



NTNU

Det skapende universitet

TDT4105/TDT4110 Informasjonsteknologi grunnkurs:

Uke 38 – Digital representasjon, del 1

- Digital representasjon
- Tekst og tall
- positive, negative, komma?

Rune Sætre

satre@idi.ntnu.no

Læringsmål og pensum

- Læringsmål
 - Digital representasjon av informasjon
 - Binære og heksadesimale tallsystemer
 - Koding av tekst, heltall og flyttall
- Pensum
 - Teoriboka, tema 2: Digital Representasjon (resten dekkes neste uke)

Bruk piazza for å få rask hjelp til alles nytte!

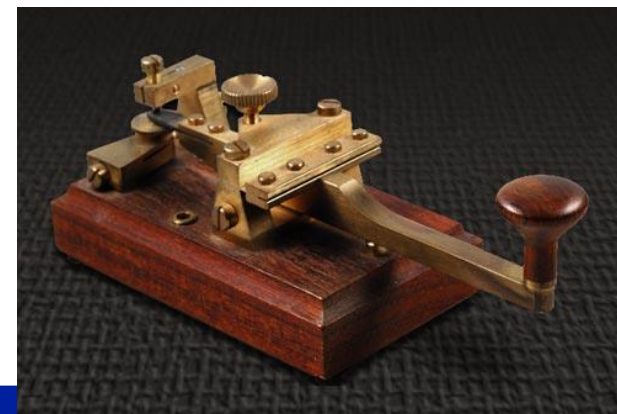
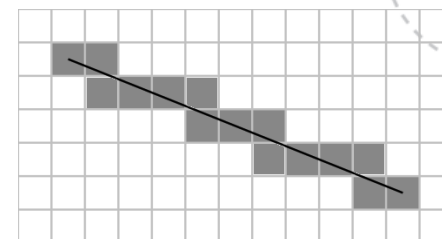
Ikke registrert deg? Gå inn på piazza.com og trykk på 'Sign up'



Allerede registrert deg? Gå inn på piazza.com og still spørsmål!

Representasjon av informasjon

- Bits: 0/1
- Bytes: 8 bits, $2^8 = 256$ symboler (mønster)
 - 0000 0000, 0000 0001, 0000 0010, ... , 1111 1110, 1111 1111
- Digitalisering: Representere informasjon ved symboler
 - Diskret versus analog (kontinuerlig), distinkt, skillbart
- P and A-koding (Present and Absent)
 - Presence / absence av fysisk fenomen
 - Strøm, spenning, magnetisme, lys/ikke-lys, ...
- Datamaskinens minne
 - Sekvens av bits gruppert i bytes
 - Hver byte, unik adresse



Flere bits, flere symboler

Desimaltall

N	10^n	Antall symboler
1	10^1	10
2	10^2	100
4	10^4	10000
8	10^8	100000000
10	10^{10}	10000000000
16	10^{16}	10000000000000000
32	10^{32}	10000000000000000000000000000000

Binærtall

N	2^n	Antall symboler
1	2^1	2
2	2^2	4
4	2^4	16
8	2^8	256
10	2^{10}	1024
16	2^{16}	65536
32	2^{32}	4294967296

Data + tolkning = informasjon

- Åtte byte (64 bits), for eksempel:
 - 1011 1101 0011 1001 1111 1011 1000 1001
1001 0001 1001 1001 1001 1001 1001 1000
- Hva kan dette representere?
 - 8 tegn (bokstaver, tall etc.)
 - Et flyttall
 - 4 (16 bits) heltall (med størrelse fra 0 til 65536)
 - 64 logiske verdier (0/1, usant / sant)
 - Et binært tall med 64 binære siffer
- Metadata er data om data, det vi trenger for å tolke riktig

Konvertering mellom binære tall og desimaltall

- Hver posisjon har en vekt: 2^{n-1} , 2^{n-2} , ..., 2^1 , 2^0
- Konvertering binært til desimalt er enkelt og rett fram
 - $101 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 4 + 0 + 1 = 5$
 - $11001010 = 2^7 + 2^6 + 2^3 + 2^1 = 128 + 64 + 8 + 2 = 202$
- Konvertering desimalt til binært er litt mer komplisert

Tall som konverteres	202	202	74	10	10	10	2	2	0
Posisjonsverdi	256	128	64	32	16	8	4	2	1
Subtraherer		128	64			8		2	
Binærtall	0	1	1	0	0	1	0	1	0

Addisjon og subtraksjon

- Samme prinsipper som for desimaltall

		1	1			
		1	1	0	0	(12)
+		1	1	1	0	(14)
	1	1	0	1	0	(26)

		2	2			
	1	1	0	1	0	(26)
-		1	1	1	0	(14)
	0	1	1	0	0	(12)

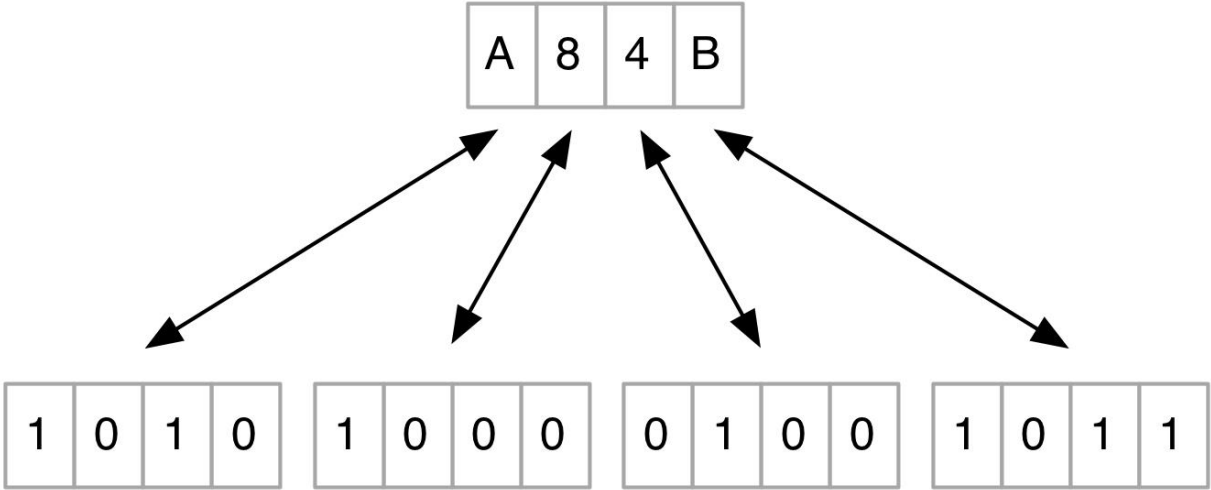
Heksadesimale tall

- Tallsystem med base 16: 0, 1, ..., 9, A, B, C, D, E, F
- Hvert heksadesimale siffer representerer 4 binære siffer

Hex	Titalls	Binært
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111

Hex	Titalls	Binært
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Hex til/fra binært



- $10 \cdot 16^3 + 8 \cdot 16^2 + 4 \cdot 16 + 11 = 43083$
- $1 \cdot 2^{15} + 1 \cdot 2^{13} + 1 \cdot 2^{11} + 1 \cdot 2^6 + 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 = 43083$
- Bruker ofte hex der vi skal gi inn binære tall
 - #FFFFFF = 1111 1111 1111 1111 1111 1111
 - Spesifiserer helt hvitt i RGB-fargekoding på WWW

Representasjon av tegn

- ASCII / ISO-8859-1 / Latin-1
 - En byte for hvert tegn
 - 256 forskjellige tegn
- Håndtere flere tegn?
 - Bruke flere byte per tegn
 - Unicode-standarden
 - Standard tegnsett som støtter alle språk som er i praktisk bruk
- Tegn kan brukes til å representere tall (som tekst)
 - 73593675
 - ASCII: 0011 0111 0011 0011 0011 0101 0011 1001
 0011 0011 0011 0110 0011 0111 0011 0101
 - 8 tegn x 1 byte = 8 byte
 - Som binærtall kodes det (lett) i 4-bytes integer (heltall)

Emne: [ITGK] Prosedyrer for fremvisning og godkjenning av Xving

Hei!

Undervisningsassistentene har rapportert at veldig mange studenters vanlig prosedyre for \diamond vingfremvisning og -godkjenning er gjenno- tildelt en studass vennligst svar på \diamond denne mailen (husk \diamond inkluderer fremvisning og godkjenning av \diamond ving for deg. Godkjenning av und

Ha en fin kveld!

Med vennlig hilsen,
P \diamond I-Christian S. Nj \diamond Istad
Vitass ITGK



NTNU

Det skapende universitet

ASCII-tabell (figur 8.3)

- A: 0100 0001
- B: 0100 0010
- ...
- 0: 0011 0000
- 1: 0011 0001
- ...
- Copyright-tegn:
1010 1001
- Æ, ø og å!

ASCII	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	
	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	
0000	N _U	S _H	S _X	E _X	E _T	E _O	A _K	B _L	B _S	H _T	L _F	Y _T	F _F	C _R	S _O	S _I
0001	D _L	D ₁	D ₂	D ₃	D ₄	N _K	S _Y	E _Z	C _N	E _M	S _B	E _C	F _S	G _S	R _S	U _S
0010		!	"	#	\$	%	&	'	()	*	+	,	-	.	/
0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	p	q	r	s	t	u	v	w	x	y	z	{		}	~	D _T
1000	S ₀	S ₁	S ₂	S ₃	I _N	N _L	S _S	E _S	H _S	H _J	Y _S	P _D	P _V	R _I	S ₂	S ₃
1001	D _C	P ₁	P ₂	S _E	C _C	M _M	S _P	E _P	Q _S	Q _Q	Q _A	C _S	S _T	O _S	P _M	A _P
1010	A _o	i	ç	£	¤	¥		\$..	©	ª	«	¬	-	®	¯
1011	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
1100	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
1101	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
1110	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
1111	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Figure 7.3 ASCII, the American Standard Code for Information Interchange.

Note: The original 7-bit ASCII is the top half of the table; the whole table is known as Extended ASCII (ISO-8859-1). The 8-bit symbol for a letter is the four row bits followed by the four column bits (e.g., A = 0100 0001, while z = 0111 1010). Characters shown as two small letters are control symbols used to encode nonprintable information (e.g., B_S = 0000 1000 is backspace). The bottom half of the table represents characters needed by Western European languages, such as Icelandic's eth (ð) and thorn (þ).

Representasjon av heltall

- Fast størrelse i antall byte
- int8 (1 byte)
 - $2^8 = 256$ symboler
 - Bare positive tall (unsigned): 0 - 255
 - Positive og negative (signed): Fra -128 til 127
- int32 (4 byte)
 - $2^{32} = 4\,294\,967\,296$ symboler
 - Bare positive tall: 0 – $(2^{32}-1)$
 - Positive og negative tall: Fra -2 147 483 648 til 2 147 483 647
- Eksakte verdier
- Kan få overflyt
 - Int8: 255 + 1 kan ikke representeres

Positive og negative tall

- Fortegnsbit (0 = positivt, 1 = negativt)
- Case: 3 bits heltall (8 symboler)
- Enkleste representasjon vist til høyre
- **To nuller! «feil vei»! (Dårlig løsning!)**
- Vanlig å bruke 2-komplement for negative tall
 - Fra + til –, og – til +, samme oppskrift:
 - Ta positiv koding, bytt alle 0 og 1, legg til 1 (ignorerer overflyt)
 - Eks:
 - 001 -> 110 -> 111 (-1)
 - 010 -> 101 -> 110 (-2)
 - 011 -> 100 -> 101 (-3)
 - Bare 1 null og "vinner" -4 (100)

Fortegnsbit
↓

0	1	1	3
0	1	0	2
0	0	1	1
0	0	0	0
1	0	0	-0
1	0	1	-1
1	1	0	-2
1	1	1	-3

-4
-3
-2
-1

2-er komplement Bonus: Subtraksjon som addisjon

$$\begin{array}{r} \\ \\ \end{array} \begin{array}{|c|c|c|} \hline 0 & 1 & 1 \\ \hline 1 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array} \begin{array}{l} 3 \\ -2 \\ 1 \end{array}$$

$$\begin{array}{r} \\ \\ \end{array} \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & 0 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \begin{array}{l} 2 \\ -3 \\ -1 \end{array}$$

(ignorerer overflyt ut av 3 bit)

- Forenkler datamaskinenes ALU

Representasjon av reelle tall

- Tallene representeres i scientific notation
 - $1234,56 = 1,23456 \times 10^3$
 - $0,0011 = 1,1 \times 10^{-3}$
 - Normalisert med ett siffer foran komma
- Kan representere tall med:
 - Mantisse (fractional part) inkludert fortegnsbitt
 - Eksponent (exponent)
- Komma alltid etter første siffer i mantissen (floating point)
- Eksempler:
 - 1234,56 har 0 (fortegn), 123456 (mantisse), 3 (eksponent)

IEEE standard for floating-point numbers

- Single precision

- 32 bits (4 byte)
- 24 bits til fortegn + mantisse
- 8 bits til eksponenten
- Absoluttverdier omtrent: $1,2 \times 10^{-38} - 3,4 \times 10^{38}$
- Omtrent 7 signifikante desimale siffer
- Eks: $10.0 = 1.25 \cdot 2^3 = 0\ 1000010\ 010000000000000000000000$

- Double precision

- 64 bits (8 byte)
- 53 bits til fortegn + mantissen
- 11 bits til eksponenten
- Absoluttverdier: $2,2 \times 10^{-308} - 1,8 \times 10^{308}$
- Nøyaktighet: Omtrent 16 desimale siffer

For- Eksponent
tegn

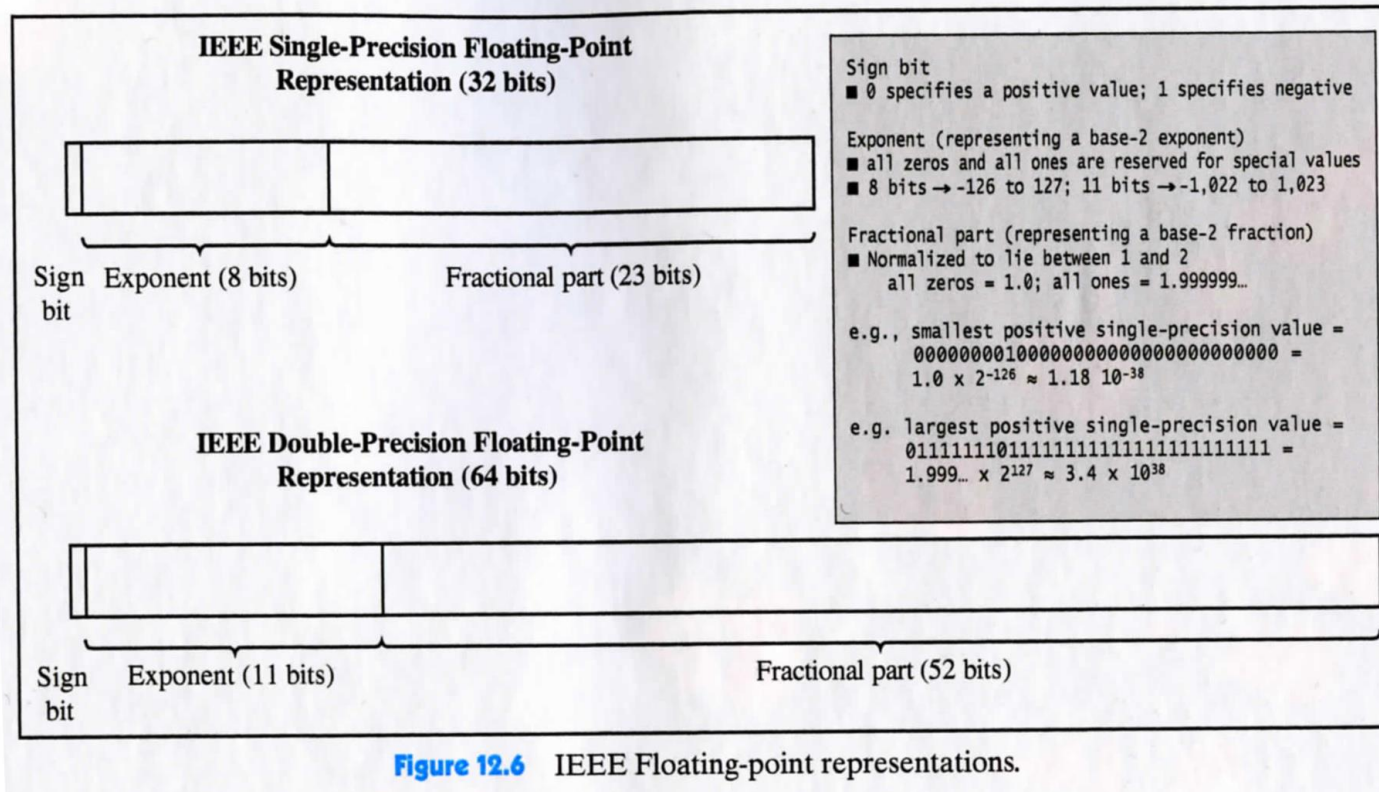
Mantisse



NTNU

Det skapende universitet

IEEE (fortsatt)



- Kapasitet og nøyaktighet versus "fotavtrykk" i minnet og ressurser forøvrig



NTNU

Det skapende universitet

IEEE (fortsatt)

- Oppgave 1
 - Hva blir 10 som en IEEEs Single Precision (32 bit) floating point number?
 - **Fortegnsbit** = 0 (positivt tall)
 - Finn den største toerpotenser som er mindre enn 10 i.e. 3 siden $2^3 = 8$ mens $2^4 = 16$
 - Legg til 127 ($=2^7-1$) til dette tallet $127+3 = 130$ og du har funnet eksponenten
 - **Eksponent** = 10000010 (binærtallet til 130)
 - Del så 10 på 8 for å finne mantissen. $10/8 = 1.25$. Siden 1 er implisitt behøver bare 0.25 å representeres. Multipliser dette med 2^{23} og representer som et binærtall. $0.25 \cdot 2^{23} = 2097152$.
 - **Mantisse** = 010000000000000000000000

Neste uke

- Representasjon og koding av multimedia-data
- Komprimering av data
- Mer om metadata (data om data)