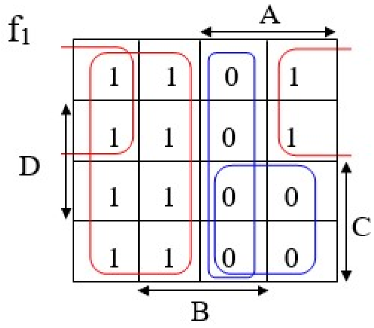
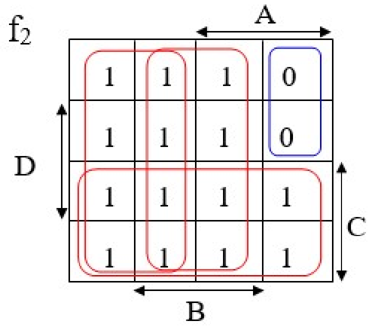
Solutions to Tutorial Sheet 8 - *Programmable Logic Devices*

1. Karnaugh Maps (individual minimisation):



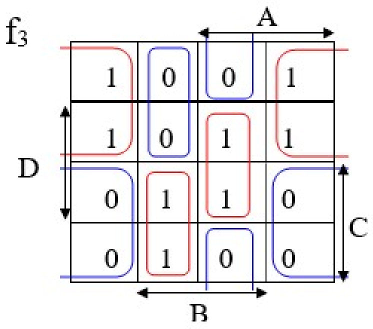


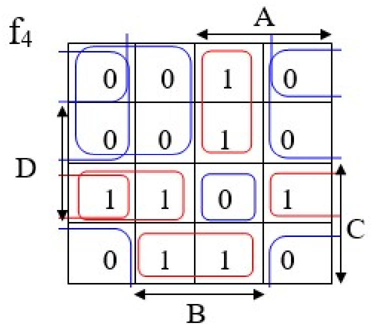
$$f_1 = \overline{A} + \overline{B}\,\overline{C}$$

$$\overline{f_1} = AB + AC$$

$$f_2 = \overline{A} + B + C$$

$$\overline{f_2} = A\overline{B}\,\overline{C}$$





$$f_3 = \overline{B}\,\overline{C} + ABD + \overline{A}BC$$

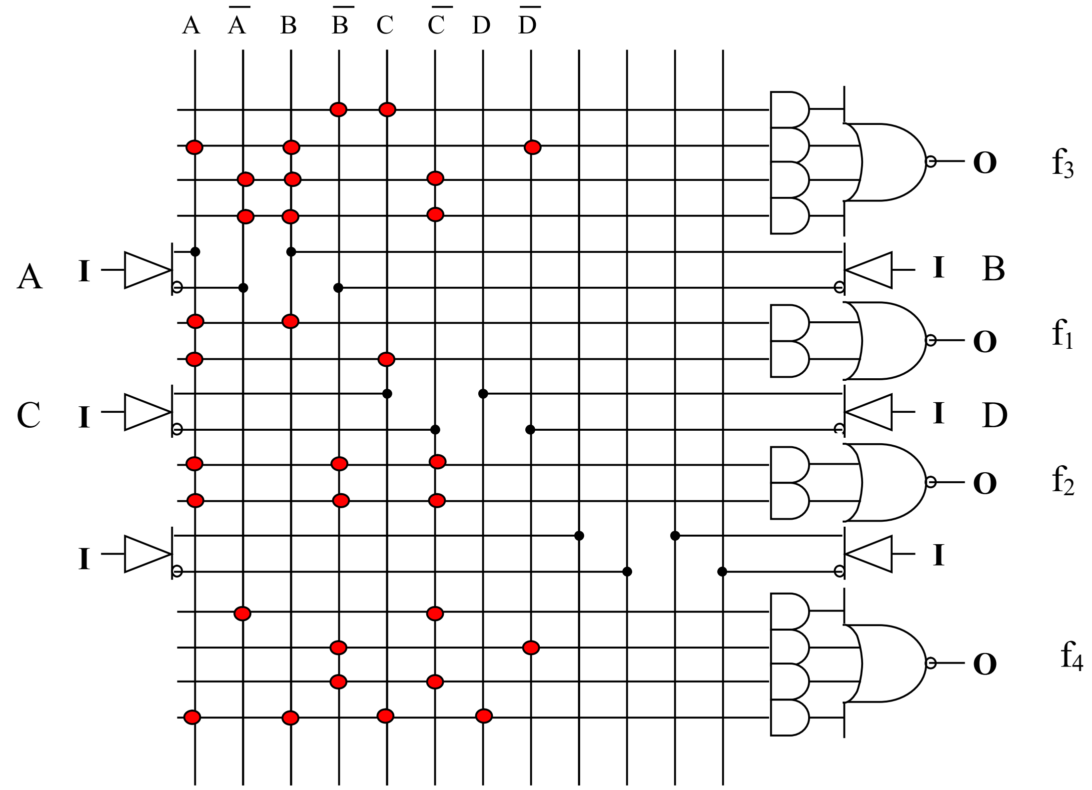
$$\overline{f_3} = \overline{B}C + AB\overline{D} + \overline{A}B\overline{C}$$

$$f_4 = AB\overline{C} + \overline{A}CD + BC\overline{D} + \overline{B}CD$$

$$\overline{f_4} = \overline{A}\,\overline{C} + \overline{B}\,\overline{D} + \overline{B}\,\overline{C} + ABCD$$

We can't implement the high functions (i.e. groupings of 1's) on the PAL6H4.

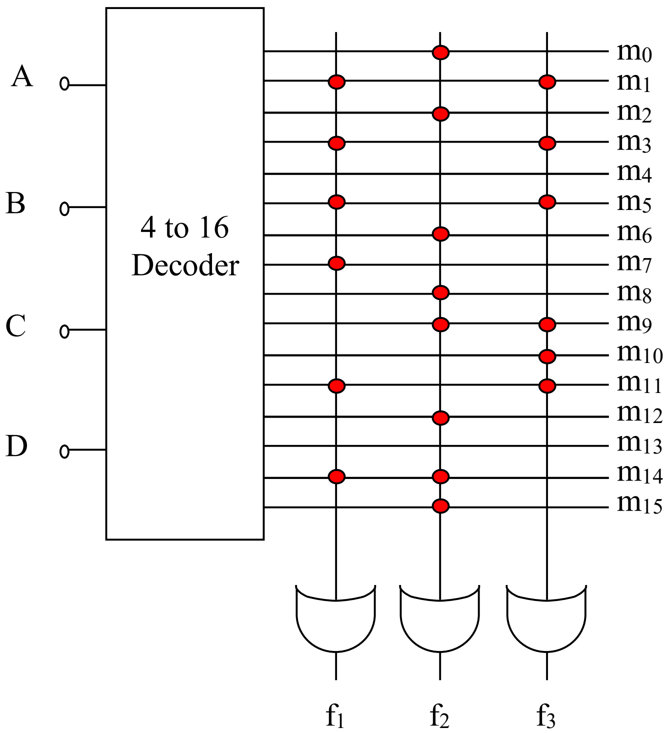
We can implement the low functions (i.e. groupings of 0's) on the PAL6L4 as follows:



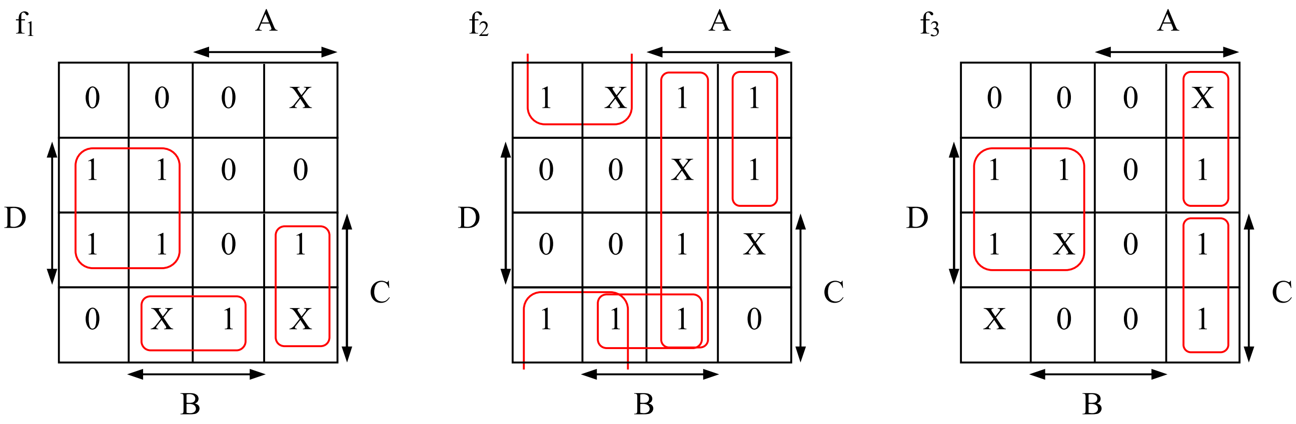
PAL6L4

2(a) The AND-plane of the ROM is a fixed structure and, for this example, is effectively a 4-to-16 decoder. Thus, the following ROM solution is obtained:

(Note, don't care terms are not included here!)



(b) We need to minimise functions using Karnaugh Maps. We need to group the 1's in the maps as we have an AND-OR structure. Also, we need to minimise maps collectively for this part.

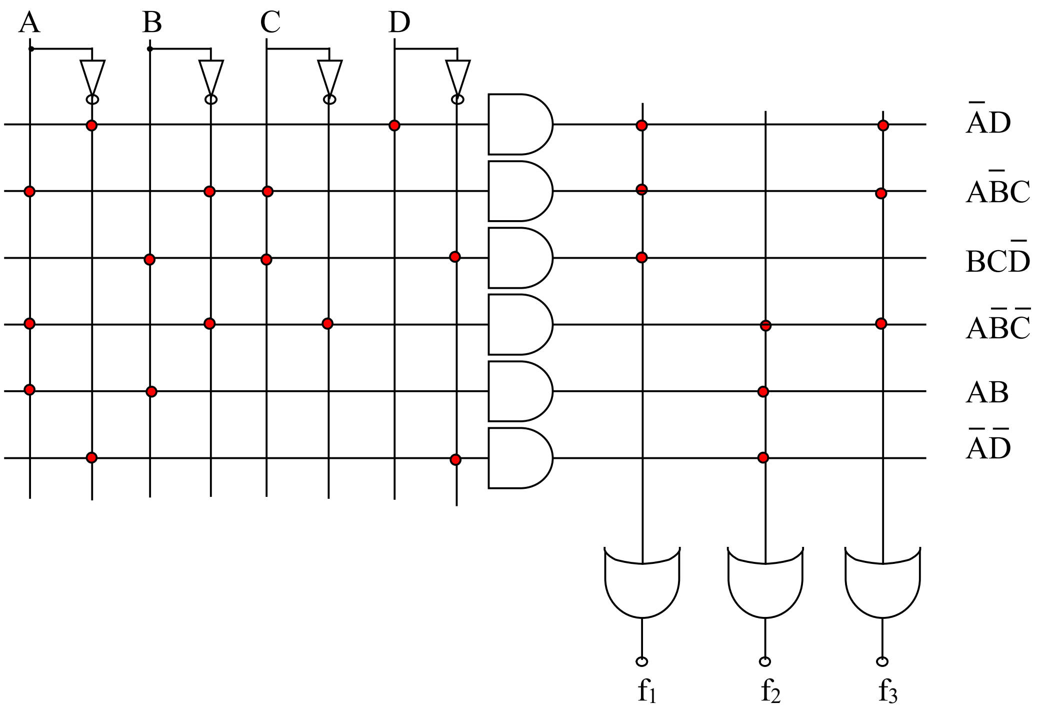


Hence, we get:

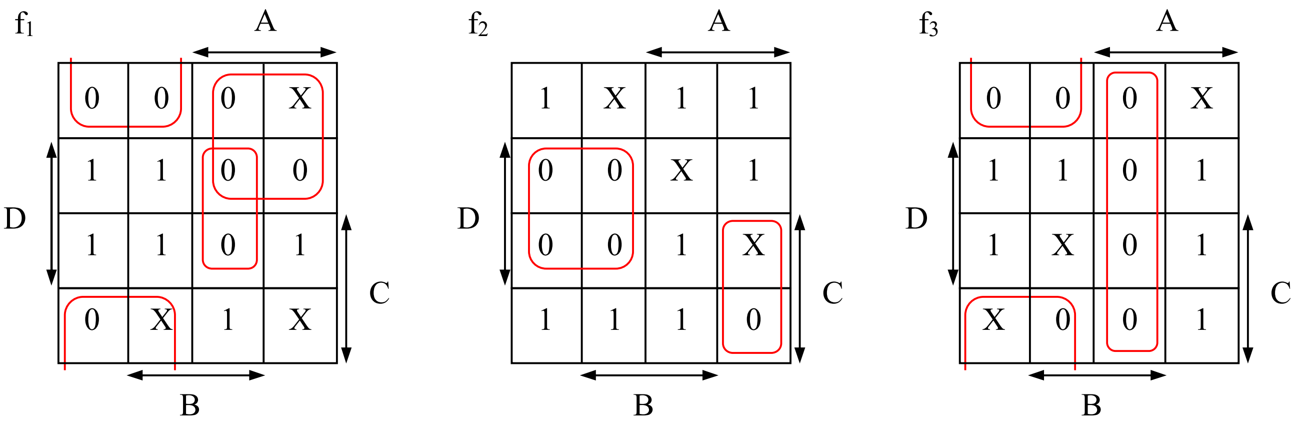
$f_1 = \bar{A}D + A\bar{B}C + BC\bar{D}$, $f_2 = A\bar{B}\bar{C} + \bar{A}\bar{D} + AB$, $f_3 = A\bar{B}C + \bar{A}D + A\bar{B}\bar{C}$

This requires 6 product terms.

Implementing previous solution using an AND-OR PLA gives:



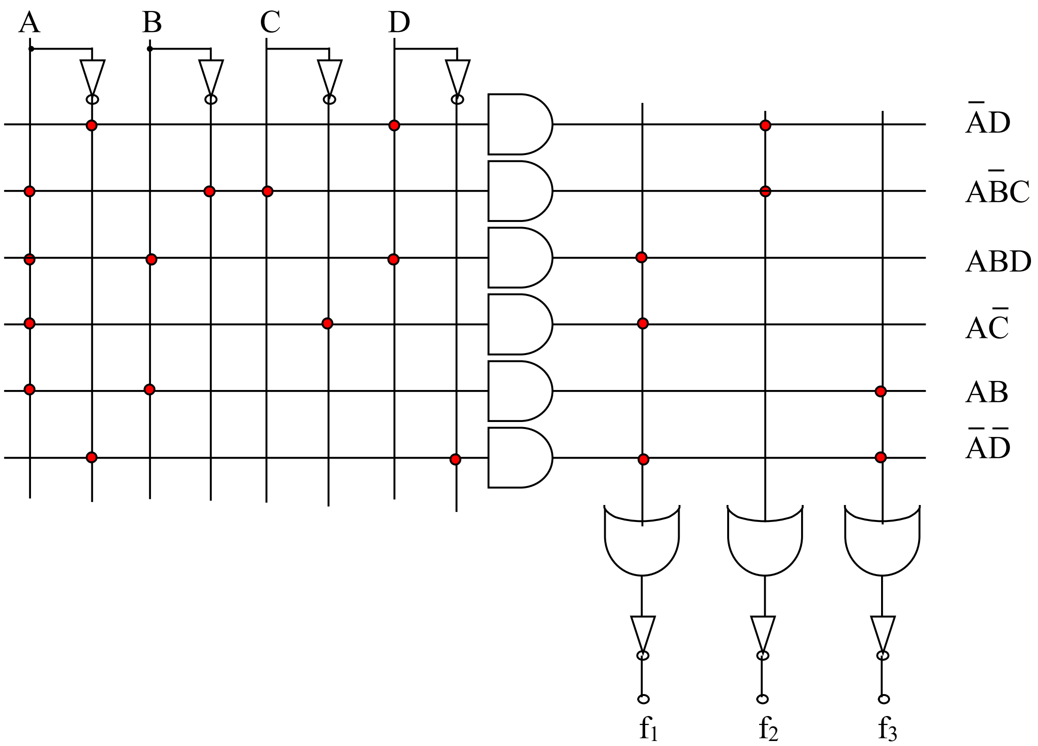
(c) Here, we need to group the 0's in the original maps to get the complement of the outputs due to the presence of the inverters at the end of the structure (i.e. we have an AND-OR-NOT structure). Again, we still need to minimise maps collectively.



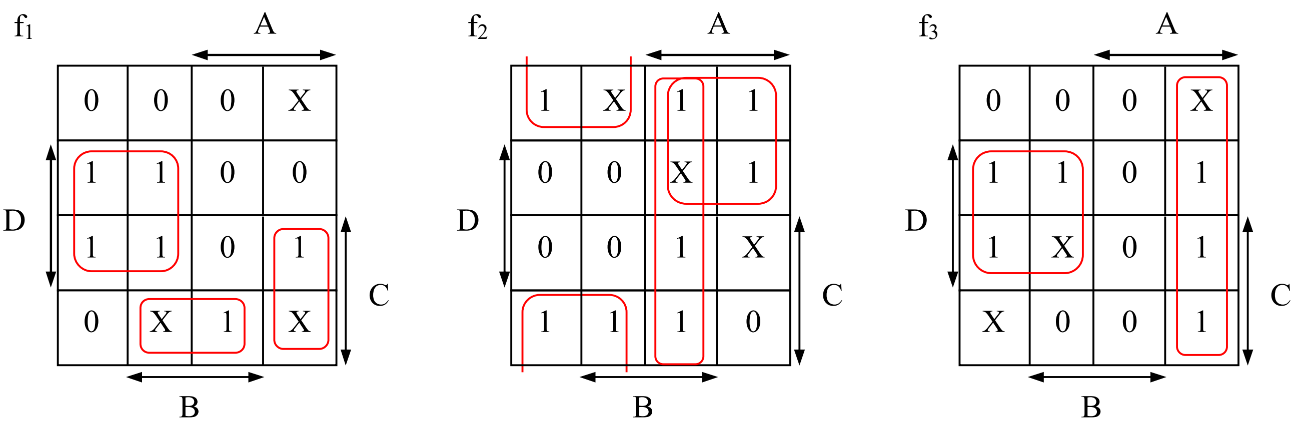
Hence, we get:

$\overline{f_1} = \overline{A}\,\overline{D} + A\overline{C} + ABD$, $\quad \overline{f_2} = \overline{A}D + A\overline{B}C$, $\quad \overline{f_3} = AB + \overline{A}\,\overline{D}$

This requires 6 product terms. Implementing using an AND-OR-INVERT PLA gives:



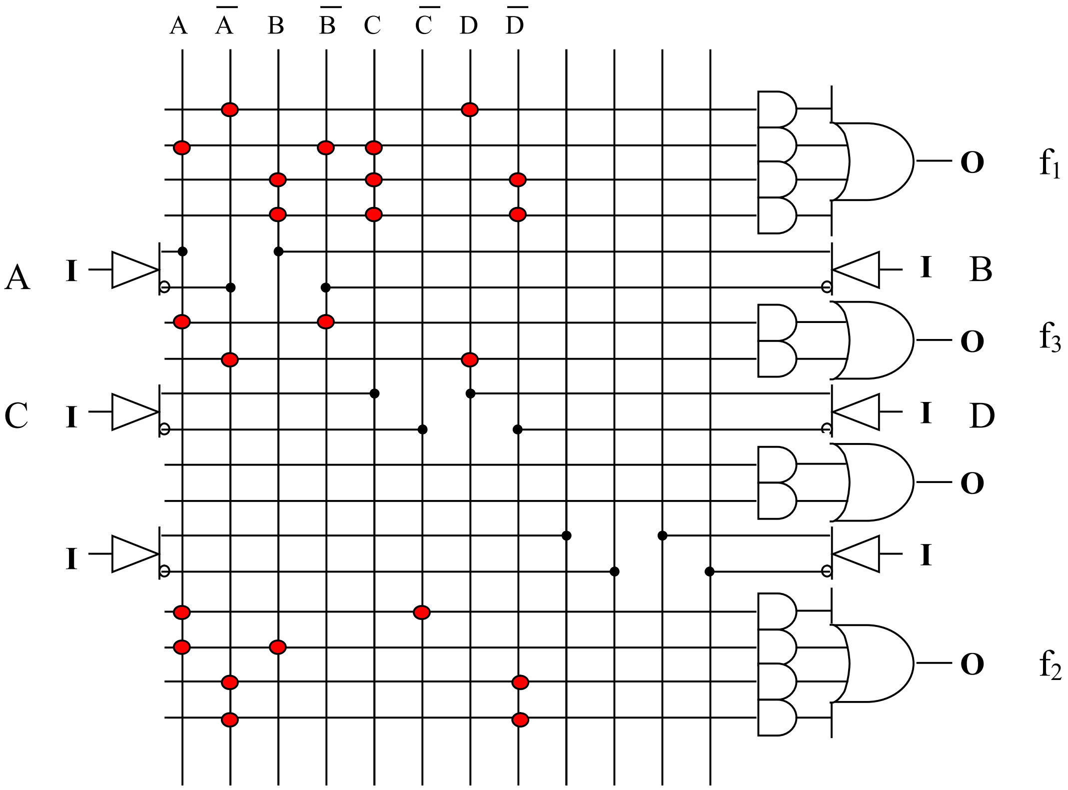
(d) Since the OR plane is fixed in the PAL structure, we don't need to minimise the Karnaugh Maps collectively. Since the PAL structure for this part is an AND-OR structure, we need to group the 1's in the individual maps. Thus, we get:



This is very similar to part (b) but now f_2 has changed slightly. Hence we get:

$f_1 = \bar{A}D + A\bar{B}C + BC\bar{D}$, $f_2 = A\bar{C} + AB + \bar{A}\bar{D}$, $f_3 = A\bar{B} + \bar{A}D$

This can be implemented on the given PAL structure as follows:



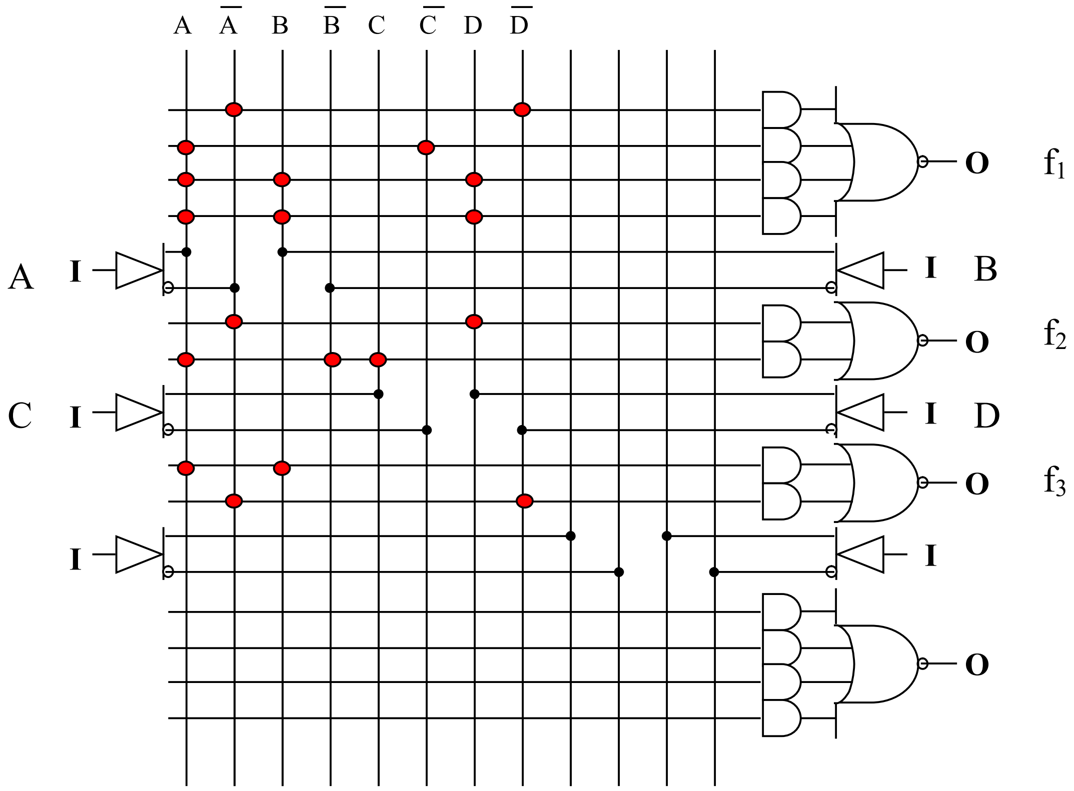
PAL6H4

(e) Since the OR plane is fixed in the PAL structure, we don't need to minimise the Karnaugh Maps collectively. Since the PAL structure for this part is an AND-NOR (or an AND-OR-NOT) structure, we need to group the 0's in the individual maps. Thus, we get:

$\overline{f_1} = \overline{A}\,\overline{D} + A\overline{C} + ABD$, $\overline{f_2} = \overline{A}D + A\overline{B}C$, $\overline{f_3} = AB + \overline{A}\,\overline{D}$

i.e. the same as part (c) in this case!

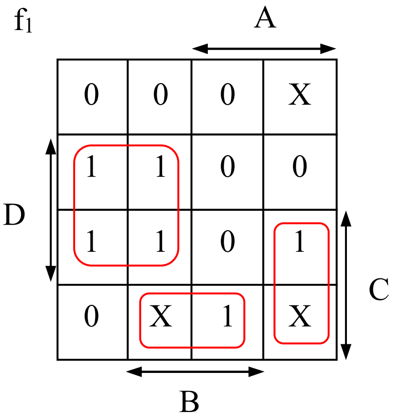
This can be implemented on the given PAL structure as follows:

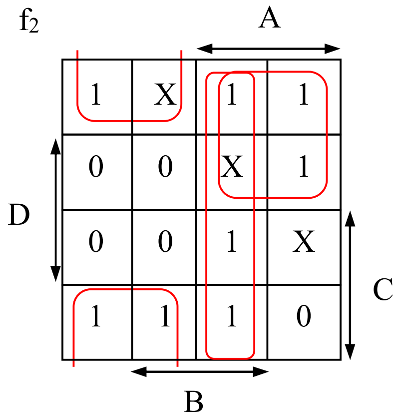


PAL6L4

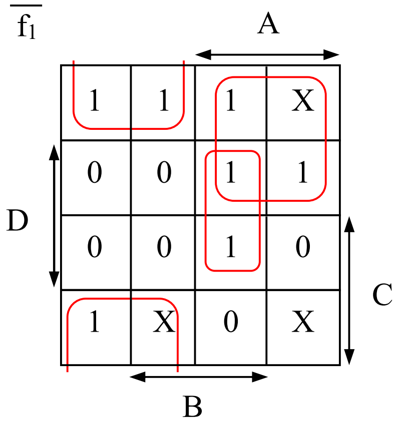
(f) The NOR-NOR-NOT structure shown in figure 2 allows the option of having a NOR-NOR or NOR-NOR-NOT implementation for each of the output functions. Thus, we need to consider the different combinations of the functions. We are now considering the first two functions only and so we need to consider 4 cases, i.e. $f_1 f_2$, $\overline{f_1} f_2$, $f_1 \overline{f_2}$ and $\overline{f_1}\,\overline{f_2}$.

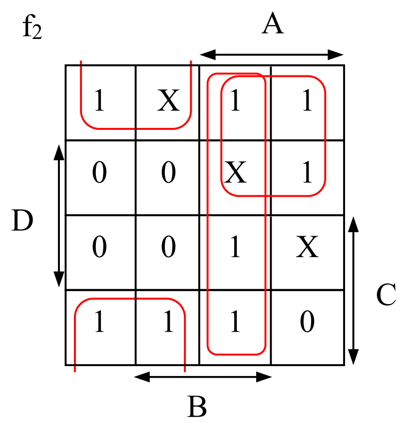
Note: We want to minimise the number of product terms collectively!



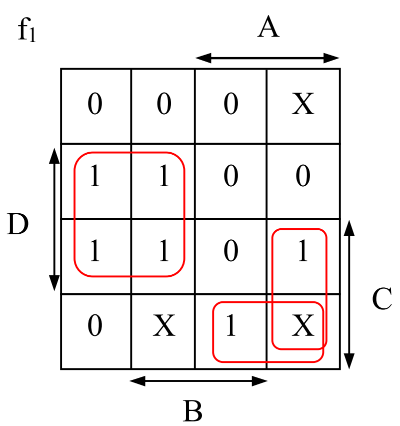


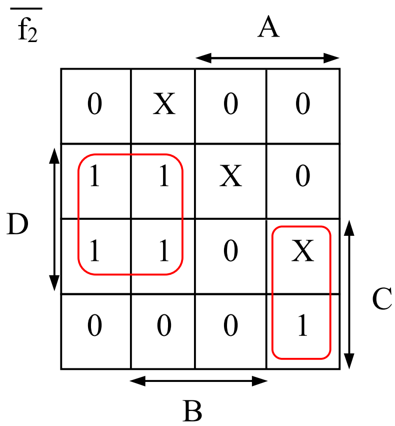
$f_1 f_2$:
6 Product terms



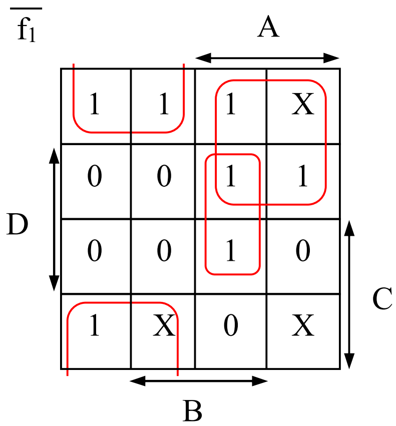


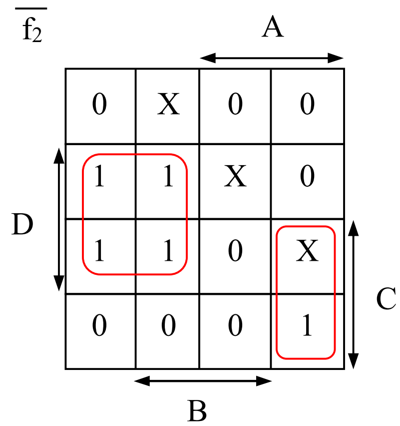
$\overline{f_1} f_2$:
4 Product terms





$f_1 \overline{f_2}$:
3 Product terms





$\overline{f_1}\,\overline{f_2}$:
5 Product terms

From the Karnaugh Maps, we can clearly see that $f_1\overline{f_2}$ offers the best possibilities.

Hence, we get:

$f_1 = \bar{A}D + AC\bar{D} + A\bar{B}C$, $\quad \overline{f_2} = \bar{A}D + A\bar{B}C$

Expressing the outputs in the appropriate format, we get:

$$f_1 = \bar{A}D + AC\bar{D} + A\bar{B}C = \overline{(A+\bar{D})} + \overline{(\bar{A}+\bar{C}+D)} + \overline{(\bar{A}+B+\bar{C})} \qquad \textit{(NOR-OR format)}$$

$$= \overline{\overline{\overline{(A+\bar{D})} + \overline{(\bar{A}+\bar{C}+D)} + \overline{(\bar{A}+B+\bar{C})}}} \qquad \textit{(NOR-NOR-NOT format)}$$

and

$$\overline{f_2} = \bar{A}D + A\bar{B}C = \overline{(A+\bar{D})} + \overline{(\bar{A}+B+\bar{C})}$$

but, we want the output f_2, therefore:

$$f_2 = \overline{\overline{(A+\bar{D})} + \overline{(\bar{A}+B+\bar{C})}} \qquad \textit{(NOR-NOR format)}$$

Finally, we have to draw out the solution. Here, we have only two output functions and need only three product lines. Hence, our implementation becomes: