

Tentamen

EDAA05 Datorer i system

Lösningsförslag

2014–10–27, 14.00–19.00

1.
 - a) Moores lag säger att antalet transistorer på ett chip fördubblas ungefär vartannat år.
 - b) Naturen och dess lagar sätter slutligen gränsen för hur små transistorer man kan tillverka. Redan idag är man nere på tillverkningsdetaljer som bara är några tiotals atomdiametrar breda.

2.
 - a) Den sifferposition som står längst till vänster i ett tal, och därmed representerar det största värdet, utgör den *mest signifikanta* siffran. Omvänt representerar entalssiffran, som står längst till höger i talet, den *minst signifikanta* siffran. Exempel: i talet 10 är ettan den mest signifikanta siffran och nollan den minst signifikanta siffran.
 - b) 100001011 och 10B
 - c) 10010110
 - d) 206
 - e)

$$\begin{array}{r}
 \text{xx} \quad \text{x} \\
 11011100 \\
 - 1110011 \\
 \hline
 1101001
 \end{array}$$

3.
 - a) Påstående 1 och 3 är sanna medan påstående 2 och 4 är falska.
 - b) Med hjälp av kommandot "od" kan man skriva ut en representation av hur texten är kodad i detalj. Man kan se precis vilka teckenkoder som använts och ser även koderna för tecken som inte annars skulle synas vid en vanlig utskrift av texten. Genom att se hur olika tecken representeras – särskilt tecken som representeras olika i olika teckenkodningar – kan vi se hur texten kodats. Vi kan även direkt se hur man valt att representera radslut.
 - c) Unix och Linux brukar vanligen representera radslut med ett enkelt tecken med teckenkoden 10 (decimalt). Hexadecimalt 0A. Tecknet kallas ofta linefeed eller newline.

4. a) Sanningstabellerna blir:

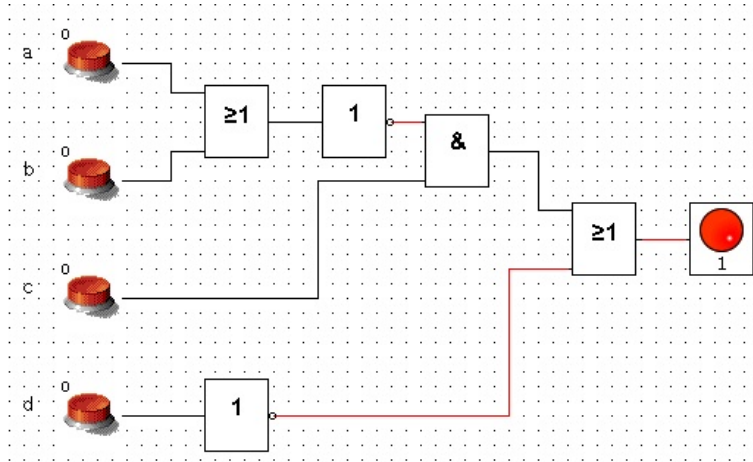
x	y	z	$\neg(x \vee y \wedge z)$	$\neg x \wedge \neg y \vee \neg z$
0	0	0	1	1
0	0	1	1	1
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	1
1	1	1	0	0

Eftersom tabellerna skiljer sig åt är uttrycken inte lika.

- b) $\neg(q \wedge p) \vee p =$ (De Morgan)
 $\neg q \vee \neg p \vee p =$ (Inverslagen)
 $\neg q \vee 1 =$ (Ett och nollagen)
 1

5. a) $x = \neg(a \vee b) \wedge (\neg b \vee \neg c)$

b) Uttrycket kan t.ex. realiseras på följande sätt (ritat i LogicSim):



6. a) $o = i \wedge s1$

$$s1' = \neg i \wedge (s1 \oplus s2) \vee i \wedge s1$$

$$s2' = i$$

b) Den färdiga maskinen kan realiseras på följande (ritat i LogicSim):

