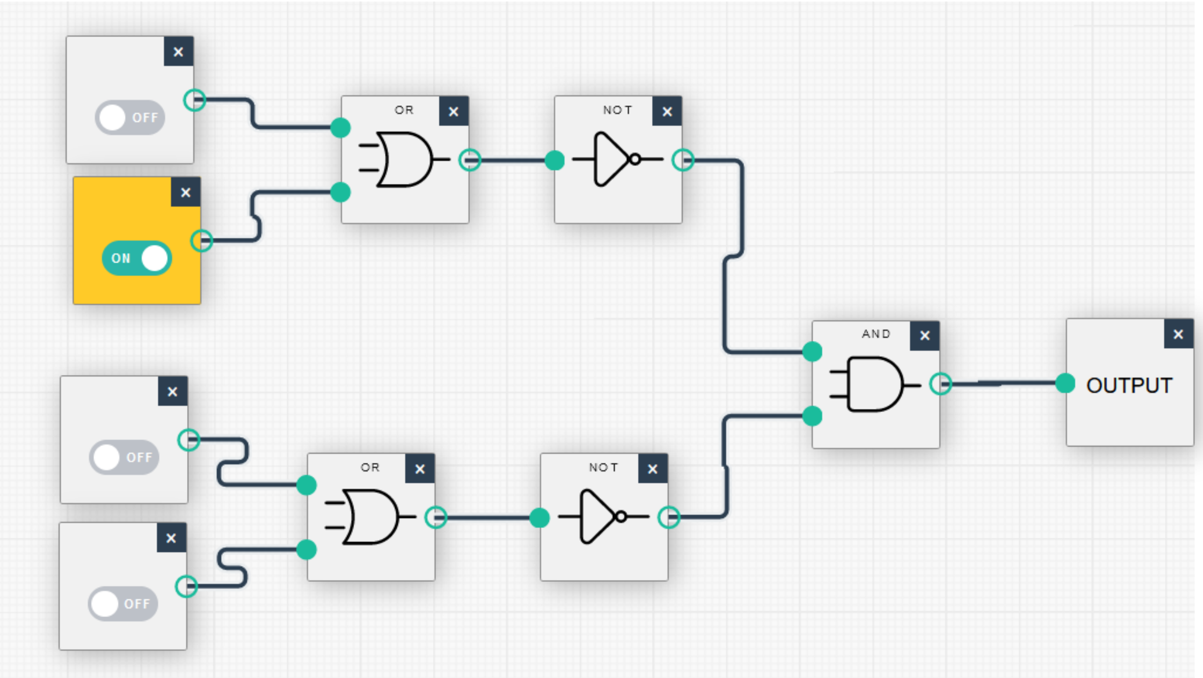
Assignment 9 – IKT-104

Logical gates

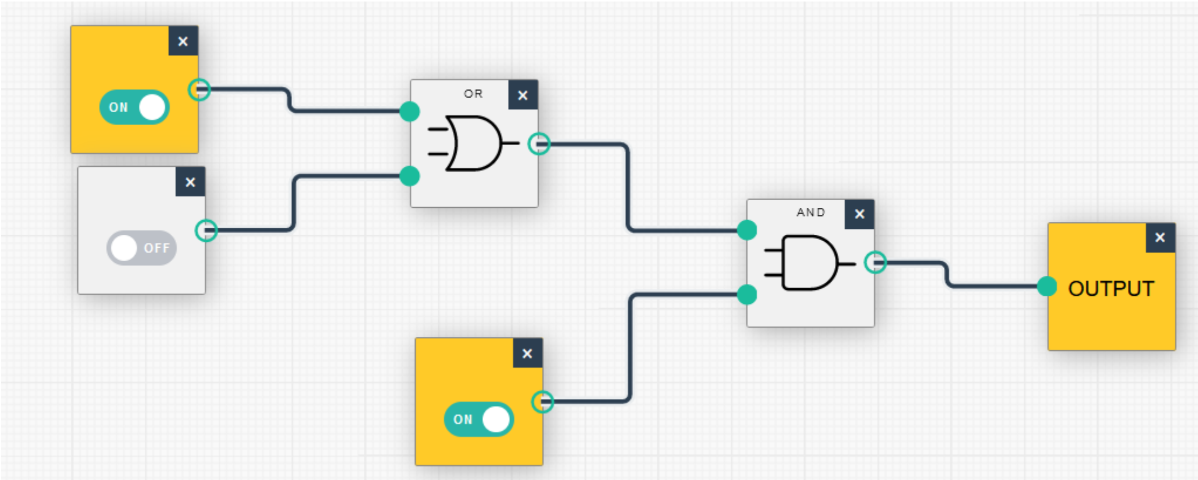
Task 1:

I dette tilfellet vil den øverste OR ha verdien 1, siden minst én av innputtene er 1, mens den nederste OR vil ha verdien 0, siden ingen av innputtene er skrudd på. Videre skiftes både øverste og nederste OR til den motsatte verdien. Dermed er innputtverdiene som kommer inn i AND 1 og 0, som vil si at resultatet av hele uttrykket er 0. Dette er også vist på bilde nedenfor.



Task 2:

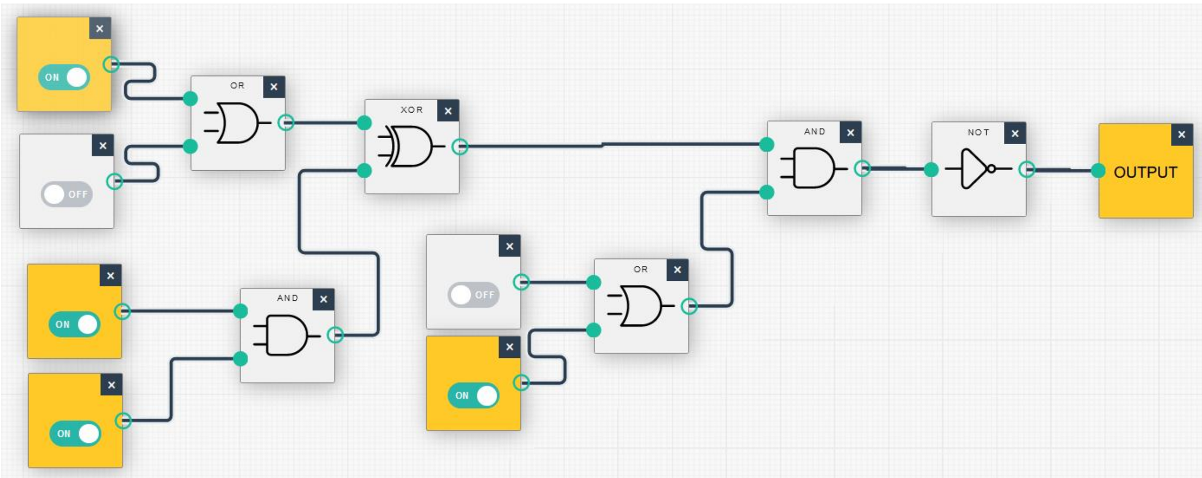
I dette tilfellet vil resultatet av OR være 1, siden en av innputtverdiene er 1. Videre er innputtverdiene som kommer inn i AND være 1 og 1, noe som vil si at resultatet av AND og hele uttrykket er 1. Dette er også demonstrert på figuren.



Task 3:

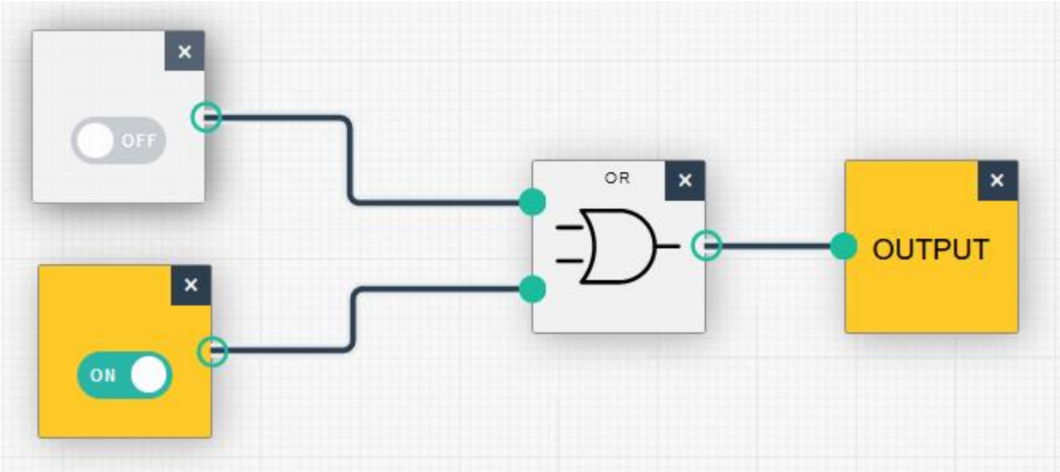
Innputtverdiene til OR i øvre venstre hjørnet er 1 og 0, som vil si at resultatet av OR er 1. Innputtverdiene til AND i nedre venstre hjørnet er 1 og 1, som gir resultat 1 av AND. Disse to verdiene (1 og 1) kommer inn i XOR, som gir verdien 0.

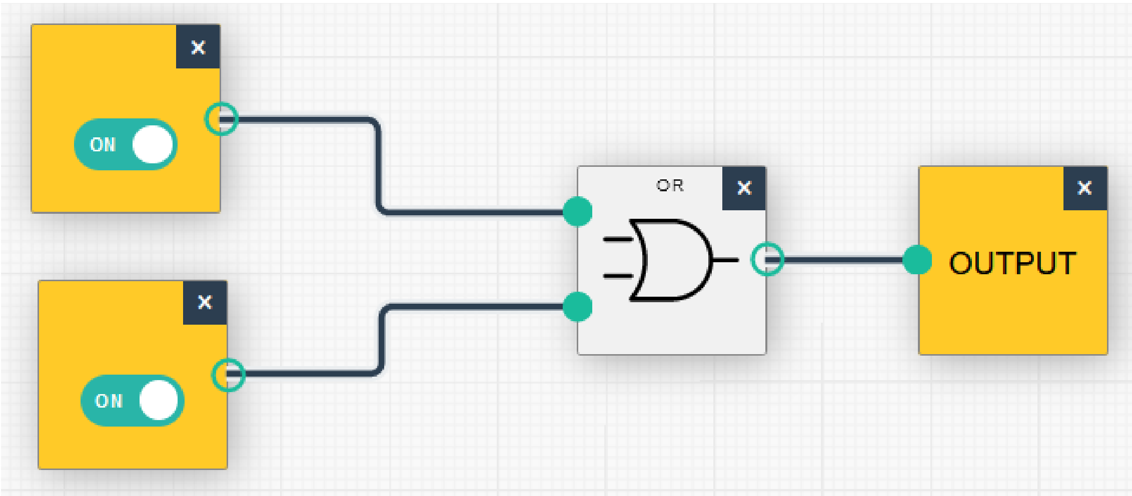
Innputtverdiene til OR i midten av figuren er 1 og 0, som gir resultat 1. Videre kommer resultatene fra OR og XOR (1 og 0) inn i AND og gir resultat 0. Sist blir denne verdien flippet med en NOT, som gir resultat 1, som er resultat av hele uttrykket.



Task 4:

Dette uttrykket avsluttes med en OR, der den ene innputtverdien er 1 og den andre er et resultat av en komplisert AND. Men i og med at den ene verdien er alltid 1, vil resultatet av hele uttrykket alltid være 1 uavhengig av den kompliserte AND-en. Dette er også vist på figurene nedenfor.





Som vi kan se fra bildene over, uavhengig av resultatet fra resultatet over blir output-en over det samme som er 1.

Numeral systems

Task 5

a. 97

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	1	1	0	0	0	0	1

Dermed ser vi at 97 i desimal tall det samme som 0b1100001 i binære tall form.

16^2	16^1	16^0
256	16	1
0	6	1

Dermed blir desimal 97 det samme som 0b1100001 i binære og 0x61 i heksadesimal, det gir:

97 = 0b1100001 = 0x61

b. ob1100 0001

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	1	0	0	0	0	0	1

Fra tebellen over kan vi se at tallet 0b1100 0001 blir 193 i desimal

16^2	16^1	16^0
256	16	1
0	12=C	1

Dermed ser vi også fra tabellen at 193 i desimal blir C1 i heksadesimal tall, dette gir svaret:

193 = 0b1100 0001 = 0xC1

c. 0xAF

16^2	16^1	16^0
256	16	1
0	A=10	F=15

Fra tabellen over, 0xAF = 175 i desimal tall.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	0	1	0	1	1	1	1

I binære tall blir 175 = 0b1010 1111 som gir,

175 = 0b1010 1111 = 0xAF

d. 256

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0	0

Fra tabellen ser vi at 256 = 0b1 0000 0000.

16^2	16^1	16^0
256	16	1
1	0	1

Siden $16^2 = 256$, ser vi at i heksadesimale får vi 256 = 0x100, som gir svaret:

256 = 0b1 0000 0000 = 0x100

e. 0xFF

16^2	16^1	16^0
256	16	1
0	F = 15	F = 15

Dermed ser vi at i desimaltall blir 0xFF = 15*16+15*1 = 255

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1
0	1	1	1	1	1	1	1	1

Fra tabellen over ser vi at 255 blir i binære tall 0b1111 1111, som gir svaret:

255 = 0b1111 1111 = 0xFF

f. 0b0011 0111

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1
0	0	0	1	1	0	1	1	1

Fra tabellen over ser vi at 0b0011 0111 = 55 i desimal tall.

16^2	16^1	16^0
256	16	1
0	3	7

Fra tabellen ser vi at 55 er det samme som 0x37 i heksadesimaltall, som gir svaret,

55 = 0b0011 0111 = 0x37

Task 6

A) 10 | 20 =

= 1010 | 0001 0100 = 30 dec

0000 1010 (10 dec)
| 0001 0101 (20 dec)
= 0001 1111 (30 dec)

B) (0x AA | 0x 55) & 0x62 =

= (1010 1010 | 0101 0101) & 0110 0010 = 98 dec

1010 1010 (170 dec)
| 0101 0101 (85 dec)
= 1111 1111 (255 dec)

1111 1111 (255 dec)
& 0110 0010 (98 dec)
= 0110 0010 (98 dec)

C) ~((0x75 | 0x16) & (0x0C | 0x48)) =

(0x75 | 0x 16) = 0111 0101 | 0001 0110 = 0111 0111

0111 0101 (117 dec)
| 0001 0110 (22 dec)
= 0111 0111 (119 dec)

(0x0C | 0x48) = 1100 | 0100 1000 = 0100 1100

0000 1100 (12 dec)
| 0100 1000 (72 dec)
= 0100 1100 (76 dec)

(0x75 | 0x 16) & (0x 0C | 0x 48) =

= 0111 0111 & 0100 1100

0111 0111 (119 dec)
& 0100 1100 (76 dec)
= 0100 0100 (68 dec)

~((0x75 | 0x16) & (0x0C | 0x48)) = ~(0100 0100) =

= 1011 1011 = 187 dec

Task 7

Left Shift

a. 1 << 3

1 er i desimaltall og ved å bytte den til binære tall for v 0b0001. Ved en 3-gange «Left Shift» blir den til 0b1000 (=16 dec) som blir **0x8 i heksadesimal tall**.

b. 2 << 3

2 er i desimaltall og på binærtall er det 0b0010. Ved en 3-gange «Left Shift» blir den 0b0001-0000 (=16 dec) som er det samme som **0x10 i heksadesimal tall**.

c. 15 << 4

15 er i desimaltall og ved å skrive den på binære for vi 0b1111. Ved en 4-gange «Left Shift» blir den til 0b1111 0000 (=240 dec) som er det somme som **0xF0 i heksadesimal tall**.

d. 0xF0 << 4

16^2	16^1	16^0
256	16	1
0	F=15	0

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1
0	1	1	1	1	0	0	0	0

Fra tabellene over ser vi at 0xF0 er 240 i desimaltall og det er det samme som 0b1111 0000 idesimatall. Ved en 4-gange «Left Shift» får vi 0b1111 0000 0000 (=3840 dec) som er det samme som **0xF00 i heksadesimale tall**.

Right Shift

e. 128 >> 1

128 er det same som 0b1000 0000 binære tall. Ved en 1-gange «Right Shift» blir den 0b100 0000 (=64 dec). Dette er det samme som **0x40 i heksadesimale tall**.

f. 48 >> 2

48 er det samme som 0b0011 0000 i binære tall. Ved en 2-gange «Right Shift» blir den ob1100 (=12 dec). Det er det samme som **0xC i heksadesimale tall.**

g. 240 >> 4

240 er det samme som 0b1111 0000 i binære tall. Ved en 4-gange «Right Shift» blir den 0b1111 (=15 dec). Vi kan se at denne operasjonen er det samme som reversen av Task 7c. Det er det samme som **0xF i heksadesimale tall.**

h. 0xF0 >> 4

16^2	16^1	16^0
256	16	1
0	F=15	0

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1
0	1	1	1	1	0	0	0	0

Fra tabellene over kan vi se at 0xF0 er det samme som 240 i desimal tall og 0b1111 0000 i binær tall. Dermed kan vi se at det blir den samme operasjonen som i oppgaven over (Task 7g). Ved en 4-gange «Right Shift» blir den til 0b1111 (=15 dec). Det er det samme som **0xF i heksadesimale tall.**