

Η κωδικοποίηση όλων των χαρακτήρων(σύμβολα, γράμματα, αριθμοί) του υπολογιστή βασίζεται στην κωδικοποίηση ASCII.

Decimal - Binary - Octal - Hex – ASCII Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	^
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	\$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(	72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29	)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[	123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D	]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

Συνήθως, μια θέση(ΚΟΥΤΑΚΙ) μνήμης αποθηκεύει ένα **byte**, το οποίο αποτελείται από **8 bits**.

Για να είναι δυνατή η αναπαράσταση γραμμάτων και άλλων συμβόλων στον υπολογιστή έπρεπε να βρεθεί ένα σχήμα **κωδικοποίησης** των συμβόλων αυτών στη γλώσσα του υπολογιστή, δηλαδή στο δυαδικό σύστημα αρίθμησης.

Ένα παρόμοιο σύστημα κωδικοποίησης αποτελεί ο κώδικας Μορς, με την βοήθεια του οποίου αντιστοιχίζονται γράμματα και αριθμοί σε ακολουθίες των συμβόλων **.** (τελεία) και **—** (παύλα). Το πιο γνωστό μήνυμα στον κώδικα Μορς είναι το μήνυμα κινδύνου **SOS**.



Ανάλυση:

- **1 bit** = Μια μονάδα δεδομένων με δυαδική τιμή, που μπορεί να είναι 0 ανοιχτό κυκλωμα ή 1 κλειστό κύκλωμα.
- **1 byte** = 8 bits, και είναι η βασική μονάδα αποθήκευσης δεδομένων στη μνήμη του υπολογιστή.

Άρα, μια τυπική θέση μνήμης (π.χ., ένας καταχωρητής ή μια θέση μνήμης στη RAM) αποθηκεύει 8 bits (ή 1 byte).

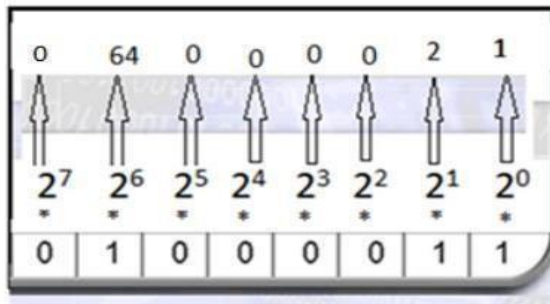
Αν έχουμε μεγαλύτερες μονάδες:

- Σε πολλές αρχιτεκτονικές υπολογιστών, μια θέση μνήμης μπορεί να είναι 16 bits παλιότεροι(2 bytes), **32 bits (4 bytes)** ή **64 bits (8 bytes)** **σημερα**, ανάλογα με την αρχιτεκτονική του συστήματος.

Άρα, η "θέση μνήμης" εξαρτάται **από την αρχιτεκτονική του υπολογιστή**, αλλά γενικά, μια θέση μνήμης αναφέρεται συνήθως σε 1 byte, δηλαδή 8 bits.

Πώς όμως μπορούμε να κάνουμε την μετατροπή από δυαδικό αριθμό σε δεκαδικό σύστημα(θέση), που αντιστοιχεί σε κάποιο χαρακτήρα του πίνακα ascii?

Μετατροπή Δυαδικού 01000011 (C) σε δεκαδικό 67



με πράξεις θα γίνει:

$$1*2^0 + 1*2^1 + 0*2^2 + 0*2^3 + 0*2^4 + 0*2^5 + 1*2^6 + 0*2^7 =$$
$$1*1 + 1*2 + 0*4 + 0*8 + 0*16 + 0*32 + 1*64 + 0*128 =$$
$$1 + 2 + 0 + 0 + 0 + 0 + 64 + 0 =$$
$$67_{(10)}$$

Άρα $1 + 2 + 64 = 67$ δεκαδικό του πίνακα Ascii που αντιστοιχεί στο χαρακτήρα C

Μετατροπή του δυαδικού 10000001₍₂₎ στον δεκαδικό(θέση) 129 του πίνακα ascii και αντιστοιχεί στον χαρακτήρα ù

Μετατροπή του δυαδικού 10000001 σε
δεκαδικό 129

128	0	0	0	0	0	0	1
↑	↑	↑	↑	↑	↑	↑	↑
2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
*	*	*	*	*	*	*	*
1	0	0	0	0	0	0	1

→ $1 + 128 = 129$

με πράξεις θα γίνει:

$$\begin{aligned} &1*2^0 + 0*2^1 + 0*2^2 + 0*2^3 + 0*2^4 + 0*2^5 + 0*2^6 + 1*2^7 = \\ &1*1 + 0*2 + 0*4 + 0*8 + 0*16 + 0*32 + 0*64 + 1*128 = \\ &1 + 0 + 0 + 0 + 0 + 0 + 0 + 128 = \\ &129_{(10)} \end{aligned}$$

Άσκηση

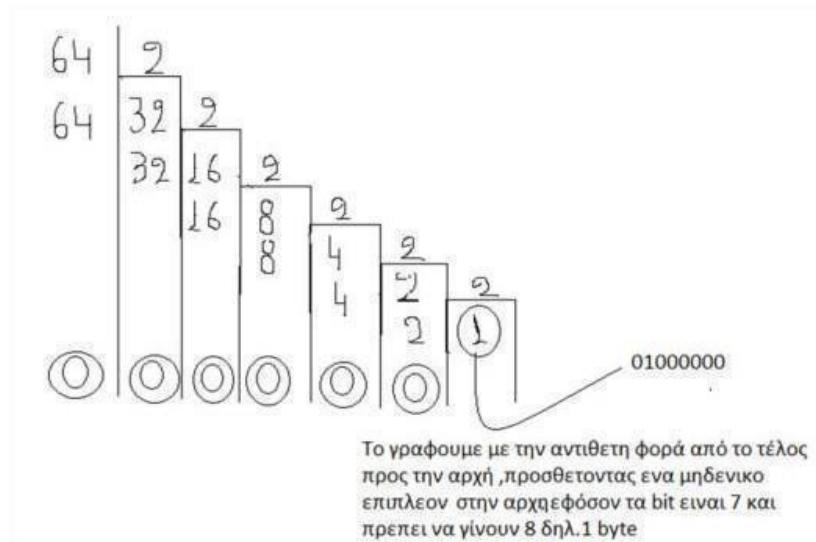
Να μετατρέψετε τον αριθμό 1111111_2 από το δυαδικό στο δεκαδικό σύστημα .

Πώς όμως μπορούμε να κάνουμε την μετατροπή από δεκαδικό σύστημα(θέση) σε δυαδικό αριθμό, που αντιστοιχεί σε κάποιο χαρακτήρα του πίνακα ascii?

Ο δεκαδικός 67 αντιστοιχεί στον χαρακτήρα C του πίνακα Ascii



Μετατροπή Δυαδικού δεκαδικό 64 σε δυαδικό 01000000



Ο δεκαδικός 64 αντιστοιχεί στον χαρακτήρα @ του πίνακα Ascii

Άσκηση Να μετατρέψετε τον αριθμό 255₍₁₀₎ από το δεκαδικό σύστημα στο δυαδικό σύστημα.

Δραστηριότητα. Αν πληκτρολογήσουμε ChatGpt από το πληκτρολόγιο, πως θα αποθηκευόταν στην μνήμη ram?



Παράδειγμα 2

Ας γράψουμε το μήνυμα ChatGPT στον κώδικα ASCII (δεκαδικό, δυαδικό) σύστημα.

C	h	a	t	G	P	T
67	104	97	116	71	80	84
1000011	1101000	1100001	1000111	1000011	1010000	1010100

Γράψτε την λέξη book δυαδική μορφή από τον πίνακα ascii στο τετράδιο.

ΟΚΤΑΔΙΚΟ ΚΑΙ ΔΕΚΑΕΞΑΔΙΚΟ

Όταν αναφερόμαστε όμως σε θέση μνήμης με λέξη 64 δυαδικά ψηφία (64-bit word), εννοούμε μια μονάδα αποθήκευσης μνήμης που έχει μέγεθος 64 bits (ή 8 bytes).

Ας δούμε πώς λειτουργεί αυτό:

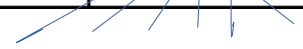
Λέξη 64 bits:

- Μια λέξη **64 bits** είναι 64 δυαδικά ψηφία (0 ή 1), που σημαίνει ότι μια θέση μνήμης αυτής της λέξης αποθηκεύει 64 bits.
- Αυτό ισοδυναμεί με **8 bytes** (καθώς 1 byte = 8 bits).
- Στις **64-bit αρχιτεκτονικές**, οι επεξεργαστές και η μνήμη διαχειρίζονται δεδομένα σε μονάδες των 64 bits (ή 8 bytes).
- Στην **64-bit αρχιτεκτονική**, ο επεξεργαστής και το σύστημα χρησιμοποιούν λέξεις 64-bit για να αποθηκεύσουν δεδομένα, εντολές ή διευθύνσεις μνήμης. Κάθε θέση μνήμης (όπως σε RAM ή cache) θα μπορούσε να είναι 64 bits, και κάθε καταχωρητής επεξεργαστή μπορεί να έχει μέγεθος 64 bits. Αυτό σημαίνει πολύ μεγαλύτερη επεξεργασία δεδομένων και πληροφοριών.

Έτσι δημιουργήθηκε το οκταδικό και δεκαεξαδικό σύστημα.

Σύστημα	Βάση	Ψηφία
Δυαδικό (Binary)	2	0,1
Οκταδικό (Octal)	8	0,1,2,3,4,5,6,7
Δεκαδικό (Decimal)	10	0,1,2,3,4,5,6,7,8,9
Δεκαεξαδικό (HexaDecimal)	16	0,1,2,3,4,5,6,7,8,9, A, B, C, D, E, F

10 11 12 13 14 15



Παρατηρήστε ότι στο δεκαεξαδικό σύστημα χρησιμοποιούνται και γράμματα ως ψηφία, επειδή τα ψηφία 0-9 δεν είναι αρκετά.

Έτσι το A είναι το 10, το B το 11, κ.ο.κ.

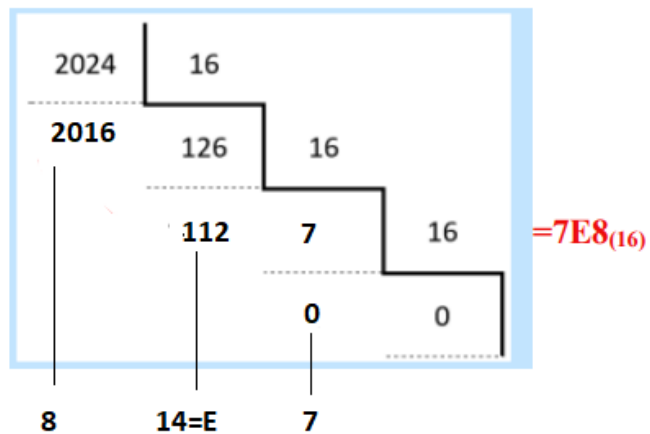
Παρακάτω παρουσιάζεται η μετατροπή του δυαδικού 101111_2 , $57_{(8)}$ και $2f_{(16)}$ στον δεκαδικό αριθμό $47_{(10)}$.

$$101111_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 32 + 8 + 4 + 2 + 1 = 47$$

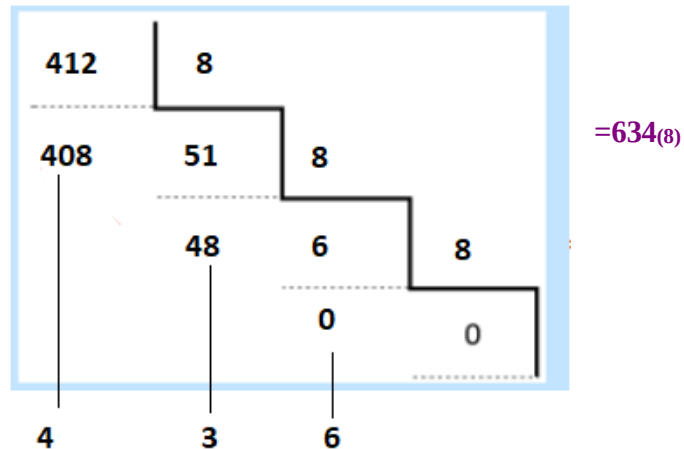
$$57_8 = 5 \cdot 8^1 + 7 \cdot 8^0 = 40 + 7 = 47$$

$$2F_{16} = 2 \cdot 16^1 + 15 \cdot 16^0 = 32 + 15 = 47$$

Μετατροπή του δεκαδικού $2024_{(10)}$ από το δεκαδικό σε δεκαεξαδικό



Αντίστοιχα κάνουμε και την μετατροπή από δεκαδικό σε οκταδικό . Δηλαδή Διάρθρωση με το 8.



Άσκηση Να μετατρέψετε τον 8F9₍₁₆₎ στο δεκαδικό και το αντίθετο δηλ. Απο δεκαδικό σε δεκαεξαδικό.
Να μετατρέψετε τον 59₍₈₎ στο δεκαδικό και το αντίθετο δηλ. Από δεκαδικό σε οκταδικό .

Το δεκαεξαδικό σύστημα και οκταδικό χρησιμοποιείται στην πληροφορική και την ηλεκτρονική για να επιτρέπει μια πιο ευανάγνωστη και αποδοτική αναπαράσταση των δυαδικών δεδομένων. Με αυτόν τον τρόπο, οι προγραμματιστές, οι μηχανικοί και οι τεχνικοί μπορούν να δουλέψουν πιο εύκολα με τα δεδομένα του υπολογιστή, να κάνουν τις μετατροπές έτσι να εντοπίζουν σφάλματα πιο γρήγορα .

Έτσι μπορεί να γίνει αναπαράσταση από το δεκαεξαδικό οκταδικό σε δυαδικό και απο δυαδικό σε δεκαεξαδικό και οκταδικό .

Για παράδειγμα:

Ο δυαδικός αριθμός 1111 1010 μπορεί να γραφτεί απλά ως DA σε δεκαεξαδικό.

15(D) 10(A)

Ο δυαδικός αριθμός 111 101 μπορεί να γραφτεί απλά ως 75 σε οκταδικό.

7 5