Boolean Function Decomposition

Most basic decomposition = Shannon's Expansion

$f(x, y, z, \ldots)$ = Boolean function

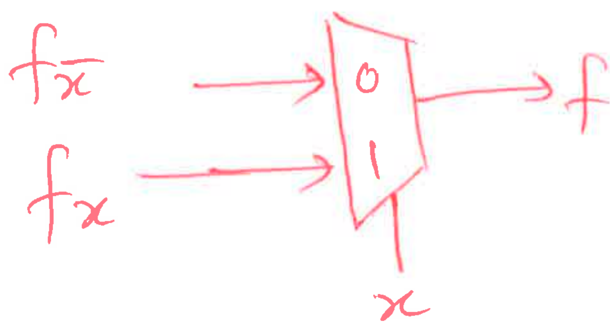
$$\boxed{f = x f_x + \bar{x} f_{\bar{x}}}$$

$f_x = f(x=1, y, z, \ldots)$ = +ve cofactor of f w.r.t. x

$f_{\bar{x}} = f(x=0, y, z, \ldots)$ = −ve " " " x

Decomposes f w.r.t. variable x

Implementation:



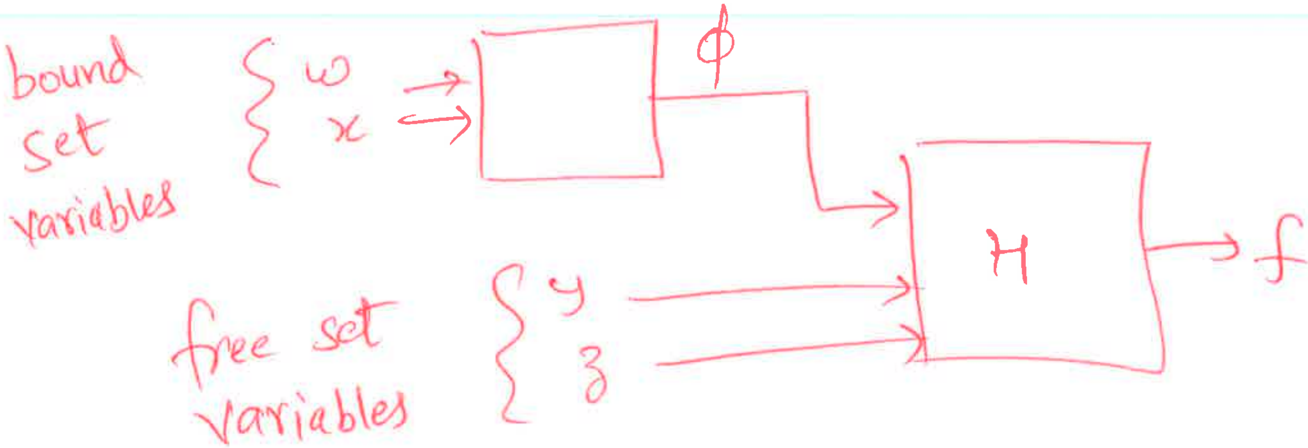
Example: Let $f = ab + ac + bc$

$$\left.\begin{array}{l} f_a = b + c \\ f_{\bar{a}} = bc \end{array}\right\} \begin{aligned} f &= a(b+c) + \bar{a} \cdot bc \\ &= ab + ac + \bar{a}bc \\ &= ab + c[a + \bar{a}b] \\ &= ab + c[a+b] = ab + ac + bc \end{aligned}$$

Can we generalize this decomposition w.r.t. a subfunction ϕ ?

$$f(w,x,y,z) = H(\phi, y, z)$$

where $\phi(w,x)$ = a computable function



Then $f = \phi f_{\phi} + \bar{\phi} f_{\bar{\phi}}$

f_{ϕ} & $f_{\bar{\phi}}$ = generalized +ve & -ve cofactors of f w.r.t ϕ

[Such a decomposition is also called Simple Disjunctive Decomposition]

Such a decomposition <u>may</u> or <u>may not</u> exist.

<u>Q1</u>. Given f, bound-set & free set variables, find if f admits a simple disjunctive decomposition:

i.e. $f = H(\phi(\text{Bound set}), \text{free set})$?

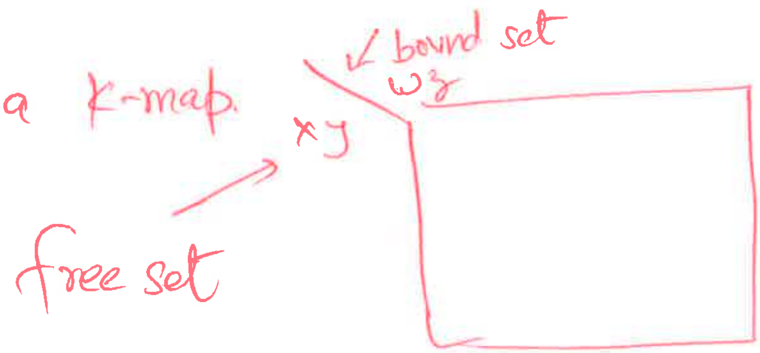
If it does, then find (compute) a corresponding ϕ & H

<u>Example:</u> $f(w,x,y,z) = \Sigma(0,2,3,7,9,10,11,14)$

Given bound set $= \{w, z\}$

Free set $= \{x, y\}$

→ Construct a K-map.



xy \ wz	00	01	11	10
00	1	0	1	0
01	1	1	1	1
11	0	1	0	1
10	0	0	0	0
	A	B	A	B

If the number of distinct column patterns (column multiplicity) is no greater than 2, then a simple disjunctive decomposition exists. Otherwise not.

$$A = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix},$$ column multiplicity $= 2$

So a decomposition exists.

Find ϕ, $\underbrace{f_{\phi} \quad f_{\bar{\phi}}}_{H}$

Let ϕ correspond to column patterns A
then $\overline{\phi}$ corresponds to column patterns B

$$\phi(w,z) = \{wz=00,\ wz=11\}$$
$$= \overline{w}\,\overline{z} + wz$$

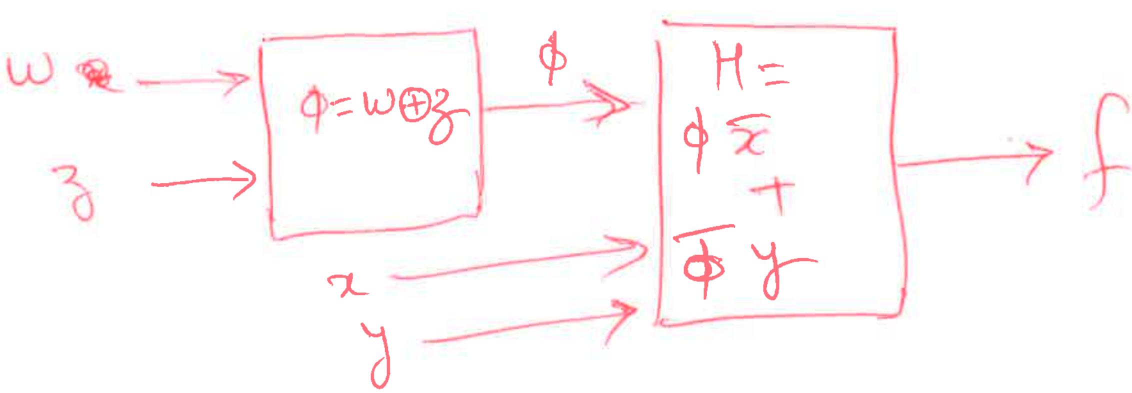
$$\overline{\phi} = \{wz=01,\ wz=10\} = \overline{w}z + w\overline{z}$$

Create a new K-map with columns A, likewise columns B, merged

This will give us H

xy	ϕ	$\overline{\phi}$
00	1	0
01	1	1
11	0	1
10	0	0

$$H = \underbrace{\phi \cdot \overline{x}}_{f_\phi} + \underbrace{\overline{\phi} \cdot y}_{f_{\overline{\phi}}}$$



Example 2

bound set: ab

free set: cd

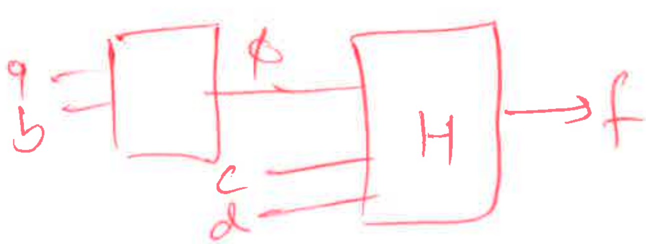
cd \ ab	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	0	1	0	1
10	0	0	0	0
	A	B	A	B

$$\phi(a,b) = \text{Columns } A$$
$$= \bar{a}\bar{b} + ab$$

$$\bar{\phi} = \text{Columns } B = \bar{a}b + a\bar{b}$$

cd	ϕ	$\bar{\phi}$
00	1	1
01	1	0
11	0	1
10	0	0

$$H = \phi \cdot \bar{c} + \bar{\phi}\,(\bar{c}\bar{d} + cd)$$



One-bit ϕ can distinguish two different column patterns = simple decomposition....

When simple decomposition does not exist, i.e. ~~there~~ column multiplicity > 2, then we need more bits to distinguish distinct column patterns.

3 or 4 column patterns, need 2 bits.

5 to 8 column patterns, need 3-bits.....

See Q7 on HW2

(Fig2 on Q7 HW2).

Bound set = a, b, c.

Free set = de

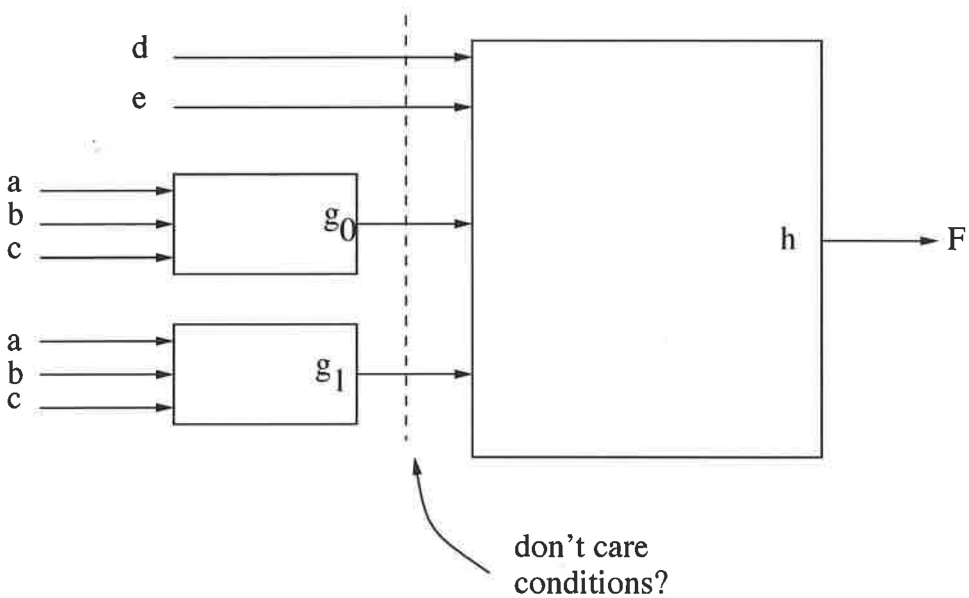
3 distinct column patterns (8)

A B C. = column multiplicity = 3

So we need two bits to distinguish them

abc	000	001	011	010	110	111	101	100
de 00	1	0	1	1	1	0	1	1
de 01	0	1	1	0	1	1	1	0
de 11	0	1	0	0	0	1	0	0
de 10	1	0	0	1	0	0	0	1
	A	C	B	A	B	C	B	A

(i) F(a, b, c, d, e)



(ii) A decomposed implementation of F(a,b,c,d,e)

Fig. 2. Decomposition of $F(a,b,c,d,e) = h(g_0(a,b,c), g_1(a,b,c), d, e)$. Compute the don't cares at the input of the $h(g_0, g_1, d, e)$ block and simplify the SOP form of h.

Let two bits be g_0 ~~and~~ and g_1

~~when g_0~~ Select an encoding

$g_0 g_1 = 11$ is a don't care condition.

$g_0 g_1 = 00 \longrightarrow$ Column A

$A = \overline{g_0}\,\overline{g_1}$

$g_0 g_1 = 10 \longrightarrow$ Column B.

$B = g_0 \overline{g_1}$

& $g_0 g_1 = 01 \longrightarrow$ Column C $\quad C = \overline{g_0} g_1$

$g_0 = 1$ when $\{abc = 011, 110, 101\}$

$g_0 = \bar{a}bc + ab\bar{c} + a\bar{b}c$

$g_1 = 1$ when $\{abc = 001, 111\}$

$g_1 = \bar{a}\bar{b}c + abc$

These are bound set functions.

So we found S.O.P. forms for g_0, g_1.

Simplify using K-maps.

Compare g_0, g_1 with those given in Q7 HW2.

Now we need to find H.

According to the K-map Q7, Fig 2 (i), and our encoding (g_0, g_1)

$$f = \bar{g}_0 \bar{g}_1 \cdot A + g_0 \cdot \bar{g}_1 \cdot B + \bar{g}_0 g_1 C.$$

$$= \bar{g}_0 \bar{g}_1 \bar{e} + g_0 \bar{g}_1 \bar{d} + \bar{g}_0 g_1 e \quad \text{(9 literals)}$$

Merge the column patterns of Fig 2 (i)

de \ $g_0 g_1$	00	01	11	10
00	1	1	X	0
01	0	1	X	1
11	0	0	X	1
10	1	0	X	0
	A			B

$g_0 g_1 = 11$ = don't care

Boolean function decomposition creates Don't cares!

$$H = \bar{g}_0 \bar{g}_1 \bar{e} + g_1 \bar{d} + g_0 e$$

(7 literals)

Compare your answer w/ solution to HW2 Q7

This type of a decomposition is called <u>Ashenhurst</u> – <u>Curtis</u> <u>decomposition</u>.

Let us take the same problem and use a different encoding for g_0 & g_1 to distinguish the columns.

$g_0 g_1$

$0\ 0 \rightarrow A$

$0\ 1 \rightarrow B$

$1\ 1 \rightarrow C.$

$1\ 0 =$ don't care

$\bar{g}_0 \bar{g}_1 = \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + a\bar{b}\bar{c}$

$\bar{g}_0 g_1 = \bar{a}bc + ab\bar{c} + a\bar{b}c$

$g_0 g_1 = \bar{a}\bar{b}c + abc.$

$g_0 = 1$ when ? $\quad g_0 = \bar{a}\bar{b}c + abc$

$g_1 = 1$ when ? $\quad g_1 = \underbrace{\bar{a}bc + ab\bar{c} + a\bar{b}c}_{\text{in column B}} + \underbrace{\bar{a}\bar{b}c + abc}_{\text{column A}}$

a \ bc	00	01	11	10
0		1	1	
1		1	1	1

$g_1 = c + ab$

$g_0 =$ already in simplified form

$$H = \bar{g}_0 \bar{g}_1 \cdot A + \bar{g}_0 g_1 \cdot B + g_0 g_1 C.$$

$$= \bar{g}_0 \bar{g}_1 \bar{e} + \bar{g}_0 g_1 \bar{d} + g_0 g_1 e \quad \text{(9 literals)}$$

Now use $g_0 g_1 = 10$ as a don't care

& simplify.

de \ $g_0 g_1$	00	01	11	10
00	1	1	0	X
01	0	1	1	X
11	0	0	1	X
10	1	0	0	X
	A	B	C	

$= H$

$$H = \bar{g}_1 + \bar{g}_0 g_1 \bar{d} + g_0 e \quad \text{(6 literals)}$$