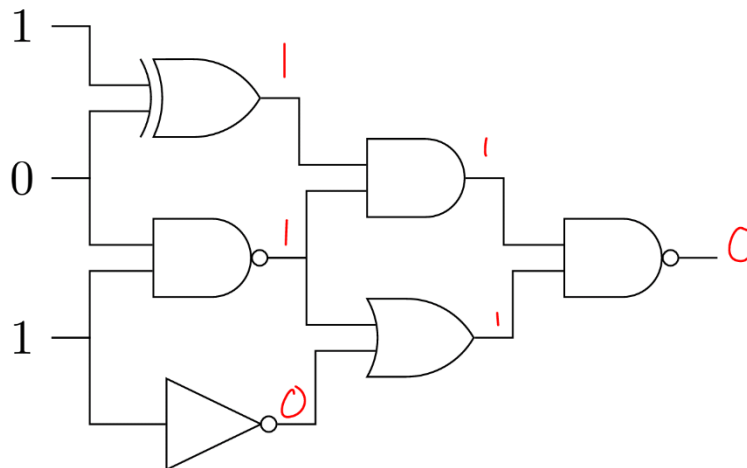


Uge 37

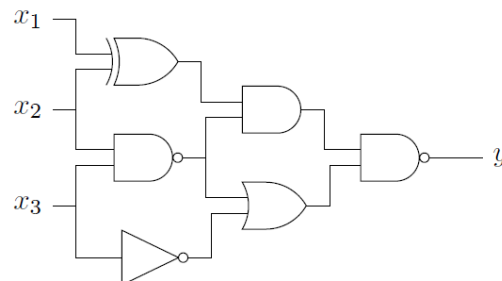
Ex1

Hvad er output af kredsløbet nedenfor?



Ex2

Opskriv hele sandhedstabellen for kredsløbet.



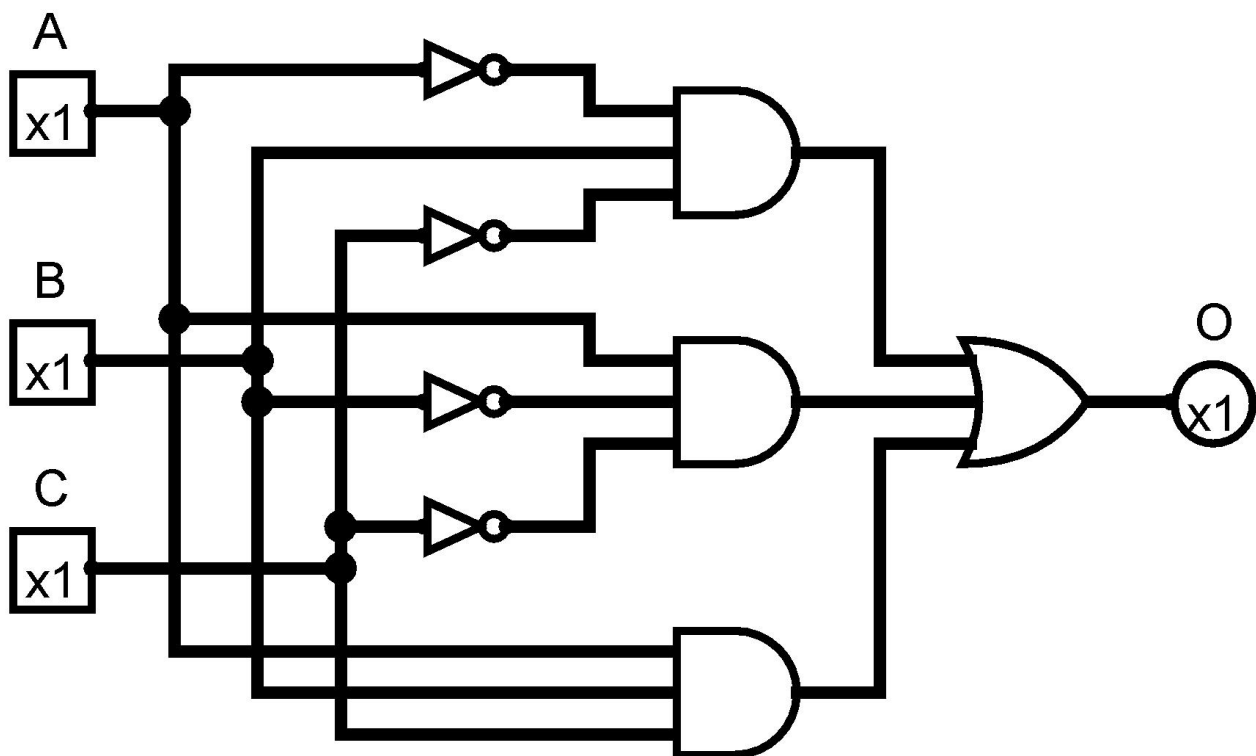
x_1	x_2	x_3	$x_1 \oplus x_2$	$\neg(x_2 \wedge x_3)$	$\neg x_3$	$(x_1 \oplus x_2) \wedge \neg(x_2 \wedge x_3)$	$\neg(x_2 \wedge x_3) \vee \neg x_3$	O
0	0	0	0	1	1	0	1	1
0	0	1	0	1	0	0	1	1
0	1	0	1	1	1	1	1	0
0	1	1	1	0	0	0	0	1
1	0	0	1	1	1	1	1	0
1	0	1	1	1	0	1	1	0
1	1	0	0	1	1	0	1	1
1	1	1	0	0	0	0	0	1

Er der evt en bedre måde at skrive op på? nummering?

Ex3

Lav et kredsløb med tre inputs, som har følgende sandhedstabel:

x_1	x_2	x_3	y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



Ex4

Konverter følgende tal i 2-talsystemet (binær repræsentation) til 10-talsystemet:

$$10101101_2 = 1 + 4 + 8 + 32 + 128 = 173$$

$$11111100_2 = 4 + 8 + 16 + 32 + 64 + 128 = 252$$

Ex5

Konverter følgende tal i 10-talsystemet til 2-talsystemet (binær repræsentation):

101:

$$\frac{101}{2} = 50 \rightarrow 1$$

$$\frac{50}{2} = 25 \rightarrow 0$$

$$\frac{25}{2} = 12 \rightarrow 1$$

$$\frac{12}{2} = 6 \rightarrow 0$$

$$\frac{6}{2} = 3 \rightarrow 0$$

$$\frac{3}{2} = 1 \rightarrow 1$$

$$\frac{1}{2} = 0 \rightarrow 1$$

256:

Genkender her at 256 er en 2^x :

$$256 = 2^8 = 100000000$$

Ex6

Two's complement

Repræsentationen two's complement har mange gode egenskaber (og vælges ofte):

- ▶ Fortegn kan ses af første bit.
- ▶ Simpel metode til at skifte fortegn findes.
- ▶ Den almindelige metode til addition virker også for negative tal. Ingen ekstra logiske kredsløb for disse (sparer transistorer på CPU).
- ▶ Subtraktion kan laves ved at vende fortegn og addere. Ingen logiske kredsløb for subtraktion (sparer transistorer på CPU).

Metode til at skifte fortegn:

Kopier bits fra højre til venstre til og med første 1-bit.
Resten af bits inverteres.

Eksempel: $6 = 0110 \rightarrow 1010 = -6$

Konverter følgende heltal i two's complement (8 bits) til 10-talsystemet:

10101010:

Negativ, vi skifter fortegn og udregner:

$$01010110 = 2 + 4 + 16 + 64 = -86$$

01010101:

positivt, kan bare udregne:

$$1 + 4 + 16 + 64 = 85$$

Ex7

Vend fortegnet på følgende tal i two's complement (8 bits):

$$10101010 = 01010110$$

$$01010101 = 10101011$$

Ex8

Konverter følgende heltal i 10-talsystemet til 8 bits two's complement (hint: find først two's complement af den positive udgave af tallet ved at konvertere det til 2-talsystemet, og vend så fortegnet):

-53:

konverterer til 2-talssystem (obs 8 bit):

00110101

vender fortegn:

→ 11001011

-126:

konverter til 2-tals system:

01111110

vender fortegn:

10000010

Ex9

Konverter følgende heltal i 10-talsystemet til excess notation (8 bits):

Negative heltal

Forskellige forslag for k bit heltal:

- A** Sign bit: Første bit = fortegn, resten af bits er binært talsystem.
- B** Excess: tæl op fra -2^{k-1} med binært talsystem.
- C** Two's complement: Første bit angiver $-(2^{k-1})$ i stedet for 2^{k-1} , ellers som binært talsystem.

Eksempel med $k = 4$:

	A	B	C		A	B	C
0111	-7	-1	7	1111	7	7	-1
0110	-6	-2	6	1110	6	6	-2
0101	-5	-3	5	1101	5	5	-3
0100	-4	-4	4	1100	4	4	-4
0011	-3	-5	3	1011	3	3	-5
0010	-2	-6	2	1010	2	2	-6
0001	-1	-7	1	1001	1	1	-7
0000	-0	-8	0	1000	0	0	-8

8 bits $\rightarrow -2^{8-1} = -128$

-53:

$$-128 + x = -53$$



Ligningen løses for x vha. CAS-værktøjet WordMat.

$$x = 75 \rightarrow 01001011$$

-126:

$$-128 + x = -126$$



Ligningen løses for x vha. CAS-værktøjet WordMat.

$$x = 2 \rightarrow 00000010$$

Flydende decimalpunkt

Flydende decimalpunkt (alias videnskabelig notation):

Tital-systemet:

$$-0.00000456 = (-1) \cdot 4.56 \cdot 10^{-6}$$

Fortegn: -1
 Eksponent: -6
 Mantis: 4.56

Total-systemet:

$$-0.01101_2 = (-1) \cdot 1.101_2 \cdot 2^{-2}$$

Der afsættes et fast antal bits til hver af de tre dele (fortegn, eksponent, mantisse). Vi bruger her: 1, 3 og 4 bits. Eksponent angives i excess notation. Mantisse fyldes om nødvendigt op med 0'er til højre.

Sign bit: 1 (sign bit 1 for negativt tal)
 Eksponent: 010 (-2 i excess notation (3 bits))
 Mantisse bits: $(1)101$ (første bit underforstået)

Så resultatet er $1|010|1010$.

Ex10

Konverter følgende tal i 2-talsystemet med fast decimalpunkt til 10-talsystemet:

$$11.101_2 = 1 + 2 + \frac{1}{2} + \frac{1}{8} = 3,625$$

$$1101.10101_2 = 1 + 4 + 8 + \frac{1}{2} + \frac{1}{8} + \frac{1}{16} \approx 13,688$$

Ex11

Konverter følgende tal i 2-talsystemet fra fast decimalpunkt til flydende decimalpunkt (med notationen fra slides for 8 bits flydende decimalpunktstal):

$$-0.00101_2 = -1.0.1 \cdot 2^{-3} = 1|001|0100$$

$$1100.0_2 = 1.1 \cdot 2^3 = 0|111|1000$$

Ex12

Konverter følgende tal i flydende decimalpunkt (med notationen fra slides for 8 bits flydende decimalpunktstal) til 2-talsystemet med fast decimalpunkt, og derefter til 10-talssystemet:

01110101

$$0|111|0101 = 1.10101 \cdot 2^3 = 1101.01 = 1 + 2 + 6 + \frac{1}{4} + \frac{1}{8} = 9,375$$

10001100

$$1|000|1100 = -1.11 \cdot 2^{-4} = -0.000111 = -\frac{1}{16} + \frac{1}{32} + \frac{1}{64} = -0,015625$$

Ex13

Hexadecimal notation

Gruppér bits i grupper af 4 (dvs. 16 forskellige muligheder):

01101010111001...

Vælg 16 tegn:

0111	7	1111	F
0110	6	1110	E
0101	5	1101	D
0100	4	1100	C
0011	3	1011	B
0010	2	1010	A
0001	1	1001	9
0000	0	1000	8

Giver en simpel/kort måde at beskrive bitstreng:

01101010111001... = 6AE...

NB: kan også bruges som cifre i et talsystem med grundtal 16.

Konverter følgende hexadecimal udtryk til bitstreng:

2:

$$= 0010$$

A1:

1010 0001

FF05:

1111 1111 0000 0101

001A:

0000 0000 0001 1010

Ex14

Konverter følgende bitstreng til hexadecimal udtryk:

1110:

$14 = E$

10101110:

$10\ 14 = AE$

0001 1101 0101 1111:

$1\ 13\ 5\ 15 = 1D5F$

Ex15

Konverter følgende hexadecimal udtryk, set som tal i 16-talsystemet, til tal i 10-talsystemet:

C:

$= 12$

1A:

$00011010 = 2 + 8 + 16 = 26$

F05:

$111100000101 = 1 + 4 + 256 + 512 + 1024 + 2048 = 3845$

Ex16

I denne opgave bruges instruktionssættet fra CPU-simulatoren beskrevet i slides om CPUer og maskinkode. Hvad er den hexadecimal notation for kommandoerne til at gøre følgende (numre på registre og RAM celler er nedenfor angivet hexadecimalt):

- Kopiere indholdet af register C til RAM celle 0A.

3C0A

- Lægge bitmønsteret 10110011 ind i register 2.

22B3

- Addere register 3 og 4, og lægge resultatet i register 5

5534

- Lave bit-wise XOR af register B og C.

9ABC

lægger her resultatet i regA, kunne være andet.

- Hoppe til instruktionen i RAM celle 14 hvis indholdet i register C er > end indholdet i register 0.

DC14

Ex17

I denne opgave bruges instruktionssættet fra CPU-simulatoren beskrevet i slides om CPUer og maskinkode. Forklar hvad følgende program gør:

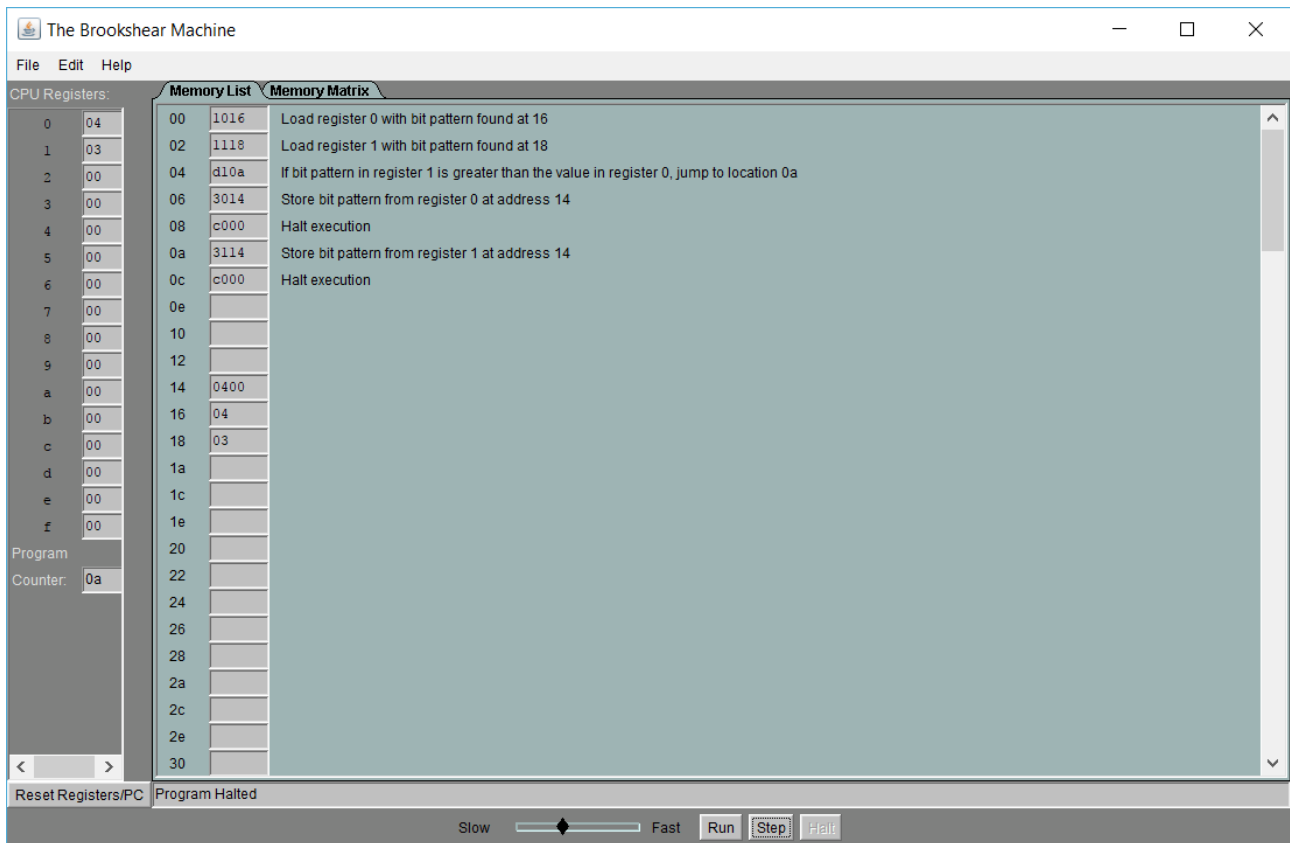
1110	COPY ram10 til reg1
1212	COPY ram12 til reg2
5112	ADD reg1 + reg2, res in reg1
1214	COPY ram14 til reg2
5112	ADD reg1+reg2, res in reg1
3118	COPY reg2 til ram18
C000	HALT

Lægger 3 tal sammen (ram10,12,14) gemmer i ram18.

Ex18

I denne opgave bruges instruktionssættet fra CPU-simulatoren beskrevet i slides om CPUer og maskinkode. Lav et program som læser to heltal fra RAM celle 16 og 18, og som skriver det største af dem i celle 14 (numre på RAM celler er her angivet hexadecimalt)

1016	COPY ram16 to reg0
1118	COPY ram18 to reg1
D10A	JUMP to RAM10 if reg1>reg0
3014	STORE reg0 at ram14
C000	HALT
3114	STORE reg1 at ram14
C000	HALT



--Code--

1016

1118

d10a

3014

c000

3114

c000

03

04

--EndCode--