Dr. Aiman El-Maleh

COE 561

HW#3

Q1

$$x = a\bar{d} + \bar{a}\bar{b} + \bar{a}\bar{d} + bc + b\bar{d} + ac$$

$$y = a + b$$

$$z = \bar{a}c + \bar{a}\bar{d} + \bar{b}\bar{c} + \bar{b}\bar{d} + e$$

$$\mu = \bar{a}c + \bar{a}d + \bar{b}d + \bar{e}$$

(a) Substitute y into fx by the algebraic division f_x / f_y

$i=1$; $c_1^B = a$

$D = \{ a\bar{d}, ac \}$

$D_1 = \{ \bar{d}, c \}$

$Q = \{ \bar{d}, c \}$

$i=2$; $c_2^B = b$

$D = \{ bc, b\bar{d} \}$

$D_2 = \{ c, \bar{d} \}$

$Q = Q \cap D_2 = \{ c, \bar{d} \}$

$R = \{ a\bar{d}, \bar{a}\bar{b}, \bar{a}\bar{d}, bc, b\bar{d}, ac \} - Q \times B$

$= \{ a\bar{d}, \bar{a}\bar{b}, \bar{a}\bar{d}, bc, b\bar{d}, ac \} - \{ ac, a\bar{d}, bc, b\bar{d} \}$

$= \{ \bar{a}\bar{b}, \bar{a}\bar{d} \}$

$\Rightarrow x = y \cdot (c + \bar{d}) + \bar{a}\bar{b} + \bar{a}\bar{d}$

$= yc + y\bar{d} + \bar{a}\bar{b} + \bar{a}\bar{d}$ (in SOP form)

$y = a + b$

# of literals saved = 4 literals

$= nl - n + \sum |b_i|$ (since y is already implemented so no $-l$).

$= 2*2 - 2 + 2 = 4$ $b_i \in$ cofactor

Note that SIS produces the same result in SOP.

(ii) We will compute first the Kernels and co-Kernels of z using the procedure KERNELS. We assume that the variables are ordered in lexicographic order $\{a, b, c, d, e\}$

$i = 1$ (a)

Cubes$(z, a) = \bar{a}\bar{c} + \bar{a}\bar{d} \geq 2$

$C = \bar{a}$

Then, the procedure KERNELS$(\bar{c}+\bar{d}, 2)$ will be called and will return $\bar{c}+\bar{d}$ as a Kernel.

$i = 2$ (b)

Cubes$(z, b) = \bar{b}\bar{c} + \bar{b}\bar{d} \geq 2$

$C = \bar{b}$

Then, the procedure KERNELS$(\bar{c}+\bar{d}, 3)$ will be called and will return $\bar{c}+\bar{d}$ as a Kernel.

$i = 3$ (c)

Cubes$(z, c) = \bar{a}\bar{c} + \bar{b}\bar{c} \geq 2$

$C = \bar{c}$

The Kernel $\bar{a}+\bar{b}$ will then be returned.

$i = 4$ (d)

Cubes$(z, d) = \bar{a}\bar{d} + \bar{b}\bar{d} \geq 2$

$C = \bar{d}$

The Kernel $\bar{a}+\bar{b}$ will then be returned.

$i = 5$ (e)

Cubes$(z, e) = e < 2 \Rightarrow$ no Kernels

Since z is cube-free it will also be a Kernel.

So, the Kernels and co-Kernels of z are:

Kernel	co-Kernel
$\bar{c}+\bar{d}$	$\{\bar{a}\}, \{\bar{b}\}$
$\bar{a}+\bar{b}$	$\{\bar{c}\}, \{\bar{d}\}$
$\bar{a}\bar{c}+\bar{a}\bar{d}+\bar{b}\bar{c}+\bar{b}\bar{d}+e$	$\{\}$

Next, we compute the Kernel and co-Kernels of $u = \bar{a}c + \bar{a}d + \bar{b}d + \bar{e}$

$i = 1$ (a)

Cubes $(u, a) = \bar{a}c + \bar{a}d \geq 2$

$C = \bar{a}$

This produces the Kernel $c+d$.

$i = 2$ (b)

Cubes $(u, b) = \bar{b}d < 2 \Rightarrow$ no Kernels

$i = 3$ (c)

Cubes $(u, c) = \bar{a}c < 2 \Rightarrow$ no Kernels

$i = 4$ (d)

Cubes $(u, d) = \bar{a}d + \bar{b}d \geq 2$

$C = d$

This produces the Kernel $\bar{a}+\bar{b}$.

$i = 5$ (e)

Cubes $(u, e) = \bar{e} < 2 \Rightarrow$ no Kernels

Since u is cube-free, it is also a Kernel.

So, the Kernels and co-Kernels of μ are:

Kernel	co-Kernel
$c + d$	$\{\bar{a}\}$
$\bar{a} + \bar{b}$	$\{d\}$
$\bar{a}c + \bar{a}d + \bar{b}d + \bar{e}$	$\{\}$

As can be seen, the Kernel $\bar{a}+\bar{b}$ is the only common Kernel between z and μ.

The value of the Kernel $= n * l - n - l + \sum |\text{co-Kernel}|$

$= 3 * 2 - 3 - 2 + (1+1+1) = 6 - 5 + 3 = 4$

So, 4 literals will be saved by extracting this Kernel. Let us assume that this Kernel is assigned node t, then after extraction we have:

$$t = \bar{a} + \bar{b}$$
$$z = t\bar{c} + t\bar{d} + e$$
$$\mu = td + \bar{a}c + \bar{e} \qquad (12 \text{ literals})$$

Note that the number of literals of z and μ before extraction is $9 + 7 = 16$ literals.

Running the command gkx produces the same result obtained.

Q2 $x = abcf + efc + de$

$y = acdef + bdef$

$z = bcd + acf$

(i)

		var	a	b	c	d	e	f
cube	ID	R/C	1	2	3	4	5	6
abcf	x	1	1	1	1	0	0	1
efc	x	2	0	0	1	0	1	1
de	x	3	0	0	0	1	1	0
acdef	y	4	1	0	1	1	1	1
bdef	y	5	0	1	0	1	1	1
bcd	z	6	0	1	1	1	0	0
acf	z	7	1	0	1	0	0	1

Prime Rectangle	cube
$(\{1,4,7\},\{1,3,6\})$	acf
$(\{1,6\},\{2,3\})$	bc
$(\{5,6\},\{2,4\})$	bd
$(\{1,5\},\{2,6\})$	bf
$(\{4,6\},\{3,4\})$	cd
$(\{2,4\},\{3,5,6\})$	cef
$(\{3,4,5\},\{4,5\})$	de
$(\{4,5\},\{4,5,6\})$	def
$(\{2,4,5\},\{5,6\})$	ef

Note that only prime rectangles with $|R| \geq 2$ are considered.

(11)

we need to compute the value of each cube to extract the best one.

The value of a cube = $n * l - n - l$

cube	value
acf	$3*3-3-3=3$
bc	$2*2-2-2=0$
bd	$2*2-2-2=0$
bf	$2*2-2-2=0$
cd	$2*2-2-2=0$
cef	$2*3-2-3=1$
de	$3*2-3-2=1$
def	$2*3-2-3=1$
ef	$3*2-3-2=1$

So, we extract the cube acf and this will save 3 literals. The resulting network is:

$t = acf$
$x = tb + efc + de$
$y = tde + bdef$
$z = bcd + t$ (21 literals)

Note that the number of the literal in the original network is 24 literals.

Note that also the cube de can be extracted and will save one literal. The resulting network is:

$u = de$
$t = acf$
$x = tb + efc + u$
$y = tu + ubf$
$z = bcd + t$ (20 literals)

The command gcx produces the same result.

Q3 $X = bd + cd + be + ce + afd + afe + abg + acg + afg$

(a) Quick Factorization:

First, we find the first level-0 kernel.

$L(a) = 5 > 1$

$C = a$

Then, we will call the procedure One-level-0-Kernel on $\frac{X}{a} = fd + fe + bg + cg + fg$

$L(f) = 2 > 1$

$C = f$

$\Rightarrow$ The first level-0 kernel obtained will be $K = d + e$.

$(h, r) = \text{Divide}(X, K = d+e)$

$h = b + c + af$

$r = abg + acg + afg$

Then, calling Quick Factor on h will return h as is since $L \leq 1$ for all literals.

Next, we call Quick Factor on r.

$L(a) = 3 > 1$

$C = ag$

we call One-level-0-Kernel on $\frac{r}{ag} = b + c + f$

This will return the kernel $K = b + c + f$

$(h, r) = \text{Divide}(r, K = b + c + f)$

$h = ag \qquad r = \phi$

So, X is factored as follows:

$X = (d + e)(b + c + af) + (b + c + f)(ag)$

11 literals

The solution returned by SIS tool using the command factor -q is

$$x = (e+d)(c+b) + a(f(e+d) + g(f+c+b))$$

12 literals

This is different from the solution we obtained because Quick factoring in SIS uses the first Kernel found and not the first level-0 Kernel found.

Let us apply Quick factoring using the first Kernel found.

Then, the first Kernel found will be

$K = fd + fe + bg + cg + fg$

$h = a$

$r = bd + cd + be + ce$

Then, we need to factor both K and r in this case. Let us first factor K.

$L(f) = 3 > 1$

$C = f$

$K = d + e + g$

$h = f$

$r = bg + cg$

Then, factoring $r = bg + cg$

$L(g) = 2 > 1$

$C = g$

$h = g$

$r = b + c$

Then, we factor $r = bd + cd + be + ce$

$L(b) = 2 > 1$

$c = b$

$k = d + e$

$h = b + c$

$r = \phi$

So, the factor obtained is:

$x = a(f(d+e+g) + g(b+c)) + (d+e)(b+c)$

12 literals

Note that the solution we obtained has the same cost as the one obtained by SIS but it is slightly difference. The difference is due to the order of selection of the literals. We assumed lexicographic order in our solution. Note that if we select the literal g before literal f in finding the cofactor of $k = fd + fe + bg + cg + fg$ we will obtain the same solution obtained by SIS.

(ii) To find a good factor, we need to find all the Kernel and compute their value and select the best Kernel (i.e. with highest value) as a divisor.

The Kernels of x and their values are shown below:

Kernel	co-Kernel	value
$d+e+g$	$\{af\}$	$1*3-1-3+2=1$
$b+c+f$	$\{ag\}$	$1*3-1-3+2=1$
$d+e+ag$	$\{b\}, \{c\}$	$2*4-2-4+1+1=4$
$b+c+af$	$\{d\}, \{e\}$	$2*4-2-4+1+1=4$

Then, we can select either $d+e+ag$ or $b+c+af$ as a divisor since both have the same value. Let us use $K = d+e+ag$

$h = (b+c)$

$r = afd + afe + afg$

Next, we need to find the best divisor for $r = afd + afe + afg$

There is only one Kernel $d+e+g$.

So, based on Good factoring we obtain

$x = af(d+e+g) + (b+c)(d+e+ag)$

11 literals.

Using the command factor -g, we obtain the same result.

(iii) Quick Decomposition

Quick decomposition is like Quick factoring except the function is decomposed into multiple nodes by creating internal nodes. So, the solution obtained will be:

[1] = d + e
[2] = b + c + af
[3] = b + c + f
x = [1][2] + [3] ag $\overset{14}{=}$ literals

Note that the SIS command decomp -q produces the same result since SIS uses in Quick Decomposition a divisor based first level-0 kernel found.

(iv) Good Decomposition

Good decomposition is like good factring except that internal nodes will be created. So, the solution obtained will be:

[1] = b + c
[2] = d + e + ag
[3] = d + e + g
x = af [3] + [1][2] $\overset{14}{=}$ literals

The command decomp -g produces similar result.

(v) The command fx produces the result:
[1] = b + c [2] = d + e [3] = [1] + f
x = [1][2] + [2] af + [3] ag $\overset{14}{=}$ literals

Q4 $x = \bar{a}\bar{b}cd + \bar{a}\bar{b}\bar{c}\bar{d} + ac\bar{d} + a\bar{c}d + bc\bar{d} + b\bar{c}d$

$y = \bar{a}\bar{b}c + \bar{a}\bar{b}d + a\bar{c}\bar{d} + b\bar{c}\bar{d}$

32 literals

Let us first compute the double-cube divisors of x

Double-cube divisor	Base
$cd + \bar{c}\bar{d}$	$\{\bar{a}\bar{b}\}$
$\bar{a}\bar{b}d + a\bar{d}$	$\{c\}$
$\bar{a}\bar{b}c + a\bar{c}$	$\{d\}$
$\bar{a}\bar{b}d + b\bar{d}$	$\{c\}$
$\bar{a}\bar{b}c + b\bar{c}$	$\{d\}$
$\bar{a}\bar{b}\bar{c} + ac$	$\{d\}$
$\bar{a}\bar{b}\bar{d} + ad$	$\{c\}$
$\bar{a}\bar{b}\bar{c} + bc$	$\{d\}$
$\bar{a}\bar{b}\bar{d} + bd$	$\{c\}$
$c\bar{d} + \bar{c}d$	$\{a\}, \{b\}$
$a + b$	$\{c\bar{d}\}, \{\bar{c}d\}$
$ac\bar{d} + b\bar{c}d$	$\{\}$
$a\bar{c}d + bc\bar{d}$	$\{\}$

Next, we compute the double-cube divisors of y

Double-cube divisor	Base
$c + d$	$\{\bar{a}\bar{b}\}$
$\bar{a}\bar{b}c + a\bar{c}\bar{d}$	$\{\}$
$\bar{a}\bar{b}c + b\bar{c}\bar{d}$	$\{\}$
$\bar{a}\bar{b}d + a\bar{c}\bar{d}$	$\{\}$
$\bar{a}\bar{b}d + b\bar{c}\bar{d}$	$\{\}$
$a + b$	$\{\bar{c}\bar{d}\}$

Next, we will combine the double-cube divisors and compute their weight. We will not consider those with an empty base as their weight will be -1.

We will use the weight function

$$w(d) = (p-1)(x+y) - p + \sum_{i=1}^{p} |b_i| + c$$

$$= p(x+y) - p - (x+y) + \sum_{i=1}^{p} |b_i| + c$$

Double-cube Divisor	Base	weight
$cd + \bar{c}\bar{d}$	$\{\bar{a}\bar{b}\}$	$(3-1)(4) - 3 + 2 + 1 + 1 = 9$
$a\bar{b}d + a\bar{d}$	$\{c\}$	$(1-1)(5) - 1 + 1 = 0$
$a\bar{b}c + a\bar{c}$	$\{d\}$	$(1-1)(5) - 1 + 1 = 0$
$\bar{a}\bar{b}d + b\bar{d}$	$\{c\}$	$(1-1)(5) - 1 + 1 = 0$
$\bar{a}\bar{b}c + b\bar{c}$	$\{d\}$	$(1-1)(5) - 1 + 1 = 0$
$\bar{a}\bar{b}\bar{c} + ac$	$\{\bar{d}\}$	$(1-1)(5) - 1 + 1 = 0$
$\bar{a}\bar{b}\bar{d} + ad$	$\{\bar{c}\}$	$(1-1)(5) - 1 + 1 = 0$
$\bar{a}\bar{b}\bar{c} + bc$	$\{\bar{c}\}$	$(1-1)(5) - 1 + 1 = 0$
$c\bar{d} + \bar{c}d$	$\{a, b\}$	$(3-1)(4) - 3 + 2 + 1 + 1 = 9$
$a + b$	$\{\bar{c}\bar{d}\}, \{c\bar{d}\}, \{\bar{c}d\}$	$(3-1)(2) - 3 + 2 + 2 + 2 + 4 = 11$
$c + d$	$\{\bar{a}\bar{b}\}$	$(1-1)(2) - 1 + 2 + 3 = 4$

So, the best double-cube divisor is $a+b$ and $W_{d\,max} = 11$

The highest weight single-cube divisor with two-literals is $\bar{a}\bar{b}$ which has a weight $= (4-2) = 2$

So, the double-cube divisor $a+b$ is best to extract.

$[1] = a + b$

$x = [1]\,c\bar{d} + [1]\,\bar{c}d + [1]'\,cd + [1]'\,\bar{c}\bar{d}$

$y = [1]'\,c + [1]'\,d + [1]\,\bar{c}\bar{d}$

$\overset{21}{=}$ literals

Note that the number of literals saved $= 32 - 21 = 11$ literals which is the computed weight of $a+b$.

We repeat the process for the resulting network.

Double-cube Divisor	Base	weight
$c\bar{d} + \bar{c}d$	$\{[1]\}$	$(2-1)(4)-2+1+1 = 4$
$[1]\bar{d} + [1]'d$	$\{c\}$	$(2-1)(4)-2+1+1 = 4$
$[1]\,c + [1]'\,\bar{c}$	$\{\bar{d}\}$	$(2-1)(4)-2+1+1 = 4$
$[1]\,\bar{c} + [1]'c$	$\{d\}$	$(2-1)(4)-2+1+1 = 4$
$[1]\,d + [1]'\,d'$	$\{\bar{c}\}$	$(2-1)(4)-2+1+1 = 4$
$cd + \bar{c}\bar{d}$	$\{[1]'\}$	$(2-1)(4)-2+1+1 = 4$
$c+d$	$\{[1]'\}$	$(1-1)(2)-1+1+2 = 2$

There are several candidate double cube divisors with the same weight $W_{dmax} = 4$

Note that $W_{smax} = (2-2) = 0$ (e.g. $\bar{c}\bar{d}$).

Let us use the divisor $c\bar{d} + \bar{c}d$.

This results in the network:

$[1] = a + b$

$[2] = c\bar{d} + \bar{c}d$

$x = [1]\,[2] + [1]'\,[2]'$

$y = [1]'c + [1]'d + [1]\,\bar{c}\bar{d}$ 17 literals

In a similar manner, the double-cube divisor $c+d$ will then be extracted which has a weight $= (1-1)(2) - 1 + 1 + 1 = 1$

The resulting network will be:

$[1] = a + b$
$[2] = c\bar{d} + \bar{c}d$
$[3] = c + d$
$x = [1][2] + [1]'[2]'$
$y = [1]'c + [1]'d + [1]\bar{c}\bar{d}$

$\underline{\underline{16}}$ literals

Running the SIS command fx produces identical result.

(iii) Running the command gkx followed by gcx produces the following network:

$[2] = a + b$
$[3] = \bar{a}\bar{b}$
$x = [2]c\bar{d} + [2]\bar{c}d + [3]cd + [3]\bar{c}\bar{d}$
$y = [2]\bar{c}\bar{d} + [3]c + [3]d$

$\underline{\underline{23}}$ literals

Note that the result produced by fx is much better than that produced by Kernel extraction followed by cube extraction.

Q5

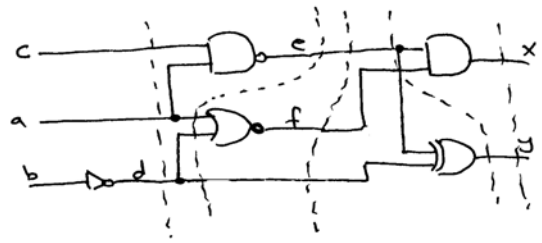
$d = \bar{b}$

$f = \overline{(a+d)}$

$e = \overline{(ca)}$

$x = fe$

$y = d \oplus e$



(i) $CDC_{in} = ab\bar{c}$

We first select vertex $\underline{d}$

$CDC_{out} = ab\bar{c} + (d \oplus \bar{b}) = ab\bar{c} + bd + \bar{b}\bar{d}$

We then drop varrable $\underline{b}$ by consensus:

$(CDC_{out})_{\bar{b}} \cdot (CDC_{out})_{b} = \bar{d} \cdot (a\bar{c} + d) = a\bar{c}\bar{d}$

Next, we select vertex $\underline{e}$

$CDC_{out} = a\bar{c}\bar{d} + (e \oplus \overline{ac})$

$= a\bar{c}\bar{d} + eac + \bar{e}(\bar{a}+\bar{c})$

We remove varrable $\underline{c}$ by consensus:

$\Rightarrow (a\bar{d} + \bar{e})(ea + \bar{e}\bar{a}) = \bar{a}\bar{e} + ae\bar{d}$

Next, we select varrable $\underline{f}$

$CDC_{out} = \bar{a}\bar{e} + ae\bar{d} + f \oplus \overline{(a+d)}$

$= \bar{a}\bar{e} + ae\bar{d} + f(a+d) + \bar{f}\bar{a}\bar{d}$

We remove varrable $\underline{a}$:

$\Rightarrow (\bar{e} + fd + \bar{f}\bar{d})(e\bar{d} + f)$

$= f\bar{e} + fd + ef\bar{d}$

Then, we select vertex $\underline{y}$

$CDC = f\bar{e} + fd + ef\bar{d} + y \oplus (e \oplus d)$

We remove varrable $\underline{d}$:

$\Rightarrow (f\bar{e} + e\bar{f}+(y \oplus e))(\underbrace{f\bar{e}+f}_{=f} + (y \oplus \bar{e}))$

$= f\bar{e} + f(y \oplus e) + e\bar{f}(y \oplus \bar{e}) + \underbrace{(y \oplus e)(y \oplus \bar{e})}_{=0}$

$= f\bar{e} + f(y\bar{e} + \bar{y}e) + e\bar{f}(ye + \bar{y}\bar{e})$

$= f\bar{e} + fy\bar{e} + e\bar{f}y = f\bar{e} + f\bar{y} + e\bar{f}y$

Finally, we add vertex $\underline{x}$:

$CDC = f\bar{e} + f\bar{y} + e\bar{f}y + x \oplus ef$

Then, we remove variable $\underline{e}$:

$\Rightarrow (f + f\bar{y} + x)(f\bar{y} + \bar{f}y + x \oplus f)$

$= (f + x)(f\bar{y} + \bar{f}y + x \oplus f)$

$= f\bar{y} + f(x \oplus f) + xf\bar{y} + x\bar{f}y + x(x \oplus f)$

$= f\bar{y} + f(x\bar{f} + \bar{x}f) + xf\bar{y} + x\bar{f}y + x(x\bar{f} + \bar{x}f)$

$= f\bar{y} + f\bar{x} + xf\bar{y} + x\bar{f}y + \bar{f}x$

$= f\bar{y} + f\bar{x} + \bar{f}x$

Finally, we remove variable $\underline{f}$:

$\Rightarrow (x) \cdot (\bar{y} + \bar{x}) = x\bar{y}$

So, the obtained $CDC = x\bar{y}$

Note that the correctness of the obtained result can be verified by noticing that $x=1$ implies $e=1$ and $f=1$. Having $f=1$ implies $a=0$ and $d=0$. However, $d=0$ and $e=1$ implies that $y=1$. So, we can't have any input assignment that sets $x=1$ and $y=0$ simultaneously. Thus, $x\bar{y}$ is a don't care condition.

(ii) $ODC_x = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, $ODC_y = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

① vertex f

$$ODC_f = ODC_x + \left(\frac{\overline{\partial x}}{\partial f}\right) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} \bar{e} \\ \bar{e} \end{pmatrix} = \begin{pmatrix} \bar{e} \\ 1 \end{pmatrix}$$

② vertex e

$$ODC_e = ODC_{ex|e=\bar{e}} \;\overline{\oplus}\; ODC_{ey}$$

$$ODC_{ex} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} \bar{f} \\ \bar{f} \end{pmatrix} = \begin{pmatrix} \bar{f} \\ 1 \end{pmatrix}$$

$$ODC_{ey} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\Rightarrow ODC_e = \begin{pmatrix} \bar{f} \\ 1 \end{pmatrix} \overline{\oplus} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} \bar{f} \\ 0 \end{pmatrix}$$

③ vertex c

$$ODC_c = \begin{pmatrix} \bar{f} \\ 0 \end{pmatrix} + \begin{pmatrix} \bar{a} \\ \bar{a} \end{pmatrix} = \begin{pmatrix} \bar{f}+\bar{a} \\ \bar{a} \end{pmatrix} = \begin{pmatrix} a+d+\bar{a} \\ \bar{a} \end{pmatrix} = \begin{pmatrix} 1 \\ \bar{a} \end{pmatrix}$$

④ vertex d

$$ODC_d = ODC_{df|d=\bar{d}} \;\overline{\oplus}\; ODC_{dy}$$

$$ODC_{df} = \begin{pmatrix} \bar{e} \\ 1 \end{pmatrix} + \begin{pmatrix} a \\ a \end{pmatrix} = \begin{pmatrix} \bar{e}+a \\ 1 \end{pmatrix} = \begin{pmatrix} ac+a \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ 1 \end{pmatrix}$$

$$ODC_{dy} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\Rightarrow ODC_d = \begin{pmatrix} a \\ 1 \end{pmatrix} \overline{\oplus} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} a \\ 0 \end{pmatrix}$$

⑤ Vertex $\underline{b}$:

$$ODC_b = \begin{pmatrix} a \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ a \end{pmatrix} = \begin{pmatrix} a \\ 0 \end{pmatrix}$$

⑥ Vertex $\underline{a}$:

$$ODC_a = ODC_{ae}\,|_{a=\bar{a}} \;\overline{\oplus}\; ODC_{af}$$

$$ODC_{ae} = \begin{pmatrix} \bar{f} \\ 0 \end{pmatrix} + \begin{pmatrix} \bar{c} \\ \bar{c} \end{pmatrix} = \begin{pmatrix} \bar{f} + \bar{c} \\ \bar{c} \end{pmatrix} = \begin{pmatrix} a + d + \bar{c} \\ \bar{c} \end{pmatrix}$$

$$ODC_{af} = \begin{pmatrix} \bar{e} \\ 1 \end{pmatrix} + \begin{pmatrix} d \\ d \end{pmatrix} = \begin{pmatrix} d + \bar{e} \\ 1 \end{pmatrix} = \begin{pmatrix} ac + d \\ 1 \end{pmatrix}$$

$$\Rightarrow ODC_a = \begin{pmatrix} \bar{a} + d + \bar{c} \\ \bar{c} \end{pmatrix} \overline{\oplus} \begin{pmatrix} ac + d \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} (\bar{a} + d + \bar{c})(ac + d) + a\bar{d}c & (d + \bar{a})(\bar{c} + d + \bar{e}) \\ \bar{c} \end{pmatrix}$$

$$= \begin{pmatrix} \bar{a}d + adc + d + d\bar{c} \\ \bar{c} \end{pmatrix}$$

$$= \begin{pmatrix} d \\ \bar{c} \end{pmatrix} = \begin{pmatrix} \bar{c} \\ q \end{pmatrix}$$

Q6

$$\mu = a\bar{b} + bc$$

$$x = a\mu + b$$

$$y = \bar{a}\bar{\mu} + \bar{c}$$

(i) SDC for node μ:

$$\mu \oplus (a\bar{b} + bc) = \bar{\mu}a\bar{b} + \bar{\mu}bc + \mu\bar{a}\bar{b} + \mu b\bar{c}$$

Next, we compute ODC_μ.

Note that μ is not observable at x when $\bar{a}+b \Rightarrow ODC_{\mu,x} = \bar{a}+b$

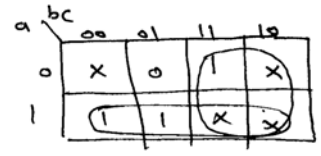
$$ODC_{\mu,y} = a + \bar{c}$$

$$\Rightarrow ODC_\mu = ODC_{\mu,x} \cap ODC_{\mu,y}$$

$$= (\bar{a}+b)(a+\bar{c}) = \bar{a}\bar{c} + ba + b\bar{c}$$

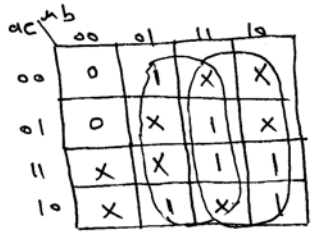
$$= \bar{a}\bar{c} + ba$$

(ii) Simplify μ using ODC_μ.



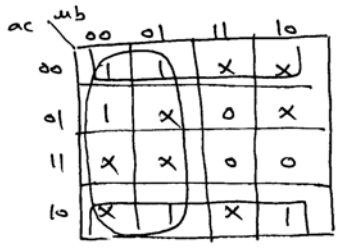
$$\Rightarrow \mu = a + b$$

(iii) Simplify x using SDC:



$$\Rightarrow x = \mu + b$$

Simplify y using SDC:



$y = \bar{\mu} + \bar{c}$

(iv) Applying the SIS command full_simplify produces the result:

$\mu = a + b$

$x = b + \mu$

$y = \bar{c} + \bar{\mu}$

The result obtained is identical to what is obtained in (ii) and (iii).

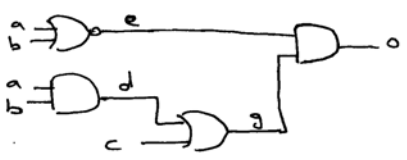
Q7

$o = eg$

$e = \overline{(a+b)}$

$g = d + c$

$d = ab$



(i) Is the perturbation replacing d by o feasible or not?

We first compute ODC$_d$.

$ODC_d = c + \bar{e} = c + a + b$

Since $\delta = ab \oplus o = ab \leq ODC_d$

$\Rightarrow$ Perturbation is feasible.

(ii) Is the fault d stuck-at-0 testable?

The fault d stuck-at-0 is untestable if the perturbation $\delta = ab \oplus 0 = ab$ is feasible. Since we found from part (i) that the perturbation is feasible, this implies that the fault d stuck-at-0 is untestable. The network can be optimized by replacing d by 0 and we obtain:

$g = c$

$e = \overline{(a+b)}$

$0 = eg$

(iii) Is the fault e stuck-at-1 testable?

The fault is untestable if the perturbation $\delta = \overline{(a+b)} \oplus 1 = a+b$ is feasible.

The perturbation $\delta = a+b$ is feasible if $\delta \leq ODC_e$.

$ODC_e = \bar{g} = \overline{(c+d)} = \overline{(c+ab)}$

$= \bar{c} \cdot (\bar{a}+\bar{b}) = \bar{c}\bar{a} + \bar{c}\bar{b}$

Since $\delta = a+b \not\leq \bar{c}\bar{a} + \bar{c}\bar{b}$, this implies that the perturbation is not feasible and the fault e stuck-at-1 is testable.

The set of all tests that detect this fault is $\bar{e} \cdot \overline{ODC_e} = (a+b)(c+ab)$

$= ac + bc + ab$

So, there are four possible test vectors that detect the fault e stuck-at-1 $\{101, 111, 011, 110\}$.