



# NTNU

Det skapende universitet

## **TDT4105/TDT4110 Informasjonsteknologi grunnkurs:**

Uke 39 – Digital representasjon, del 1

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

**Alf Inge Wang**

[alfw@idi.ntnu.no](mailto:alfw@idi.ntnu.no)

# Auditorieøving neste uke (40)!

- Hva?
  - Øving på papir (som på eksamen) i maks 2 timer. Både teori og programmering. Uten hjelpemidler.
- Når?
  - I øvingsforelesningstimen din
- Hvor?
  - Sjekk Blackboard!

# Bruk piazza for å få rask hjelp til alles nytte!

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



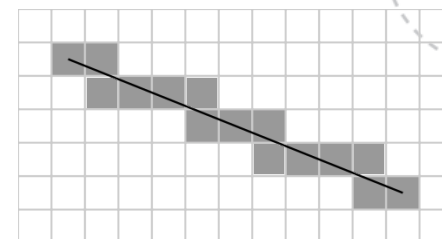
**Allerede registrert deg?** Gå inn på piazza.com og still spørsmål!

# 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)

# 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	$2^{10}$	1 024
16	$2^{16}$	65 536
32	$2^{32}$	4 294 967 296
64	$2^{64}$	18 446 744 073 709 551 616



**NTNU**

Det skapende universitet

# 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.
  - Eks: Hva er 202 binært?

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





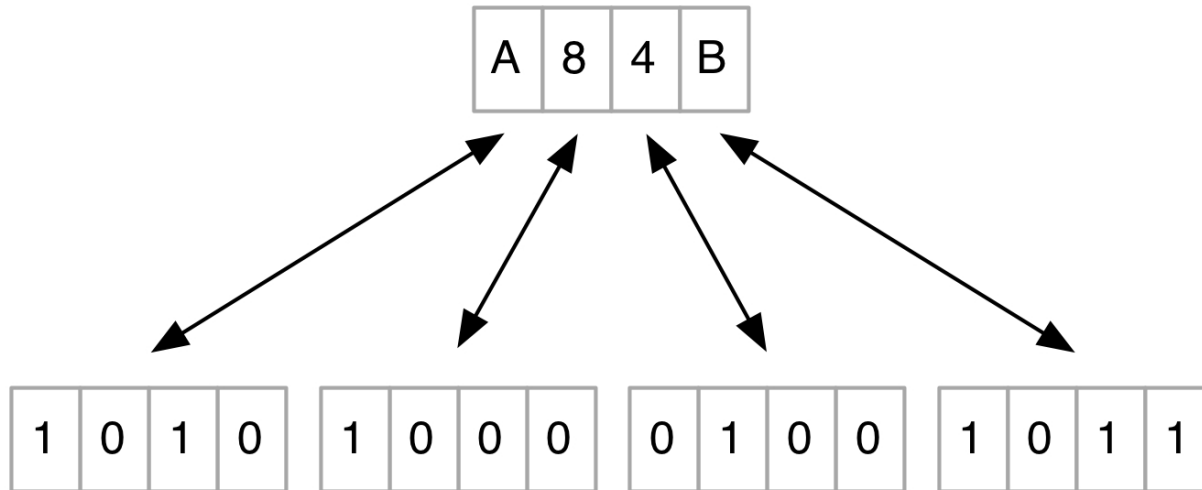
# Heksadesimale tall

- 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

# Oppgave: Fra binært til Hex



- Hva blir følgende binærtall som heksadesimalt tall?

11010000 00100000

# 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

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

Hei!

Undervisningsassistentene har rapportert at veldig mange studenters vanlig prosedyre for  $\diamond$ vingsfremvisning 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

# 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	0	1	1	1	1	1	1	1	1
	0	0	0	0	1	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	1
	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0000	N <sub>U</sub>	S <sub>H</sub>	S <sub>X</sub>	E <sub>X</sub>	E <sub>T</sub>	E <sub>O</sub>	A <sub>K</sub>	B <sub>L</sub>	B <sub>S</sub>	H <sub>T</sub>	L <sub>F</sub>	Y <sub>T</sub>	F <sub>F</sub>	C <sub>R</sub>	S <sub>O</sub>	S <sub>I</sub>
0001	D <sub>L</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	N <sub>K</sub>	S <sub>V</sub>	E <sub>Σ</sub>	C <sub>N</sub>	E <sub>M</sub>	S <sub>B</sub>	E <sub>C</sub>	F <sub>S</sub>	G <sub>S</sub>	R <sub>S</sub>	U <sub>S</sub>
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 <sub>T</sub>
1000	Š <sub>0</sub>	Š <sub>1</sub>	Š <sub>2</sub>	Š <sub>3</sub>	I <sub>N</sub>	N <sub>L</sub>	S <sub>S</sub>	E <sub>S</sub>	H <sub>S</sub>	H <sub>J</sub>	Y <sub>S</sub>	P <sub>D</sub>	P <sub>V</sub>	R <sub>I</sub>	S <sub>2</sub>	S <sub>3</sub>
1001	D <sub>C</sub>	P <sub>1</sub>	P <sub>2</sub>	S <sub>E</sub>	C <sub>C</sub>	M <sub>M</sub>	S <sub>P</sub>	E <sub>P</sub>	Q <sub>S</sub>	Q <sub>Q</sub>	Q <sub>A</sub>	C <sub>S</sub>	S <sub>T</sub>	O <sub>S</sub>	P <sub>M</sub>	A <sub>P</sub>
1010	A <sub>O</sub>	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.

# Oppgave: ASCII og binært



- Fra eksamensoppgave (kont 2017):
- Hvis 'OSTE' kodet i ASCII blir  
'0100 1111 0101 0011 0101 0100 0100 0101',  
hva blir 'POP' kodet i ASCII?
  - a. '0100 1111 0101 0011 0101 0100'
  - b. '0101 0011 0101 0100 0100 0101'
  - c. '0100 1111 0101 0000 0101 0000'
  - d. '0101 0000 0100 1111 0101 0000'

# 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 til  $(2^{32}-1)$
  - Positive og negative tall: Fra -2 147 483 648 til 2 147 483 647
- Kan få overflyt
  - Int8: 255 + 1 kan ikke representeres



# Positive og negative tall: Fortegnsbit

- Fortegnsbit: Første bit er fortegn:
  - 0 = positivt
  - 1 = negativt
- Case: 3 bits heltall (8 symboler)
- Enkleste representasjon vist til høyre
- Dårlig løsning:
  - To nuller!
  - Tungvint for ALU i prosessoren

Fortegnsbit verdi	Bits
0	<u>0</u> 00
-0	<u>1</u> 00
1	<u>0</u> 01
-1	<u>1</u> 01
2	<u>0</u> 10
-2	<u>1</u> 10
3	<u>0</u> 11
-3	<u>1</u> 11

↑  
Fortegnsbit



**NTNU**

Det skapende universitet

# Positive og negative tall: 2-komplement

- Fra + til -, og - til +, med samme oppskrift:
  - Ta positiv koding
  - Bytt alle 0 og 1,
  - Legg til 1 (ignorerer overflyt)

Eks:

0 000 -> 111 +1 -> 000 (0)  
 1 001 -> 110 +1 -> 111 (-1)  
 2 010 -> 101 +1 -> 110 (-2)  
 3 011 -> 100 +1 -> 101 (-3)  
**4 100 -> 011 +1 -> 100 (-4)**

2-komplement	Bits
3	011
2	010
1	001
0	000
-1	111
-2	110
-3	101
-4	100

# 2-er komplement Bonus: Subtraksjon som addisjon

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

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

(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 flyttall med:
  - Mantisse (fractional part) inkludert fortegnsbitt (som heltall)
  - Eksponent
- Komma alltid etter første siffer i mantissen (floating point)
- Eksempel:
  - $1234,56 = 1,23456 \times 10^3$  som representeres som:
  - Fortegn: 0
  - Mantisse: 123456
  - Eksponent: 3

# 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
- 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

# Kahoot! challenge (frivillig)

- Du kan bryne deg på oppgaver fra tidligere eksamenssett om Digital Representasjon ved å laste ned Kahoot! appen på App Store eller Google Play (gratis).
- For å spille kan du enten benytte Gamepin: 0135368 eller følgende lenke: <https://kahoot.it/challenge/0135368>
- Deadline: 1.oktober kl 23:00



# Neste uke

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