 1's complement
 - En yüksek anlamlı bit (Most Significant Bit, MSB) işaret (sign) biti olarak kullanılır.
 - 0 : pozitif sayı
 - 1 : negatif sayı
 - Sayının ikilik sistemdeki yazımının her bitin 1'e göre tersi alınır.
 - 11 : 0000 1011
 - -11 : 1111 0100
- 2's complement
 - 1'e göre ters alma işleminde sıfır için iki farklı değer üretilir. 2'ye göre ters alma işleminde ise bu problem yoktur.
 - 1'e göre ters alma işleminden sonra sayıya 1 eklenir.

Alternatif: Excess Eight

Bilginin Eşlenmesi – Negatif Sayılar

2's complement alternatif dönüşüm:



2's complement 3-8 – Excess 3-8

a. Using patterns of length three

Bit pattern	Value represented
011	3
010	2
001	1
000	0
111	-1
110	-2
101	-3
100	-4

b. Using patterns of length four

Bit pattern	Value represented
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

Bit pattern	Value represented
111	3
110	2
101	1
100	0
011	-1
010	-2
001	-3
000	-4

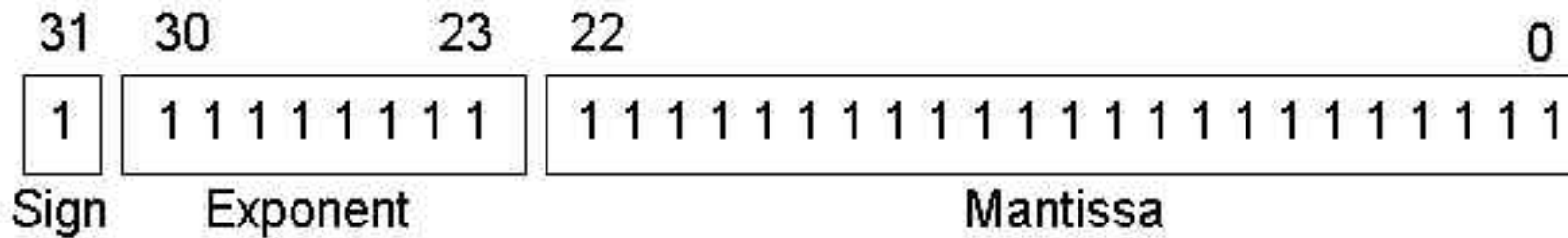
Bit pattern	Value represented
1111	7
1110	6
1101	5
1100	4
1011	3
1010	2
1001	1
1000	0
0111	-1
0110	-2
0101	-3
0100	-4
0011	-5
0010	-6
0001	-7
0000	-8

Bilginin Eşlenmesi – Kesirli Sayılar

- IEEE Standard 754: Floating Point Numbers
- Single/Double precision

Her iki formatta kullanılan yöntem benzerdir.

- Kesirli sayının ifade edilmesi için kullanılacak 32-bit veya 64-bit uzunluğundaki alan **Sign**, **Exponent** ve **Mantissa** olarak adlandırılan 3 parçaya bölünür.



Bilginin Eşlenmesi – Kesirli Sayılar

İşaret (Sign) Bit

- 0 pozitif bir sayıyı, 1 ise negatif bir sayıyı ifade eder.

Üs (Exponent) Bit'leri

- Hem pozitif hem de negatif üs bilgisinin ifade edilebilmesi için biased notation adı verilen yöntem kullanılır. IEEE standardında single precision için bu değer 127, double precision için ise 1023'tür.

Ondalıklı (Mantissa) bitler

- Normalizasyon yapılmış olarak saklanır.
- İkili sistemde yapılan normalizasyon bir bit kazandıracaktır!

Bilginin Eşlenmesi – Kesirli Sayılar

01101011 → 0 110 1011

İşaret biti: 0

Üs: 110 → 2 (Excess-3)

Mantis: 1011

.1011 → $10.11 = 2 + \frac{1}{2} + \frac{1}{4} = 2 \frac{3}{4}$

00111100 → 0 011 1100

İşaret biti: 0

Üs: 011 → -1 (Excess-3)

Mantis: 1100

.1100 → $.01100 = 0 + \frac{1}{4} + \frac{1}{8} = \frac{3}{8}$

Bilginin Eşlenmesi – Kesirli Sayılar

1 1/8 ?

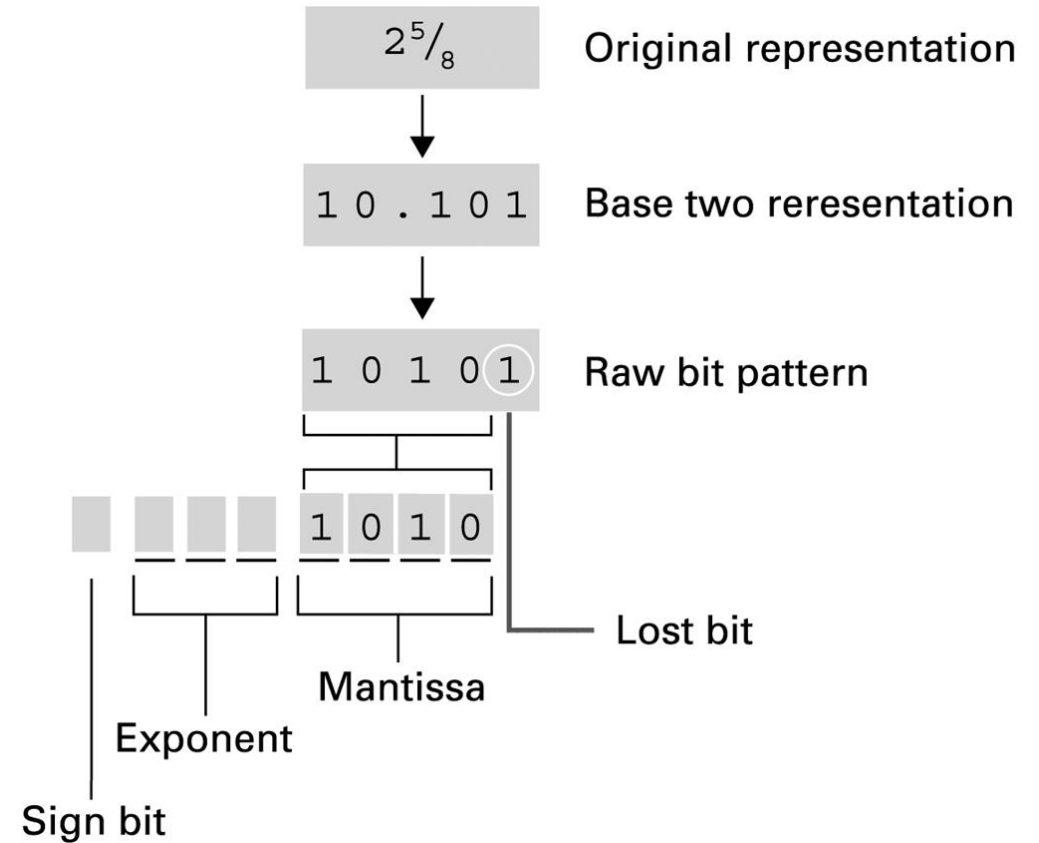
1.001 → _ _ _ _ 1 0 0 1 mantis

Üs? 1 olmalı: 101 (Excess-3)

İşaret? Pozitif: 0

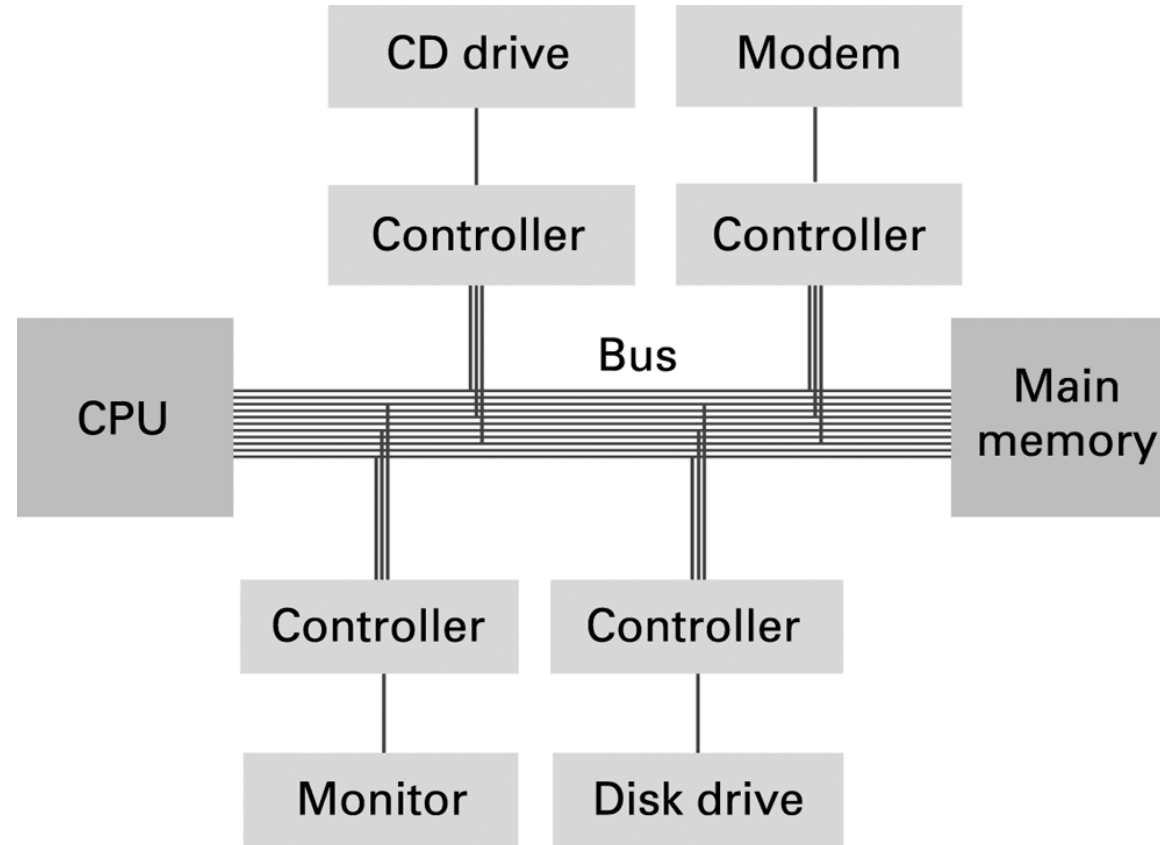
Sonuç:

0 101 1001

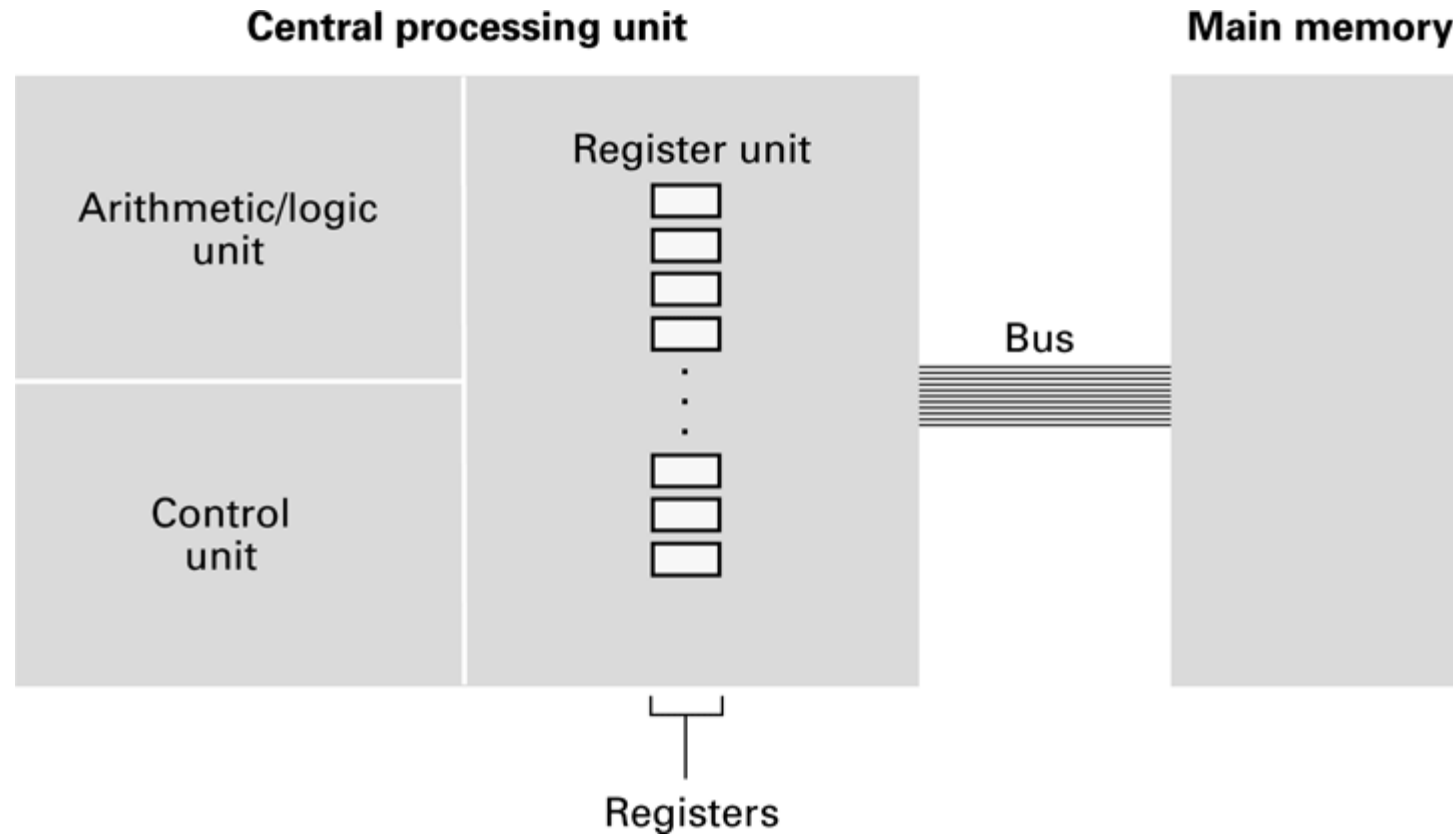


Bilgisayar Mimarisi

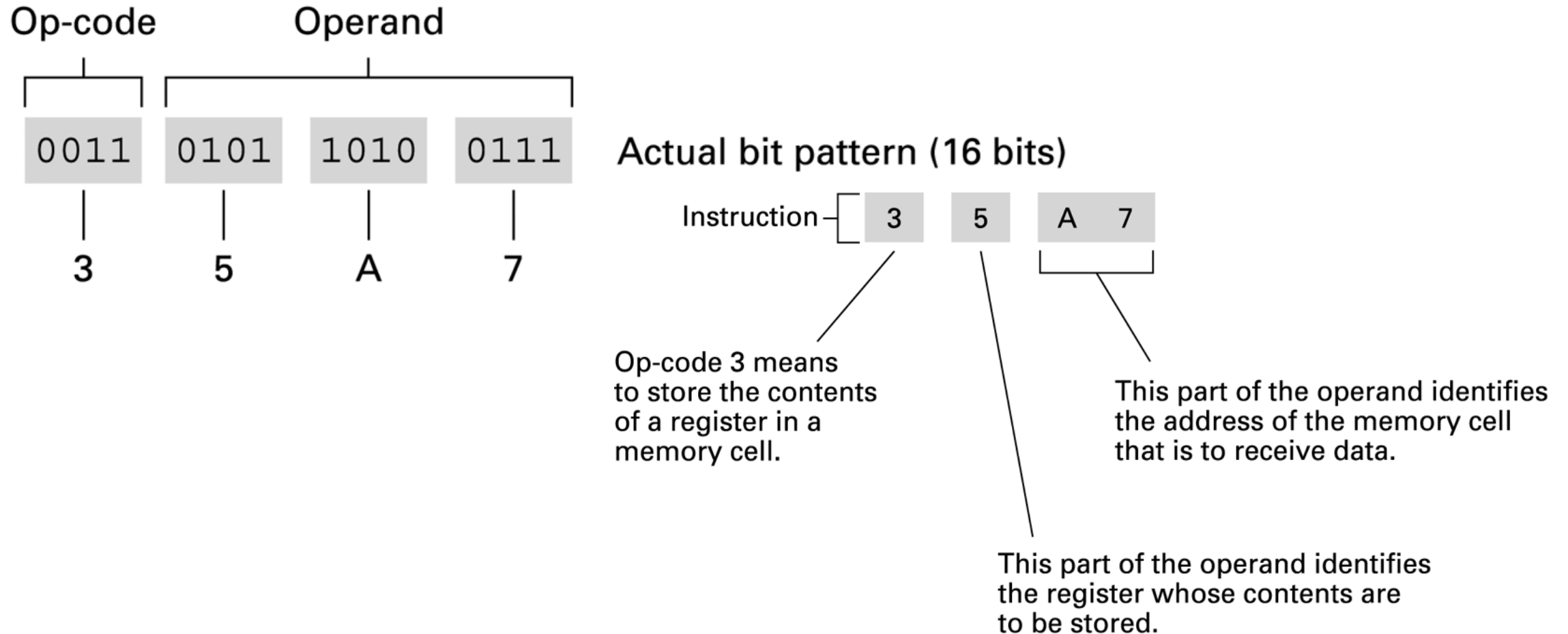
Bilgisayar Mimarisi



Bilgisayar Mimarisi



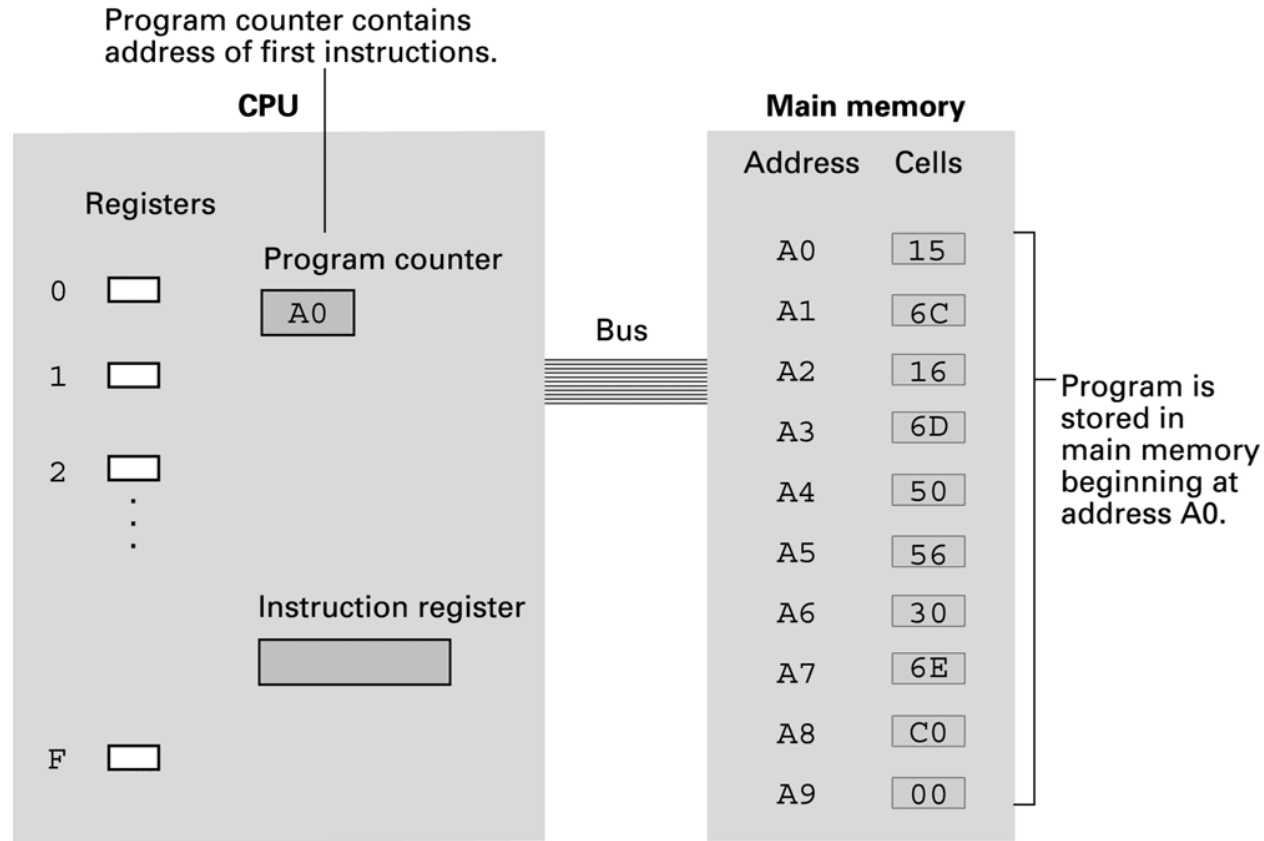
Örnek Makine Kodu



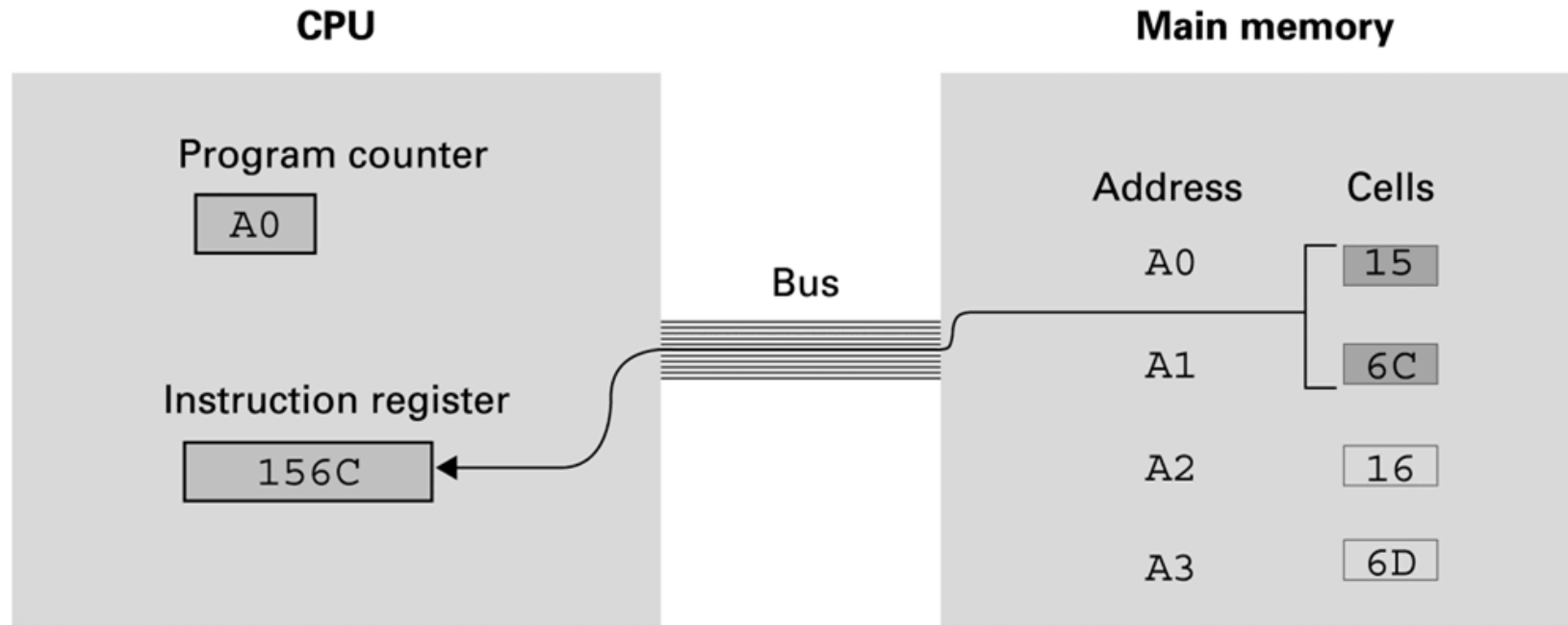
Örnek Makine Kodu

Encoded instructions	Translation
156C	Load register 5 with the bit pattern found in the memory cell at address 6C.
166D	Load register 6 with the bit pattern found in the memory cell at address 6D.
5056	Add the contents of register 5 and 6 as though they were two's complement representation and leave the result in register 0.
306E	Store the contents of register 0 in the memory cell at address 6E.
C000	Halt.

Örnek İşletim

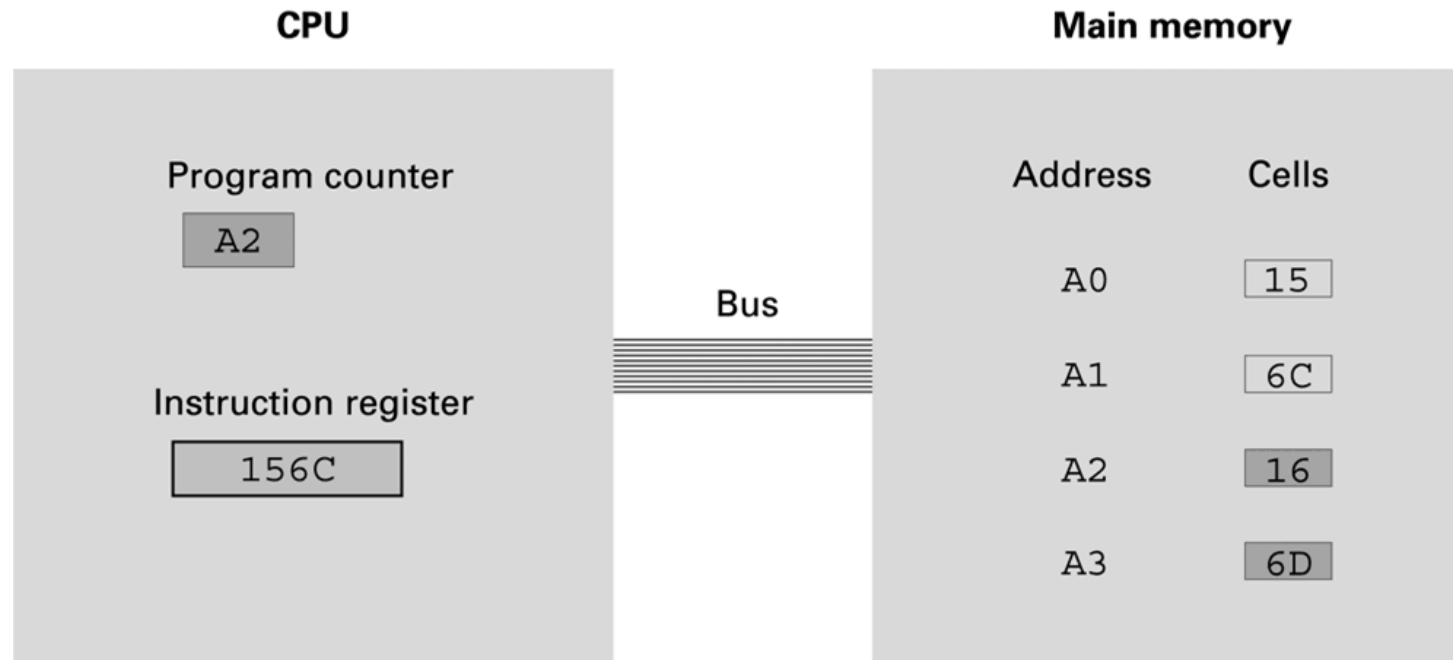


Örnek İşletim



- a. At the beginning of the fetch step the instruction starting at address A0 is retrieved from memory and placed in the instruction register.

Örnek İşletim



b. Then the program counter is incremented so that it points to the next instruction.