

Et vilkårlig 4-bits tall $abcd$ ganget med 10 ser slik ut:

```

abcd * 1010
-----
0000
abcd
0000
abcd
-----
*****cd0

```

Det betyr at funksjonsuttrykkene for F, G og H blir:

F: c

G: d

H: 0

Resten løste jeg ved hjelp av sannhetsverditabellen og karnaughdiagrammer:

A: $abd + abc$

ab \ cd	00	01	11	10
00				
01			1	
11			1	
10			1	

B: $ab' + ac'd' + a'bcd$

ab \ cd	00	01	11	10
00			1	1
01				1
11		1		1
10				1

C: $a'bc' + bc'd' + ab'c + a'bd'$

ab \ cd	00	01	11	10
00		1	1	
01		1		
11				1
10		1		1

D: $a'b'c + ab'c' + a'cd' + ac'd' + a'bc'd + abcd$

ab \ cd	00	01	11	10
00			1	1
01		1		1
11	1		1	
10	1	1		

E: $b'd + bd'$

ab \ cd	00	01	11	10
00		1	1	
01	1			1
11	1			1
10		1	1	

F, G og H som skrevet tidligere er:

F = c, G = d, H = 0

(men her er altså karnaughdiagrammene for disse.. just in case.

F: c

ab \ cd	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

G: d

ab \ cd	00	01	11	10
00				
01	1	1	1	1
11	1	1	1	1
10				

Inngang Sign bestemmer om kretsen velger å gå den *vanlige** veien (når Sign er 0 (10 er positiv)) eller om 2er-komplement skal brukes for å få ut et negativt resultat. Selektorene sjekker hvert bit. Er Sign høy, skjer konverteringen (2er kompl. tar 8 bit inn og kan gi ut 9). 2er-kompl: Invertere hvert bit og legge til 1. OR-portene rett før utgangene sørger for at bare ett signal går gjennom.

Sannhetsverditabell for (+10):
Sign = 0 (trenger bare 8 bit)

a	b	c	d	A	B	C	D	E	F	G	H
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	0	1	0
0	0	1	0	0	0	0	1	0	1	0	0
0	0	1	1	0	0	0	1	1	1	1	0
0	1	0	0	0	0	1	0	1	0	0	0
0	1	0	1	0	0	1	1	0	0	1	0
0	1	1	0	0	0	1	1	1	1	0	0
0	1	1	1	0	1	0	0	0	1	1	0
1	0	0	0	0	1	0	1	0	0	0	0
1	0	0	1	0	1	0	1	1	0	1	0
1	0	1	0	0	1	1	0	0	1	0	0
1	0	1	1	0	1	1	0	1	1	1	0
1	1	0	0	0	1	1	1	1	0	0	0
1	1	0	1	1	0	0	0	0	0	1	0
1	1	1	0	1	0	0	0	1	1	0	0
1	1	1	1	1	0	0	1	0	1	1	0

Sannhetsverditabellen
fra Selector: (v = vanlig)

A	S	2's	v
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	0

*vanlig vei vil si multiplikasjon med +10.
Selektorene sjekker om Sign er høy eller lav.

Sannhetsverditabell for (-10): Sign = 1 (trenger 9 bit)
S = Sign, Sb = Sign_bit

S	a	b	c	d	Sb	A	B	C	D	E	F	G	H
1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	1	1	1	0	1	1	0
1	0	0	1	0	0	1	1	1	0	1	1	0	0
1	0	0	1	1	0	1	1	1	0	0	0	1	0
1	0	1	0	0	0	1	1	0	1	1	0	0	0
1	0	1	0	1	0	1	1	0	0	1	1	1	0
1	0	1	1	0	0	1	1	0	0	0	1	0	0
1	0	1	1	1	0	1	0	1	1	1	0	1	0
1	1	0	0	0	0	1	0	1	1	0	0	0	0
1	1	0	0	1	0	1	0	1	0	0	1	1	0
1	1	0	1	0	0	1	0	0	1	1	1	0	0
1	1	0	1	1	0	1	0	0	1	0	0	1	0
1	1	1	0	0	0	1	0	0	0	1	0	0	0
1	1	1	0	1	0	0	1	1	1	1	1	1	0
1	1	1	1	0	0	0	1	1	1	0	1	0	0
1	1	1	1	1	0	0	1	1	0	1	0	1	0