Øving 3

IELET1002 - Datateknikk

Gunnar Myhre, BIELEKTRO

19. oktober 2021

1 Oppgåve 1

Setter opp funksjonstabell for 4-bit gray-kode

Indeks	$a_3a_2a_1a_0$	$g_3g_2g_1g_0$
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Frå dette finner vi sum av standardprodukt

- $g_3 = \Sigma(8, 9, 10, 11, 12, 13, 14, 15)$
- $g_2 = \Sigma(4, 5, 6, 7, 8, 9, 10, 11)$
- $g_1 = \Sigma(2, 3, 4, 5, 10, 11, 12, 13)$

- $g_0 = \Sigma(1, 2, 5, 6, 9, 10, 13, 14)$

Teikner opp Karnaugh-diagram

a_3a_2 \ a_1a_0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

$g_3 = a_3$

a_3a_2 \ a_1a_0	00	01	11	10
00	0	0	0	0
01	1	1	1	1
11	0	0	0	0
10	1	1	1	1

$g_2 = \overline{a_3}a_2 + a_3\overline{a_2}$

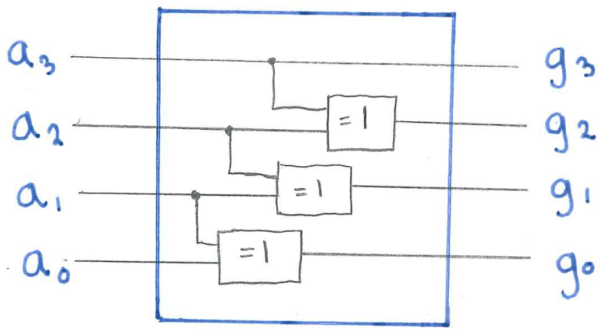
a_3a_2 \ a_1a_0	00	01	11	10
00	0	0	1	1
01	1	1	0	0
11	1	1	0	0
10	0	0	1	1

$g_1 = \overline{a_2}a_1 + a_2\overline{a_1}$

a_3a_2 \ a_1a_0	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	1	0	1
10	0	1	0	1

$g_2 = \overline{a_1}a_0 + a_1\overline{a_0}$. Forenkler uttrykka vha. XOR sidan $\overline{A}B + A\overline{B} = A \oplus B$

- $g_3 = a_3$
- $g_2 = a_3 \oplus a_2$
- $g_1 = a_2 \oplus a_1$
- $g_0 = a_1 \oplus a_0$



4-bit binær → gray med
4 XOR-gates

2 Oppgåve 2

2.1 a)

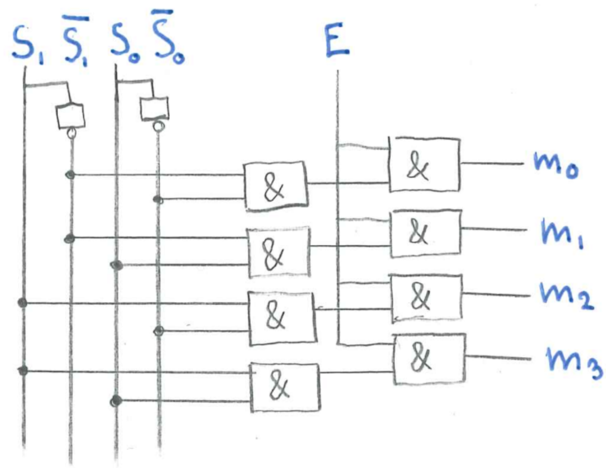
Først setter vi opp funksjonstabell

S_1	S_0	$m_3m_2m_1m_0$
0	0	0 0 0 1
0	1	0 0 1 0
1	0	0 1 0 0
1	1	1 0 0 0

uttrykka er gitt ved

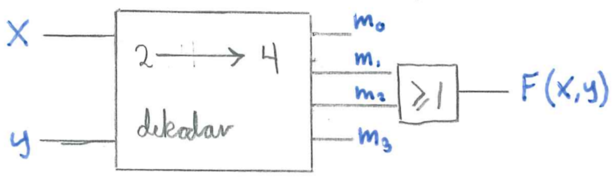
- $m_0 = \bar{S}_1\bar{S}_0$
- $m_1 = \bar{S}_1 S_0$
- $m_2 = S_1\bar{S}_0$
- $m_3 = S_1 S_0$

vi kan teikne dette som eit logisk skjema vha. åtte AND-portar

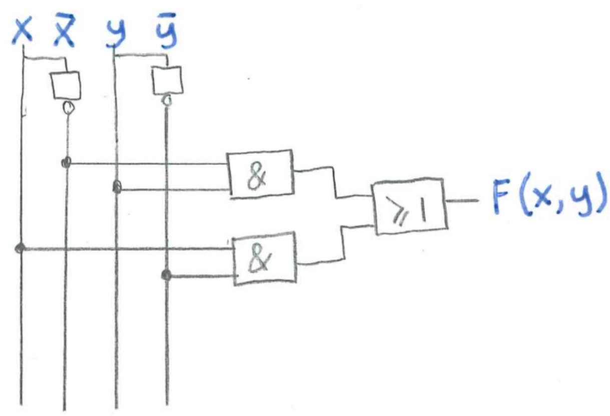


2.2 b)

Funksjonen $F(x, y) = \Sigma(1, 2)$ kan implementerast med ein XOR-port. Men dersom vi ønsker å bruke ein 2→4-dekodar kan vi gjere slik som dette:

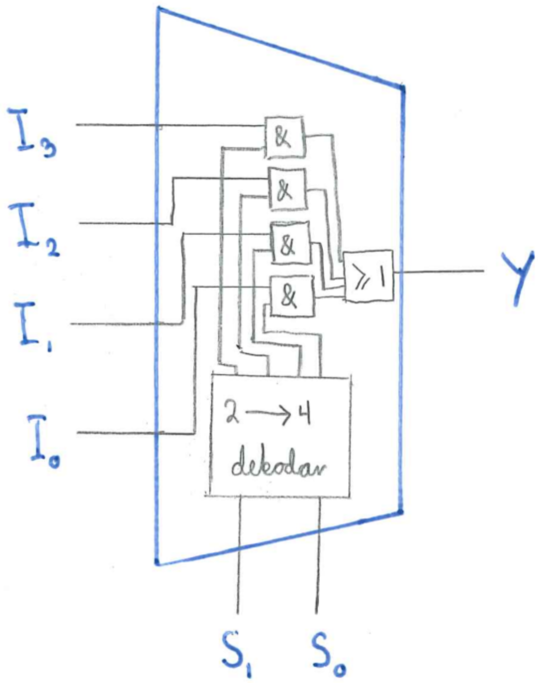


Vi kan også implementere denne funksjonen med to NOT-portar, to AND-portar og éin OR-port



3 Oppgåve 3

Antar at det her er snakk om fire databit og ikkje fire addressebit for multipleksaren. Då kan vi implementere den vha. ein 2→4-dekodar slik som dette:



Dekodaren som styrt av S_1 og S_0 velger kva for input I som har moglegheit til å slippe igjennom til utgongen Y

4 Oppgåve 4

4.1 a)

Setter først opp ufullstendig funksjonstabell for ein 4→2-kodar

$I_0I_1I_2I_3$	S_1	S_0
1000	0	0
0100	0	1
0010	1	0
0001	1	1

4.2 b)

For å potensielt forenkle logikken i kodaren setter vi opp fullstendig funksjonstabell

$I_0I_1I_2I_3$	S_1	S_0
0000	-	-
0001	1	1
0010	1	0
0011	-	-
0100	0	1
0101	-	-
0110	-	-
0111	-	-
1000	0	0
1001	-	-
1010	-	-
1011	-	-
1100	-	-
1101	-	-
1110	-	-
1111	-	-

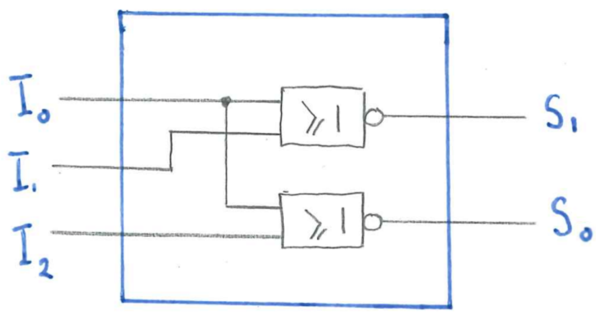
Dette kan vi forenkle vha. Karnaugh-diagram:

I_0I_1 \ I_2I_3	00	01	11	10
00	-	1	-	0
01	1	-	-	-
11	-	-	-	-
10	-	0	-	-

Finner $S_0 = \bar{I}_0\bar{I}_2$

I_0I_1 \ I_2I_3	00	01	11	10
00	-	1	-	1
01	0	-	-	-
11	-	-	-	-
10	-	0	-	-

Finner $S_0 = \bar{I}_0\bar{I}_2$ og $S_1 = \bar{I}_0\bar{I}_1$. Desse uttrykka kan vi igjen forenkle med DeMorgan: $S_0 = \overline{I_0 + I_2}$, $S_1 = \overline{I_0 + I_1}$. Funksjonsskjemaet kan vi teikne på denne måten vha. to NOR-portar:



5 Oppgåve 5

Sekvenslogisk skjema for tre ulike vipper: MS-JK, positivt flanketrigga JK og negativt flanketrigga JK: