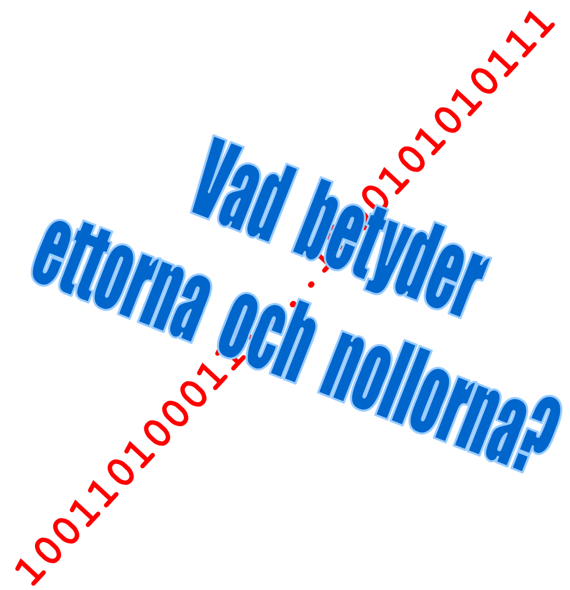
Binär kodning

Dagens föreläsning:
Läroboken kapitel 3

Ur innehållet:
Grundläggande binära koder
Talomvandlingar

Binära koder

- Begrepp
- Tal och talsystem
- Talomvandling
- ASCII-kod
- NBCD
- Gray-kod



Begrepp

begrepp	*betydelse*	*exempel...*
bit/bitar	minsta informationsenhet, kan anta två värden	0 eller 1
bitsträng binärt ord	sekvens av bitar	101100100001...
kodord	$\kappa_7\, \kappa_6\, \kappa_5\, \kappa_4\, \kappa_3\, \kappa_2\, \kappa_1\, \kappa_0$ också ett binärt ord men med en fastställd kodning (betydelse)	1000001 = "A" (ASCII) 1000001 = 65 (naturligt tal) 1000001 = -63(heltal)
ordlängd	antal bitar i ordet	
nibble	ordlängden 4 bitar	0101
byte	ordlängden 8 bitar	01011100

Tal och talsystem – positionssystem för basen 10

Ett ***N***-bitars tal. ***N*** = ***n***+***m*** där ***n*** är antalet siffror i heltalsdelen och ***m*** är antalet siffror i bråkdelen skriver vi allmänt:

$$d_{n-1}d_{n-2}\ldots d_1 d_0\,.\,d_{-1}d_{-2}\ldots d_{-(m-1)}d_{-m}$$

Mest signifikanta siffra (MSD) — Decimalpunkt — Minst signifikanta siffra (LSD)

Exempelvis, talet:

$$123{,}456 = 1\times10^{2} + 2\times10^{1} + 3\times10^{0} + 4\times10^{-1} + 5\times10^{-2} + 6\times10^{-3}$$
$$= 100 + 20 + 3 + 0{,}4 + 0{,}05 + 0{,}006$$

Där ***N***=6, ***n***=***m***=3, varje siffras vikt avgörs av dess position i talet...

Generellt positionssystem

Talbasen β kan dock vara praktiskt taget vad som helst...

$$d_{n-1}\times\beta^{n-1}+d_{n-2}\times\beta^{n-2}+...+d_1\times\beta^1+d_0\times\beta^0+d_{-1}\times\beta^{-1}+d_{-2}\times\beta^{-2}+...d_{-(m-1)}\times\beta^{-(m-1)}+d_{-m}\times\beta^{-m}$$

Exempel: $\beta=10$

$$d_{n-1}\times10^{n-1}+d_{n-2}\times10^{n-2}+...+d_1\times10^1+d_0\times10^0+d_{-1}\times10^{-1}+d_{-2}\times10^{-2}+...d_{-(m-1)}\times10^{-(m-1)}+d_{-m}\times10^{-m}$$

Exempel: $\beta=2$

$$d_{n-1}\times2^{n-1}+d_{n-2}\times2^{n-2}+...+d_1\times2^1+d_0\times2^0+d_{-1}\times2^{-1}+d_{-2}\times2^{-2}+...d_{-(m-1)}\times2^{-(m-1)}+d_{-m}\times2^{-m}$$

Vi använder vanligen det enklare skrivsättet

$$N=(d_{n-1}d_{n-2}...d_1d_0\;.\;d_{-1}d_{-2}...d_{-(m-1)}d_{-m})_\beta$$

Talbaser

Vi använder huvudsakligen tre olika talbaser:

Decimalt, för att vi är vana vid det.

Binärt, för att det motsvarar informationselementen i det digitala systemet.

Hexadecimalt, därför att det är ett bekvämt sätt att skriva grupper av binära siffror

Exempel:
$(13)_{10} = (1101)_2=(D)_{16}$

bas 10 decimalt	bas 2 binärt	bas 16 hexadecimalt
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Talomvandling

För talomvandling ***till*** basen 10 använder vi definitionen direkt...

(Uppgifter ur exempelsamlingen)

Omvandla till decimal form:
2.9d: $(101.011)_2$
2.10f: $(B3.D4)_{16}$

Vi löser på tavlan….

Omvandling från N_{10} till N_β

1. Dela upp N_{10} i heltalsdel och bråktalsdel.
2. Heltalsdelen omvandlas via succesiva divisioner med β.
3. Bråkdelen omvandlas via succesiva multiplikationer med β.

Exempelsamling uppgift 2.11d:
Omvandla $(122,18)_{10}$ till binär form. Bråkdelen avkortas vid behov till 6 korrekta bråksiffror.

1. Omvandla $(122)_{10}$ till binär form

$$
\begin{array}{rcll}
122/2 & = & 61 + 0/2 & \rightarrow d_0 = 0 \\
61/2 & = & 30 + 1/2 & \rightarrow d_1 = 1 \\
30/2 & = & 15 + 0/2 & \rightarrow d_2 = 0 \\
15/2 & = & 7 + 1/2 & \rightarrow d_3 = 1 \\
7/2 & = & 3 + 1/2 & \rightarrow d_4 = 1 \\
3/2 & = & 1 + 1/2 & \rightarrow d_5 = 1 \\
1/2 & = & 0 + 1/2 & \rightarrow d_6 = 1
\end{array}
$$

Terminerings-
villkor

Heltalsdelen således:

$(1111010)_2$

2. Omvandla $(0,18)_{10}$ till binär form

$$
\left.
\begin{array}{lll}
0{,}18 \times 2 = & \mathbf{0}{,}36 & \rightarrow d_{-1} = \mathbf{0} \\
0{,}36 \times 2 = & \mathbf{0}{,}72 & \rightarrow d_{-2} = \mathbf{0} \\
0{,}72 \times 2 = & \mathbf{1}{,}44 & \rightarrow d_{-3} = \mathbf{1} \\
0{,}44 \times 2 = & \mathbf{0}{,}88 & \rightarrow d_{-4} = \mathbf{0} \\
0{,}88 \times 2 = & \mathbf{1}{,}76 & \rightarrow d_{-5} = \mathbf{1} \\
0{,}76 \times 2 = & \mathbf{1}{,}52 & \rightarrow d_{-6} = \mathbf{1}
\end{array}
\right\}
$$

Termineringsvillkor
enligt uppgiftstexten
6 st. bråksiffror

Bråkdelen således:

$(0.001011)_2$

Omvandla till hexadecimal form

Exempel:
Omvandla $(122{,}18)_{10}$ till hexadecimal form. Bråkdelen avkortas vid behov till 2 korrekta bråksiffror.

Heltalsdelen:

$$
\begin{array}{rcll}
122/16 & = & 7 + 10/16 & \rightarrow d_0 = (10)_{10} = (A)_{16} \\
7/16 & = & 0 + 7/16 & \rightarrow d_1 = (7)_{10} = (7)_{16}
\end{array}
$$

Bråkdelen:

$$
\begin{array}{rcll}
0{,}18 \times 16 & = & 2{,}88 & \rightarrow d_{-1} = (2)_{10} = (2)_{16} \\
0{,}88 \times 16 & = & 14{,}08 & \rightarrow d_{-2} = (14)_{10} = (E)_{16}
\end{array}
$$

Svar:

$$(122{,}18)_{10} \approx 7A.2E$$

Trunkering och noggrannhet

Vi kan försäkra oss om att vi inte förlorar noggrannhet vid en talomvandling utan att samtidigt bestämma onödigt många (onödiga) bråksiffror på följande sätt.

- Vi vill omvandla **från radix *b* till radix *a***.
- Antag antal **noggranna siffror i radix *b* är *n***.
- Hur många siffror x i radix a krävs för att behålla noggrannheten?
- Vi har alltså sambandet: $a^x = b^n$ där a,b och n är kända, således:

$$x \log a = n \log b \Leftrightarrow x = n \log b / \log a$$

- Eftersom högerledet ger ett reellt tal och antalet siffror måste uttryckas som ett heltal gäller det då: $\boldsymbol{x = \lceil n \log b / \log a \rceil}$

Trunkering och noggrannhet

EXEMPEL:
Talet $(0.1234)_{16}$ är givet på hexadecimal form. Vi vill omvandla detta till decimal form, hur många decimala siffror krävs då i decimalutvecklingen för att noggrannheten ska behållas?

Lösning:
Radix b=16, radix a=10, antal noggranna siffror n=4, dvs

$$x = \lceil n \log b / \log a \rceil = \lceil 4 \log 16/\log 10 \rceil = \lceil 4 \log 16 \rceil = \lceil 4{,}8 \rceil = 5$$

DVS: 5 decimala siffror

Alfanumeriska tecken → ASCII

American Standard Code for Information Interchange

Typiskt användningsområde: Tangentbord

7-bitars ASCII kodning

0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1	$K_6K_5K_4$ / $K_3K_2K_1K_0$
NUL	DLE	SP	0	@	P	`	p	0 0 0 0
SOH	DC1	!	1	A	Q	a	q	0 0 0 1
STX	DC2	“	2	B	R	b	r	0 0 1 0
ETX	DC3	#	3	C	S	c	s	0 0 1 1
EOT	DC4	$	4	D	T	d	t	0 1 0 0
ENQ	NAK	%	5	E	U	e	u	0 1 0 1
ACK	SYN	&	6	F	V	f	v	0 1 1 0
BEL	ETB	‘	7	G	W	g	w	0 1 1 1
BS	CAN	(	8	H	X	h	x	1 0 0 0
HT	EM	)	9	I	Y	i	y	1 0 0 1
LF	SUB	*	:	J	Z	j	z	1 0 1 0
VT	ESC	+	;	K	[Ä	k	{ä	1 0 1 1
FF	FS	,	<	L	\Ö	l	|ö	1 1 0 0
CR	GS	-	=	M	]Å	m	}å	1 1 0 1
S0	RS	.	>	N	^	n	~	1 1 1 0
S1	US	/	?	O	_	o	RUBOUT (DEL)	1 1 1 1

ASCII– Exempel

Textsträngen ”Hej” representeras som:

```
1001000   1100101   1101010
    'H'       'e'       'j'
```

Textsträngen ”9756” representeras som:

```
0111001   0110111   0110101   0110110
    '9'       '7'       '5'       '6'
```

NBCD – Natural Binary Coded Decimal

- 4 bitars kodord
- Kodar decimala siffrorna 0-9

decimal siffra	NBCD kodord
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

NBCD – Exempel

Decimala talet 9756 representeras som:

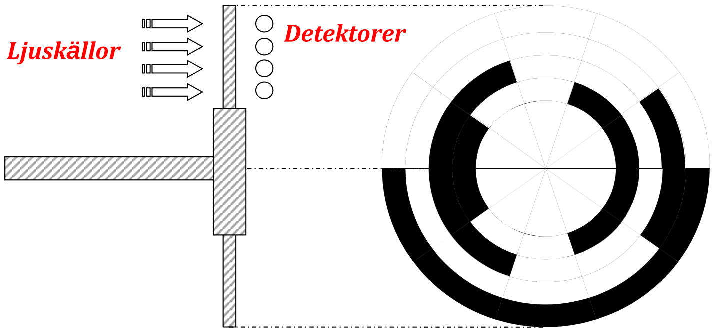
$$9756 = \underbrace{1001}_{9}\ \underbrace{0111}_{7}\ \underbrace{0101}_{5}\ \underbrace{0110}_{6}$$

Decimala talet 563,782 representeras som:

$$\underbrace{0101}_{5}\ \underbrace{0110}_{6}\ \underbrace{0011}_{3}\ ,\ \underbrace{0111}_{7}\ \underbrace{1000}_{8}\ \underbrace{0010}_{2}$$

Gray kod

Kodskiva – vanlig komponent i olika typer av vinkelgivare.



I "övergångarna": Koder ändrar sig endast i en bit. Förhindrar tillfälliga felavläsningar.

Gray kod

Decimal ordning	Kodord i trebitars Graykod	Kodord i fyrbitars Graykod
0	000	0000
1	001	0001
2	011	0011
3	010	0010
4	110	0110
5	111	0111
6	101	0101
7	100	0100
8		1100
9		1101
10		1111
11		1110
12		1010
13		1011
14		1001
15		1000

Gray-kod tillhör gruppen "reflekterande koder"