BINARNI KODOVI

BINARNO KODIRANI DECIMALNI BROJEVI (BCD KOD)

BCD kod za prikaz jedne znamenke koristi 4 bita, pri čemu se znamenka zapisuje kao binarni ekvivalent njene vrijednosti.

Primjer 2.1:

Dekadski broj 3720 prikažite u BCD kodu.

Rješenje:

Pretvorimo znamenke iz dekadskih u binarne vrijednosti u grupama po 4 bita:

3	7	2	0
0011	0111	0010	0000

Slijedi da je broj 3720 kodiran u BCD kod:

0011011100100000.

Zadatak 2.1:

Dekadski broj 952713 prikažite u BCD kodu.

Rješenje:

1001 0101 0010 0111 0001 0011

SIGURNOSNI KODOVI ZA KONTROLU PARITETA

U prijenosu signala se u praksi uvijek događaju pogreške, najčešće uzrokovane smetnjama na prijenosnim sustavima. Da bi se signali zaštitili, a time i informacija koju prenosimo, koristimo se kodovima koji omogućuju detekciju greške.

Redudanciju (zalihost) koda definiramo izrazom:

R = broj zaštitnih bitova / ukupan broj bitova

Primjer 2.2:

U računalu koristimo memoriju s neparnim paritetom. Podatak se smješta u 8 bita, a 9-ti bit je bit za otkrivanje i korekciju greške. Prilikom zapisa podatka u registar, bit pariteta se koristi tako da se postavlja na 1 ako je broj jedinica u bloku podatka neparan (neparan paritet). Kolika je redudancija ovakvog kodiranja ?

Ako broj zaštitnih bitova označimo sa r, a ukupan broj bitova sa n :

R= r/n=1/9=0,1111

Zadatak 2.2:

Objasni svojstva ovakvog kodiranja.

Rješenje:

Ovakav kod otkriva svaku jednostruku i neparan broj grešaka. Greške se ne mogu popraviti.

Zadatak 2.3:

U računalu koristimo memoriju s parnim paritetom. Kontrolni bit se postavlja na 1 ako je broj jedinica u registru podatka paran.
U registru je smješten binarni podatak: 1101 0101.
Na koju vrijednost se postavlja bit pariteta ?

Rješenje:

U registru podatka imamo 5 jedinica (neparan broj). Bit pariteta se postavlja na 0.

HAMMINGOVI KODOVI

Hamming je prvi izveo linearne kodove (n,k) koji korigiranju pojedinačnu pogrešku, pri čemu je:

n = ukupan broj bitova u kodnoj riječi
k = broj bitova informacije
m = n-k = broj kontrolnih bitova

Vrijedi pravilo da mora postojati odnos:

$2^m \geq k + m + 1$

Kontrolni bitovi se umeću na pozicijama koje su potencije broja 2.

Primjer 2.3:

Koliko je bitova potrebno za prijenos 12 bita informacije Hammingovim kodom.

Rješenje:

Ako sa C označimo kontrolne, a sa P podatkovne bitove:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
2^0	2^1		2^2				2^3								2^4	
C	C	P	C	P	P	P	C	P	P	P	P	P	P	P	C	P

potrebno je poslati ukupno 17 bitova da bismo prenijeli 12 bitova informacije, upotrebom Hammingova koda.

Zadatak 2.4:

Formirajte Hammingovu matricu za slanje 4 bita informacije

C1	C2	P1	C3	P2	P3	P4
1	2	3	4	5	6	7
2^0	2^1	x	2^2	x	x	x
0	0	0	1	1	1	1
0	1	1	0	0	1	1
1	0	1	0	1	0	1

Zadatak 2.5:

Upotrebom Hammingove matrice iz prethodnog zadatka zaštite 4 bita informacije: 0110
Primjenite neparni paritet.

Rješenje:

Kontrolne bitove određujemo uz pomoć matrice, primjenjujući slijedeće **pravilo**:

1. **Pravilo za računanje kontrolnog bita je sadržano u onom retku matrice gdje je za kontrolni bit kojeg računamo postavljena vrijednost 1**.

Prvi kontrolni bit ima jedinicu u 3. retku, ispitujemo paritet na P1, P2, P4
Drugi kontrolni bit ima jedinicu u 2. retku, ispitujemo paritet na P1, P3, P4
Treći kontrolni bit ima jedinicu u 1. retku, ispitujemo paritet na P2, P3, P4

2. **Računamo paritet na zadanim vrijednostima, iz bit-ova informacije**

C1 = ?

P1=0, P2=1, P4=0 -> UKUPAN BROJ JEDINICA NEPARAN

C1 = 1, ZBOG TRAŽENOG NEPARNOG PARITETA

C2 = ?

P1=0, P3=1, P4 =0

C2 = 1

C3 = ?

P2=1, P3=1, P4=0

C3 = 0

Spojimo kontrolne bitove s bitovima informacije i dobivamo za slanje:

1 1 0 **0** 1 1 0

Zadatak 2.7:

Hammingovim kodom zaštite podatak: 1000 1111 01. Primjenite neparni paritet. Izračunajte redudanciju takvog kodiranja.

1	2	3	4	5	6	7	8	9	10	11	12	13	14
c	c	p	c	p	p	p	c	p	p	p	p	p	p
0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	1	1	1	1	0	0	0	0	1	1	1
0	1	1	0	0	1	1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0	1	0	1	0	1	0
x	x	1	x	0	0	0	x	1	1	1	1	0	1
c1	c2		c3				c4						

c1	0
c2	0
c3	0
c4	0

R=r/n=4/14=0,286

EXCESS-3 KOD

Kao i BCD kod za kodiranje jedne dekadske znamenke koristi 4 bita, pri čemu se znamenka zapisuje kao binarni ekvivalent njene vrijednosti uvećane za 3.

Primjer 2.4:

Dekadski broj 3720 prikažite u Excess-3 kodu.

Rješenje:

3	7	2	0
0110	1010	0101	0011

Slijedi da se 3720 zapisuje kao: 0110 1010 0101 0011

BIKVINARNI KOD

Za kodiranje dekadskih znamenki se primjenuje slijedeća tablica :

broj	**bikvinarni kod** 5 0 4 3 2 1 0
0	0 1 0 0 0 0 1
1	0 1 0 0 0 1 0
2	0 1 0 0 1 0 0
3	0 1 0 1 0 0 0
4	0 1 1 0 0 0 0
5	1 0 0 0 0 0 1
6	1 0 0 0 0 1 0
7	1 0 0 0 1 0 0
8	1 0 0 1 0 0 0
9	1 0 1 0 0 0 0

Koristeći ovaj kod postižemo uvijek isti broj jedinica za svako kodiranje.

Primjer 2.5:

Dekadski broj 255 prikažite u bikvinarnom kodu.

Rješenje:

2	5	5
0 1 0 0 1 0 0	1 0 0 0 0 0 1	1 0 0 0 0 0 1

ASCII KOD

Služi za kodiranje znakova:

- 26 velikih slova engleske abecede
- 26 malih slova engleske abecede
- 10 znamenaka
- operatori, interpunkcije, upravljački znakovi

Za pohranu 1 znaka dovoljan je 1 byte.

Najstariji standard je ASCII (ISO-7 standard): 7 bita za informaciju + 1 bit za *paritet* ⇨ 2^7 = 128 različitih znakova.

Satoji se od specijalnih znakova (znakovi za upravljanje ulazno-izlaznim jedinicama računala) -> (0-31 dekadski)

i od znakova koji se mogu tiskati -> (32-127 dekadski)

Najvažnije ASCII vrijednosti:

0 – znak NULL (‘\0’)
32 – praznina (‘ ’)
48 – **57** – znamenke ‘0’-‘9’
65 – **90** – velika slova ‘A’ do ‘Z’
97 –**122** – mala slova ‘a’ do ‘z’ (97-65=**32** – razlika između malog i velikog slova!)

Ispis ASCII tablice:

Dekadski	Oktalno	Hex	Binarno	Vrijednost	
000	000	000	00000000	NUL	(Null char.)
001	001	001	00000001	SOH	(Start of Header)
002	002	002	00000010	STX	(Start of Text)
003	003	003	00000011	ETX	(End of Text)
004	004	004	00000100	EOT	(End of Transmission)
005	005	005	00000101	ENQ	(Enquiry)
006	006	006	00000110	ACK	(Acknowledgment)
007	007	007	00000111	BEL	(Bell)
008	010	008	00001000	BS	(Backspace)
009	011	009	00001001	HT	(Horizontal Tab)
010	012	00A	00001010	LF	(Line Feed)
011	013	00B	00001011	VT	(Vertical Tab)
012	014	00C	00001100	FF	(Form Feed)
013	015	00D	00001101	CR	(Carriage Return)
014	016	00E	00001110	SO	(Shift Out)
015	017	00F	00001111	SI	(Shift In)
016	020	010	00010000	DLE	(Data Link Escape)

017	021	011	00010001	DC1 (XON)	(Device Control 1)
018	022	012	00010010	DC2	(Device Control 2)
019	023	013	00010011	DC3 (XOFF)	(Device Control 3)
020	024	014	00010100	DC4	(Device Control 4)
021	025	015	00010101	NAK	
022	026	016	00010110	SYN	(Synchronous Idle)
023	027	017	00010111	ETB	(End of Trans. Block)
024	030	018	00011000	CAN	(Cancel)
025	031	019	00011001	EM	(End of Medium)
026	032	01A	00011010	SUB	(Substitute)
027	033	01B	00011011	ESC	(Escape)
028	034	01C	00011100	FS	(File Separator)
029	035	01D	00011101	GS	(Group Separator)
030	036	01E	00011110	RS	(Request to Send)
031	037	01F	00011111	US	(Unit Separator)
032	040	020	00100000	SP	(Space)
033	041	021	00100001	!	(exclamation mark)
034	042	022	00100010	"	(double quote)
035	043	023	00100011	#	(number sign)
036	044	024	00100100	$	(dollar sign)
037	045	025	00100101	%	(percent)
038	046	026	00100110	&	(ampersand)
039	047	027	00100111	'	(single quote)
040	050	028	00101000	(	(left parenthesis)
041	051	029	00101001	)	(right parenthesis)
042	052	02A	00101010	*	(asterisk)
043	053	02B	00101011	+	(plus)
044	054	02C	00101100	,	(comma)
045	055	02D	00101101	-	(minus or dash)
046	056	02E	00101110	.	(dot)
047	057	02F	00101111	/	(forward slash)
048	060	030	00110000	0	
049	061	031	00110001	1	
050	062	032	00110010	2	
051	063	033	00110011	3	
052	064	034	00110100	4	
053	065	035	00110101	5	
054	066	036	00110110	6	
055	067	037	00110111	7	
056	070	038	00111000	8	
057	071	039	00111001	9	
058	072	03A	00111010	:	(colon)
059	073	03B	00111011	;	(semi-colon)
060	074	03C	00111100	<	(less than)
061	075	03D	00111101	=	(equal sign)
062	076	03E	00111110	>	(greater than)
063	077	03F	00111111	?	(question mark)
064	100	040	01000000	@	(AT symbol)
065	101	041	01000001	A	
066	102	042	01000010	B	
067	103	043	01000011	C	
068	104	044	01000100	D	
069	105	045	01000101	E	
070	106	046	01000110	F	
071	107	047	01000111	G	
072	110	048	01001000	H	
073	111	049	01001001	I	
074	112	04A	01001010	J	
075	113	04B	01001011	K	
076	114	04C	01001100	L	
077	115	04D	01001101	M	
078	116	04E	01001110	N	
079	117	04F	01001111	O	
080	120	050	01010000	P	
081	121	051	01010001	Q	
082	122	052	01010010	R	
083	123	053	01010011	S	
084	124	054	01010100	T	
085	125	055	01010101	U	
086	126	056	01010110	V	
087	127	057	01010111	W	
088	130	058	01011000	X	
089	131	059	01011001	Y	
090	132	05A	01011010	Z	
091	133	05B	01011011	[	(left/opening bracket)
092	134	05C	01011100	\	(back slash)
093	135	05D	01011101	]	(right/closing bracket)

094	136	05E	01011110	^	(caret/cirumflex)
095	137	05F	01011111	_	(underscore)
096	140	060	01100000	`	
097	141	061	01100001	a	
098	142	062	01100010	b	
099	143	063	01100011	c	
100	144	064	01100100	d	
101	145	065	01100101	e	
102	146	066	01100110	f	
103	147	067	01100111	g	
104	150	068	01101000	h	
105	151	069	01101001	i	
106	152	06A	01101010	j	
107	153	06B	01101011	k	
108	154	06C	01101100	l	
109	155	06D	01101101	m	
110	156	06E	01101110	n	
111	157	06F	01101111	o	
112	160	070	01110000	p	
113	161	071	01110001	q	
114	162	072	01110010	r	
115	163	073	01110011	s	
116	164	074	01110100	t	
117	165	075	01110101	u	
118	166	076	01110110	v	
119	167	077	01110111	w	
120	170	078	01111000	x	
121	171	079	01111001	y	
122	172	07A	01111010	z	
123	173	07B	01111011	{	(left/opening brace)
124	174	07C	01111100	|	(vertical bar)
125	175	07D	01111101	}	(right/closing brace)
126	176	07E	01111110	~	(tilde)
127	177	07F	01111111	DEL	(delete)

Primjer: 2.6.

U računalu je binarno zapisana informacija: „VSTSI".

Prikažite zapis u računalu.

Rješenje:

V -> ascii: 086 -> binarno u računalu: 01010110
S -> ascii: 083 -> 01010011
T -> ascii: 084 -> 01010100
S -> ascii: 083 -> 01010011
I -> ascii: 073 -> 01001001

Zadatak: 2.8.

Prikažite zapis u računalu iz prethodnog zadatka pod pretpostavkom da se 1. bit s lijeve strane koristi za kontrolu pariteta. Primjenite neparni paritet.

01010110 - > broj jedinica: 4 - > bit pariteta=0 ->zapis s paritetom: 0 1010110
01010011 - > broj jedinica: 4 - > bit pariteta=0 ->zapis s paritetom: 0 1010011
01010100 - > broj jedinica: 3 - > bit pariteta=1 ->zapis s paritetom: 1 1010100
01010011 - > broj jedinica: 4 - > bit pariteta=0 ->zapis s paritetom: 0 1010011
01001001 - > broj jedinica: 3 - > bit pariteta=1 ->zapis s paritetom: 1 1001001