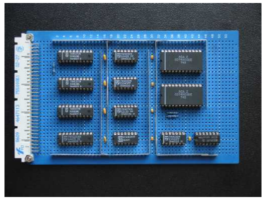
CHALMERS
UNIVERSITY OF TECHNOLOGY

Digital- och datorteknik



Föreläsning #2

Biträdande professor Jan Jonsson

Institutionen för data- och informationsteknik
Chalmers tekniska högskola

Talomvandling

Principer för omvandling mellan olika talsystem:

Från binärt till oktalt/hexadecimalt talsystem:

- Gruppering av binära siffror till ny siffra (fyll på med nollor vid behov)

Från binärt/oktalt/hexadecimalt till decimalt talsystem:

- Direkt decimal utvärdering av de olika positionernas värden

Från decimalt till binärt/oktalt/hexadecimalt talsystem:

- Upprepad division av heltalsdel
- Upprepad multiplikation av bråktalsdel

Avrundning efter omvandling:

För binära resultat tillämpar vi avrundning nedåt, s k trunkering:

- Man stryker de bråksiffror som inte längre behövs

Talomvandling

Binärt till oktalt/hexadecimalt:

Exempel:

Binära talet $(101\ 011)_2$ omvandlas till motsvarande oktala tal $(53)_8$.

Exempel:

Binära talet $(0100\ 1101)_2$ omvandlas till motsvarande hexadecimala tal $(4D)_{16}$.

bas 2 binärt	bas 16 hexa-decimalt	bas 10 decimalt	bas 8 oktalt
10000	10	16	20
10001	11	17	21
10010	12	18	22
10011	13	19	23
10100	14	20	24
10101	15	21	25
10110	16	22	26
10111	17	23	27
11000	18	24	30
11001	19	25	31
11010	1A	26	32
11011	1B	27	33
11100	1C	28	34
11101	1D	29	35
11110	1E	30	36
11111	1F	31	37

Talomvandling

Binärt/oktalt/hexadecimalt till decimalt:

Ett talvärde N i ett positionssystem med bas r (även kallat *radix*) kan alltså uttryckas som:

(Ekv. 2.1)

$$N = d_{n-1} \times r^{n-1} + d_{n-2} \times r^{n-2} + \cdots + d_0 \times r^0 + d_{-1} \times r^{-1} + \cdots + d_{-(m-1)} \times r^{-(m-1)} + d_{-m} \times r^{-m}$$

Exempel:

Binära talet $(101.11)_2$ omvandlas till bas 10 genom direkt decimal utvärdering med r=2, n=3, m=2, vilket ger $(5,75)_{10}$.

Exempel:

Hexadecimala talet $(2A1.F)_{16}$ omvandlas till bas 10 genom direkt decimal utvärdering med r=16, n=3, m=1, vilket ger $(673,9375)_{10}$.

Talomvandling

Decimalt till binärt/oktalt/hexadecimalt:

Det decimala talet delas upp i en heltalsdel (integer part) och en bråktalsdel (fractional part) som sedan behandlas separat: $(N)_{10} = (N_I,N_F)_{10}$

Algoritm för omvandling av heltalsdelen N_I

Låt N_I vara ett heltal i en godtycklig bas där de aritmetiska operationerna är definierade. Vi vill omvandla N_I till basen ***r*** och utför därför divisionen:

Ekv. 2.3 $$\frac{N_I}{r} = A + \frac{d_0}{r}$$

där A betecknar kvoten och d_0 betecknar resten. Av operationen framgår att $d_0 < r$, dvs en siffra i talsystemet med basen r. Vi fortsätter med succesiva divisioner tills kvoten är 0.

Algoritm för omvandling av bråktalsdelen N_F

Låt N_F vara en bråkdel i en godtycklig bas där de aritmetiska operationerna är definierade. Vi vill omvandla N_F till basen ***r*** och utför därför multiplikationen:

$$r\,N_F = d_{-1} + A'$$

där A' är en ny bråkdel och d_{-1} är ett heltal sådant att $d_{-1} < r$.

Vi utför nu upprepade multiplikationer tills den nya bråkdelen *antingen* är 0 *eller* tills tillräckligt många bråktalssiffror med den nya basen bestämts.

Talomvandling

Decimalt till binärt/oktalt/hexadecimalt:

Exempel:

Decimala talet $(N)_{10} = (211,678)_{10}$ skall omvandlas till bas 2, och avkortas till 6 stycken bråktalssiffror. Separation av heltalsdel och bråktalsdel ger: $(N_I)_{10} = (211)_{10}$ och $(N_F)_{10} = (0,678)_{10}$.

Omvandling av heltalsdel:

Heltalsdelen $(211)_{10}$ omvandlas till bas 2 genom successiva divisioner med 2, vilket ger $(N_I)_2 = (11010011)_2$.

Omvandling av bråktalsdel:

Bråktalsdelen $(0,678)_{10}$ omvandlas till bas 2 genom successiva multiplikationer med 2, vilket ger $(N_F)_2 = (0.101011\ldots)_2$.

Resultat:

$(N)_{10} = (211,678)_{10}$ motsvarar alltså $(N)_2 = (11010011.101011\ldots)_2$.

Binärkodning

Begrepp vid binär kodning

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 = -127 (heltal)
ordlängd	antal bitar i ordet	
nibble	ordlängden 4 bitar	0101
byte	ordlängden 8 bitar	01011100

Binärkodning

Vad döljs i detta mönster av hexadecimala siffror?

Är det bild (jpg)?

Är det text (ASCII)?

Är det musik (mp3)?

Är det maskinkod?

```
B4 87 04 00 E8 AB 52 01 00 CC 48 69 C6 88 00 00
00 42 C6 44 00 18 01 48 83 7D 08 10 72 09 48 8B
4D F0 E8 61 47 00 00 4C 8B 05 DA 4C 05 00 85 FF
0F 84 AF 00 00 00 48 8B 0D D3 4C 05 00 49 2B C8
49 8B C4 48 F7 E9 48 C1 FA 06 48 8B C2 48 C1 E8
3F 48 03 D0 48 3B D6 77 0D 48 8D 0D 58 87 04 00
E8 4F 52 01 00 CC 48 69 C6 88 00 00 00 4A 83 7C
00 10 00 74 70 48 3B D6 77 0D 48 8D 0D 37 87 04
00 E8 2E 52 01 00 CC 48 69 DE 88 00 00 00 49 03
D8 FF 15 91 79 01 00 39 43 04 77 49 48 8B 0D 6D
4C 05 00 4C 8B 05 5E 4C 05 00 49 2B C8 49 8B C4
48 F7 E9 48 C1 FA 06 48 8B C2 48 C1 E8 3F 48 03
D0 48 3B D6 77 0D 48 8D 0D EB 86 04 00 E8 E2 51
01 00 CC 48 69 CE 88 00 00 00 4A 8B 4C 01 08 FF
15 4B 79 01 00 49 C7 47 10 01 00 00 00 45 89 77
28 85 FF 0F 94 C1 B3 01 88 5C 24 30 84 C9 0F 84
DD 00 00 00 48 8B 15 05 4C 05 00 48 2B 15 F6 4B
05 00 49 8B C4 48 F7 EA 48 C1 FA 06 48 8B C2 48
C1 E8 3F 48 03 D0 0F 84 B5 00 00 00 48 B9 54 06
60 D9 D2 9D A7 A4 E8 F5 32 00 00 48 8D 0D 2E 43
05 00 FF 15 A8 43 05 00 8B 05 3A 43 05 00 85 C0
74 49 4C 8D 2D A7 D8 FF FF 48 8D 1D 50 43 05 00
FF C8 89 05 20 43 05 00 8B D0 49 8B 8C C5 90 6A
05 00 48 03 D2 8B 04 D3 89 01 8B 44 D3 04 89 41
08 8B 44 D3 08 89 41 10 8B 05 FA 42 05 00 85 C0
```

Binärkodning

Betrakta dessa bitsträngar som representerar tal?

Är det positiva tal?

Är det negativa tal?

Är det heltal?

Är det bråktal?

10000000
00000101
00001100
10111101
01110111
10000000
00000101
00000110
10110011
01110111
10000000
00000101
11111100
10101101
01110111
10000000
00000101
00110000
10100101

Binärkodning

Kodning:

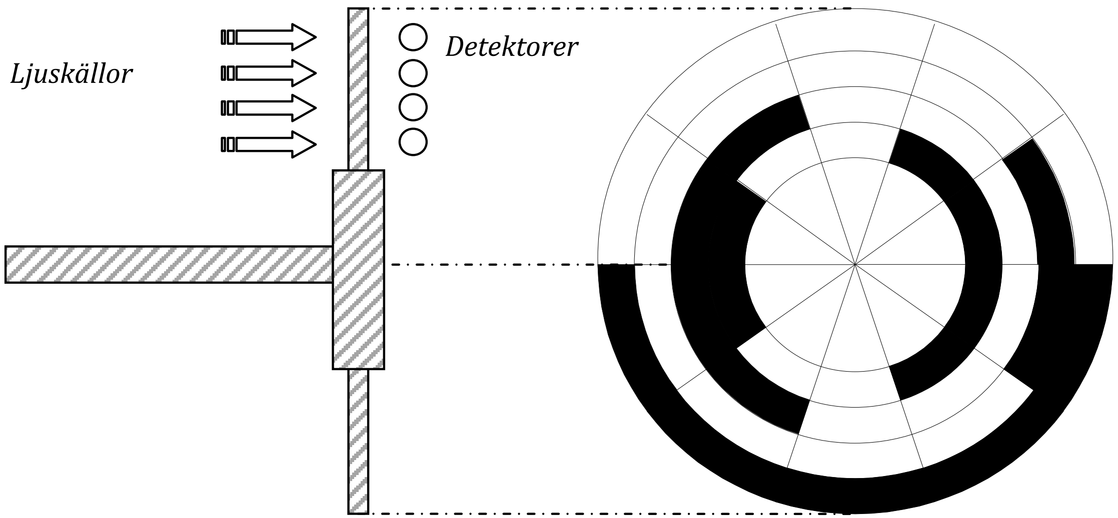
Tolkning av en sekvens av 1:or och 0:or som ett binärt tal är bara ett av flera sätt att koda en sådan sekvens. Exempel på andra sätt att koda binärt är:

- Gray-kod (reflekterande binärkod)
- Excess-kod (förskjuten binärkod)
- NBCD-kod (binärkodade decimaltal)
- ASCII, UTF-8 (alfanumeriska koder)
- Huffman-kod (komprimerande kod) – vanliga symboler får färre bitar
- Paritetskod (feldetekterande kod) – fåtal redundanta bitar läggs till
- Hammingkod (felrättande kod) – stor andel redundanta bitar läggs till

Binärkodning

Graykod:

För reflekterande koder gäller att två intilliggande kodord endast skiljer sig åt i en bitposition. Denna egenskap är viktig i vissa tillämpningar, t ex för att reducera fel i avläsning av kodskivor på roterande axlar.



FIGUR 2.1 KODSKIVA PÅ VRIDBAR AXEL

Binärkodning

Graykod:

Här framgår styrkan hos Graykoden: oavsett var kodskivan läses av kan felet i avläsningen inte överstiga 1 position.

Jämför detta med vanlig binärkod: om skivan skulle läsas av mellan 0111 och 1000 (och därmed kunna tolkas som 1111) skulle felet bli hela 8 positioner.

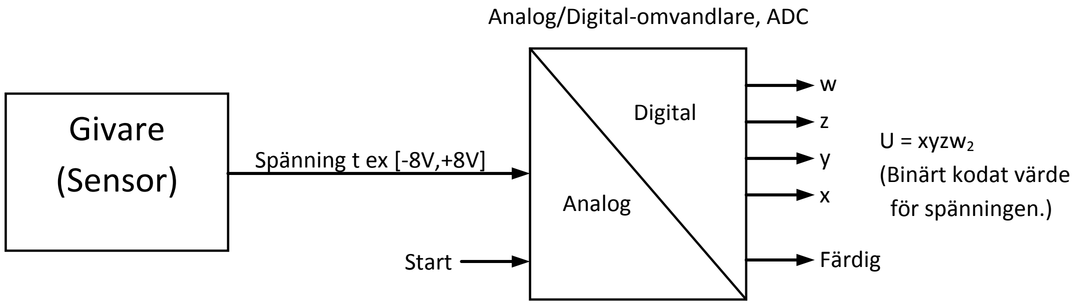
Decimal ordning	*binärkod*	*2-bitars Graykod*	*3-bitars Graykod*	*4-bitars Graykod*
0	0000	00	000	0000
1	0001	01	001	0001
2	0010	11	011	0011
3	0011	10	010	0010
4	0100		110	0110
5	0101		111	0111
6	0110		101	0101
7	0111		100	0100
8	1000			1100
9	1001			1101
10	1010			1111
11	1011			1110
12	1100			1010
13	1101			1011
14	1110			1001
15	1111			1000

Vi stöter på Graykod igen inom kort, i samband med minimering av grindnät.

Binärkodning

Excess-N kod:

I en excess-N-kod får man tolkningen genom att subtrahera N från binärkodens talvärde. På så sätt kan man justera tallinjen för att bäst passa tillämpningen, t ex för att representera negativa värden.



Koden decimalt:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Koden binärt:	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
Spänningsvärde:	-8V	-7V	-6V	-5V	-4V	-3V	-2V	-1V	0V	+1V	+2V	+3V	+4V	+5V	+6V	+7V	+8V

U

Man får här spänningen i volt genom att subtrahera 8 från kodordets värde (excess 8).

Binärkodning

NBCD-kod:

Istället för att omvandla ett decimalt tal till det binära talsystemet kan man binärt koda varje decimal siffra som ingår i talet.

I NBCD-kod (Natural Binary Coded Decimal) representeras varje decimal siffra naturligt nog med sitt ekvivalenta tal i det binära talsystemet.

Exempel:

$(23)_{10} = (0010\ 0011)_{NBCD}$

Decimal siffra Z	NBCD $\kappa_3\kappa_2\kappa_1\kappa_0$	Excess-3 kod $\kappa_3\kappa_2\kappa_1\kappa_0$	Excess-3 Graykod $\kappa_3\kappa_2\kappa_1\kappa_0$
0	0000	0011	0010
1	0001	0100	0110
2	0010	0101	0111
3	0011	0110	0101
4	0100	0111	0100
5	0101	1000	1100
6	0110	1001	1101
7	0111	1010	1111
8	1000	1011	1110
9	1001	1100	1010

Genom att använda excess-N- och Gray-koder kan man dessutom erhålla koder med symmetri.

Binärkodning

ASCII-kod:

När det som skall kodas inte enbart är siffror, utan även bokstäver och speciella symboler, krävs en expanderad kod. ASCII (7 bitar) och UTF-8 (8 bitar) är två vanliga sådana koder.

Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII	Hex	ASCII
0	NUL	20		10	DLE	30	0	40	@	50	P	60	`	70	p
1	SOH	21	!	11	DC1	31	1	41	A	51	Q	61	a	71	q
2	STX	22	"	12	DC2	32	2	42	B	52	R	62	b	72	r
3	ETX	23	#	13	DC3	33	3	43	C	53	S	63	c	73	s
4	EOT	24	$	14	DC4	34	4	44	D	54	T	64	d	74	t
5	ENQ	25	%	15	NAK	35	5	45	E	55	U	65	e	75	u
6	ACK	26	&	16	SYN	36	6	46	F	56	V	66	f	76	v
7	BEL	27	'	17	ETB	37	7	47	G	57	W	67	g	77	w
8	BS	28	(	18	CAN	38	8	48	H	58	X	68	h	78	x
9	HT	29	)	19	EM	39	9	49	I	59	Y	69	i	79	y
A	LF	2A	*	1A	SUB	3A	:	4A	J	5A	Z	6A	j	7A	z
B	VT	2B	+	1B	ESC	3B	;	4B	K	5B	[	6B	k	7B	{
C	FF	2C	,	1C	FS	3C	<	4C	L	5C	\	6C	l	7C	|
D	CR	2D	-	1D	GS	3D	=	4D	M	5D	]	6D	m	7D	}
E	SO	2E	.	1E	RS	3E	>	4E	N	5E	^	6E	n	7E	~
F	S1	2F	/	1F	US	3F	?	4F	O	5F	_	6F	o	7F	DEL

TABELL 2.3 ASCII-KOD