MAT1030 – Diskret matematikk

Plenumsregning 3: Ukeoppgaver fra kapittel 2 & 3

Roger Antonsen

Matematisk Institutt, Universitetet i Oslo

31. januar 2008

Oppgave 2.7 - Horners metode

(a) 7216_8: $7 \overset{\cdot 8+2}{\longrightarrow} 58 \overset{\cdot 8+1}{\longrightarrow} 465 \overset{\cdot 8+6}{\longrightarrow} 3726$. Svar: 3726_{10}

(b) 543517_8

(c) $8CB2_{16}$: $8 \overset{\cdot 16+12}{\longrightarrow} 140 \overset{\cdot 16+11}{\longrightarrow} 2251 \overset{\cdot 16+2}{\longrightarrow} 36018$. Svar: 36018_{10}

(d) $E490DF_{16}$

Oppgave 2.8

Skriv ned Horners metode som pseudokode. Anta at basen blir gitt som første argument og deretter at tallet som skal overføres til desimal form blir gitt som en liste med sifre.

Løsning

1. Input *base*
2. Input $x_1, \dots, x_n$
3. $resultat \leftarrow x_1$
4. **For** $i = 2$ **to** n **do**
   4.1. $resultat \leftarrow (resultat \cdot base) + x_i$
5. Output *resultat*

Oppgave 2.14

Utfør følgende regnestykker i binær aritmetikk:

(a) $1101101_2 + 1011110_2$

(b) $1001101_2 + 101011_2$

(c) $1110011_2 - 101101_2$

(d) $1100010_2 - 1010111_2$

(e) $10011_2 \cdot 1101_2$

(f) $11010_2 \cdot 10101_2$

(g) $110110_2 \div 1001_2$

(h) $10110_2 \div 11_2$ (3 siffer etter komma)

Løsning

Gjennomgått på tavlen.

Oppgave 3.1

Finn toerkomplementet til følgende 8-bits binære tall.

(a) 11010100_2

(b) 01101001_2

Løsning

(a) $11010100_2 \rightsquigarrow$
00101100_2

(b) $01101001_2 \rightsquigarrow$
10010111_2

Oppgave 3.6

Hvor mange bit brukes for å lagre $2,8 \cdot 10^{14}$ heltall (ikke-negative heltall lagret uten fortegnsbit)?

Løsning

Hvis vi har n bit, så kan vi lagre 2^n heltall. Vi må derfor finne n slik at $2^n = 2,8 \cdot 10^{14}$, det vil si $n = \log_2(2,8 \cdot 10^{14}) = 47.99242_{10} \approx 48$

Oppgave 3.9

Uttrykk $1101110100,1001_2$ i normalisert binær form, og finn dens 32-bit-datarepresentasjon. Anta at 8 bit brukes for å lagre eksponenten og at eksponentbiasen er $2^7 - 1 = 127$.

Løsning

- $1101110100,1001_2 = 0,11011101001001_2 \cdot 2^{10}$
- Fortegnsbitet er 0, siden tallet er positivt.
- Siden bias er 127, blir eksponentdelen 8-bit-representasjonen av $127_{10} + 10_{10}$. Det vil si $01111111_2 + 1010_2$ eller $10000000_2 + 1001_2$, som er 10001001_2.
- Signifikanden (mantissen) er 11011101001001 000000000
- Svar: $0\,\underbrace{1000100\ 1}_{\text{eksp.delen}}\ \underbrace{1101110\ 10010010\ 00000000}_{\text{signifikanden}}$

Oppgave 3.10

Finn 32-bit-datarepresentasjonen av følgende tall. Anta at 8 bit brukes for å lagre eksponenten og at eksponentbiasen er $2^7 - 1 = 127$.

(a) $5894,376$

(b) $-0,0387$

Løsning (a: $5894,376$)

- $5894,376_{10} =$
  $1011100000110,0110000001_2 =$
  $0,1011100000110\ \ 0110000001 \cdot 2^{13}$
- Eksponentdelen er $127_{10} + 13_{10} = 01111111_2 + 1101_2 = 10001100_2$.
- Signifikanden er 1011100 00011001 10000001
- Svar: $0\,\underbrace{1000110\ 0}_{\text{eksp.delen}}\ \underbrace{1011100\ 00011001\ 10000001}_{\text{signifikanden}}$

Løsning (b: $-0,0387$)

- Fortegnsbitet er 1.
- $0,0387_{10} =$
  $0,0000\ \ 1001111\ 01000001\ 11110010_2 =$
  $0,1001111\ 01000001\ 11110010_2 \cdot 2^{-4}$
- Eksponentdelen er
  $127_{10} + (-4_{10}) = 01111111_2 + (-0100_2) = 01111011_2$.
- Signifikanden er 1001111 01000001 11110010.
- Svar: $1\,\underbrace{0111101\ 1}_{\text{eksp.delen}}\ \underbrace{1001111\ 01000001\ 11110010}_{\text{signifikanden}}$

Oppgave 3.11

Gjenta oppgave 3.9 for en 32-bit datamaskin hvor 12 bit brukes til eksponenten og eksponentbiasen er $2^{11} - 1 = 2047$.

Løsning

- $1101110100,1001_2 = 0,11011101001001_2 \cdot 2^{10}$
- Fortegnsbitet er 0.
- Siden bias er 2047, blir eksponentdelen 12-bit-representasjonen av $2047_{10} + 10_{10}$. Det vil si $0111\ 1111\ 1111_2 + 1010_2$ eller $1000\ 0000\ 0000_2 + 1001_2$, som er $1000\ 0000\ 1001_2$.
- Signifikanden er på $32 - 13 = 19$ bit: 11011101001001 00000
- Svar: $0\,\underbrace{1000000\ 0100\ 1}_{\text{eksp.delen: 12 bit}}\underbrace{110\ 11101001\ 00100000}_{\text{signifikanden: 19 bit}}$

Oppgave 3.12

Finn, på desimalform, det omtrentlige området av positive reelle tall som kan representeres med 64 bit, hvor 11 bit brukes til eksponentdelen og eksponentbiasen er $2^{10} - 1 = 1023$

Løsning

- Eksponentdelen tar verdier fra 0 til $2^{11} - 1 = 2047$, så eksponentene som kan representeres går fra -1023 (som representeres med 11 nullere) til 1024 (som representeres med 11 enere).
- Signifikanden m er slik at $0,1_2 \leq m < 1_2$.
- Det minste tallet som kan representeres er derfor $0,1_2 \cdot 2^{-1023}$
- En øvre grense for hva som kan representeres er $1_2 \cdot 2^{1024}$.
- Hva om vi ønsker å uttrykke dette på normalisert form med base 10, det vil si på formen $m \cdot 10^e$, hvor e er et heltall og $0,1_{10} \leq m < 1_{10}$?

Løsning

- Vi ser først på $0,1_2 \cdot 2^{-1023} = 2^{-1024}$ og finner x slik at $10^x = 2^{-1024}$.

$$
\begin{aligned}
x &= \log_{10}(2^{-1024}) \\
&= -1024 \cdot \log_{10}(2) \\
&= -308,25472 \\
10^x &= 10^{-308,25472} \\
&= 10^{-0,25472} \cdot 10^{-308} \\
&= 0,55626278 \cdot 10^{-308} \\
&\approx 0,56 \cdot 10^{-308}
\end{aligned}
$$

- Det minste tallet som kan representeres er $0,56 \cdot 10^{-308}$.

Løsning

- Vi ser så på $1_2 \cdot 2^{1024}$ og finner x slik at $10^x = 2^{1024}$.

$$
\begin{aligned}
x &= \log_{10}(2^{1024}) \\
&= 1024 \cdot \log_{10}(2) \\
&= 308,25472 \\
10^x &= 10^{308,25472} \\
&= 10^{0,25472} \cdot 10^{308} \\
&= 1,7977115 \cdot 10^{308} \\
&= 0,17977115 \cdot 10^{309} \\
&\approx 0,18 \cdot 10^{309}
\end{aligned}
$$

- En øvre grense for hva som kan representeres er $0,18 \cdot 10^{309}$.

Oppgave 3.13

Meningen med følgende algoritme er at den skal skrive ut kubene av tall fra $0,1$ til 10 i steg på $0,1$. Forklar problemet som kan oppstå når algoritmen implementeres på en datamaskin.

1. $x \leftarrow 0,0$
2. **Repeat**
   2.1. $x \leftarrow x + 0,1$
   2.2. $x\_cubed \leftarrow x^3$
   2.3. Output x, x_cubed

   until $x = 10,0$

Løsning

Problemet ligger i at reelle tall ikke representeres nøyaktig og at vi ikke har noen garanti for at betingelsen $x = 10,0$ vil være oppfylt. En løsning er å erstatte $x = 10,0$ med $|x - 10,0| < 0,05$.

Oppgave 1, side 11, forelesningen 23.01.08

Finn en pseudokode for å beregne toerkomplementet til en sekvens av lengde 8, av lengde 16 og av lengde 32.

Løsning

Vi skisserer først en idé og skriver deretter pseudokode.

- Vi må undersøke sekvensen fra høyre og hoppe over alle nullere.
- En **For**-løkke er uegnet, siden vi må avbryte når vi treffer en ener.
- Det er naturlig å bruke en variabel i for å telle. Denne kan settes til lengden på sekvensen ved start.
- Siden første bit fra høyre kan være en ener, og vi da må avbryte, er en **While**-løkke velegnet.
- Når vi er ute av **While**-løkken, må vi sørge for at eneren ikke flippes.
- Deretter skal alle bit til venstre for eneren flippes. Vi må ta høyde for at det ikke trenger å være noen bit igjen, det vil si at i er 0.
- Derfor kan enda en **While**-løkke være passende her.

Løsning

1. Input $x_1, x_2, \ldots, x_n$ [n lik 8, 16 eller 32]
2. $i \leftarrow n$
3. **While** $1 \leq i$ **and** $x_i = 0$ **do**
   3.1. $i \leftarrow i - 1$
4. $i \leftarrow i - 1$
5. **While** $1 \leq i$ **do**
   5.1. $x_i \leftarrow 1 - x_i$
   5.2. $i \leftarrow i - 1$
6. Output $x_1, \ldots, x_n$

Oppgave 2, side 11, forelesningen 23.01.08

Hvis vi tar toerkomplementet av toerkomplementet av en sekvens $x_1 \cdots x_k$ av nuller og enere, får vi den opprinnelige sekvensen tilbake. Forklar hvorfor.

Løsning

Anta at den første eneren fra høyre fins på posisjon n, hvor $1 \leq n \leq k$. Når vi tar toerkomplementet, så vil x_1 til x_{n-1} flippes og x_n til x_k forbli urørt. Når vi tar toerkomplementet en gang til skjer det samme igjen: x_1 til x_{n-1} flippes og x_n til x_k forblir urørt. Siden x_1 til x_{n-1} blir flippet to ganger, får vi tilbake utgangspunktet.

$$
\begin{array}{rl}
 & x_1x_2x_3\ldots x_{n-1}x_nx_{n+1}x_{n+2}\ldots x_k \\
\text{toerkomplementet:} & \overline{x_1x_2x_3\ldots x_{n-1}}x_nx_{n+1}x_{n+2}\ldots x_k \\
\text{toerkomplementet:} & x_1x_2x_3\ldots x_{n-1}x_nx_{n+1}x_{n+2}\ldots x_k
\end{array}
$$