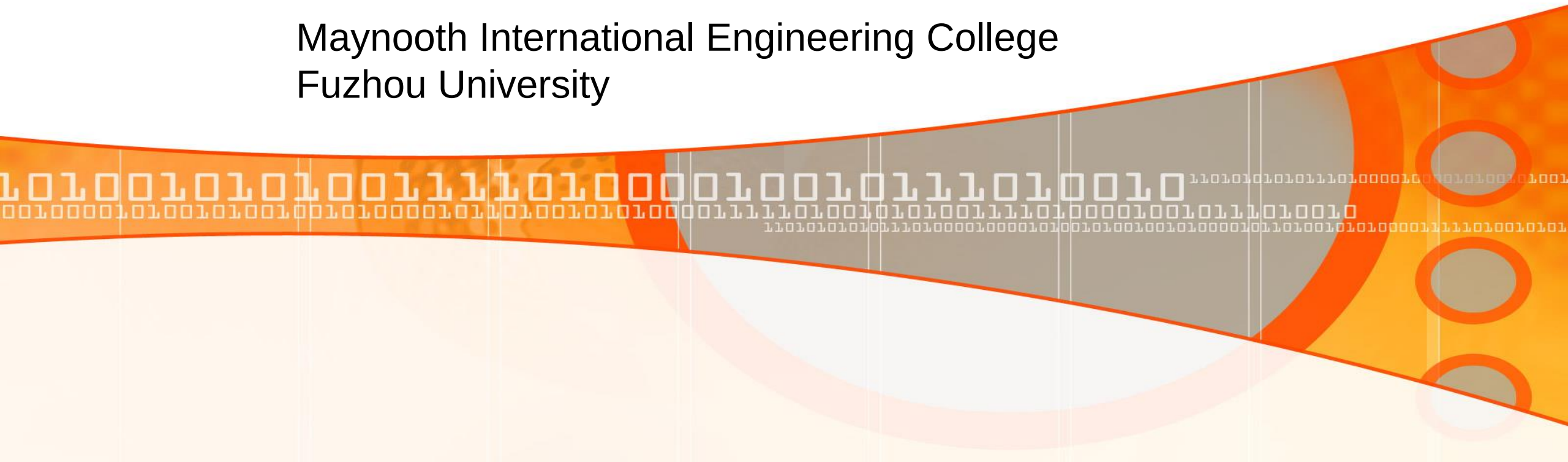
EE103 Digital Systems 1

Wong Chin Hong

106

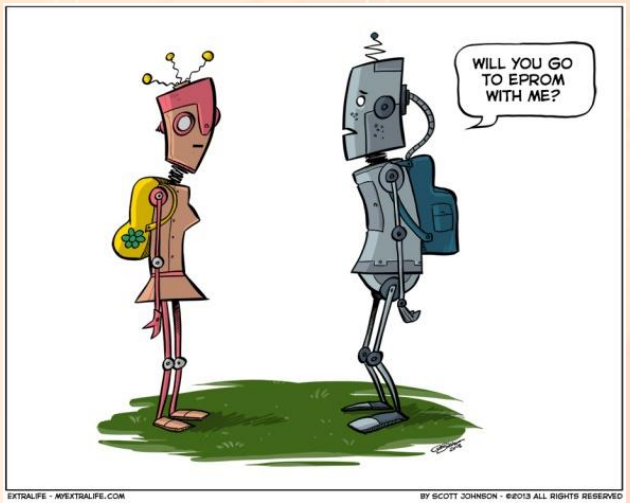
Maynooth International Engineering College
Fuzhou University



So far ... !

- We've used Karnaugh Maps to minimise logic ...
- We've implemented circuits using NAND only or NOR only gates ...
- We've analysed and designed a counter ...
- We've carried out multi-output minimisation and looked at the seven segment display …
- We've looked at Programmable Logic Devices (ROM, PAL & PLA)

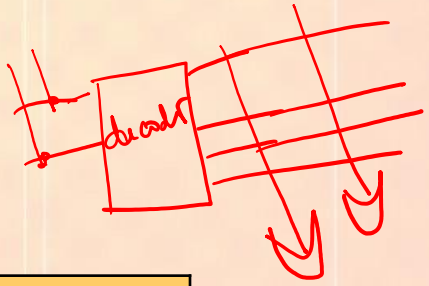
multi-output minimization



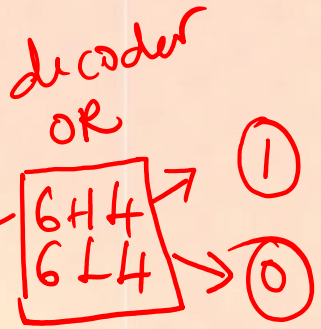
TODAY, we are going to look at the PLA device in more detail …

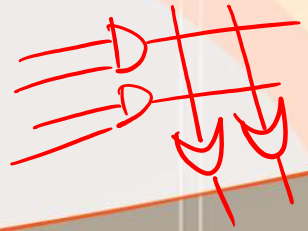
SUMMARY (ROM, PLA & PAL)

- We can summarise the **programmability** of each device as follows:



	AND plane	OR plane
ROM	Fixed	Programmable
PAL	Programmable	Fixed
PLA	Programmable	Programmable





SUMMARY (ROM, PLA & PAL)

- We can summarise the **minimisation process** for each device as follows:

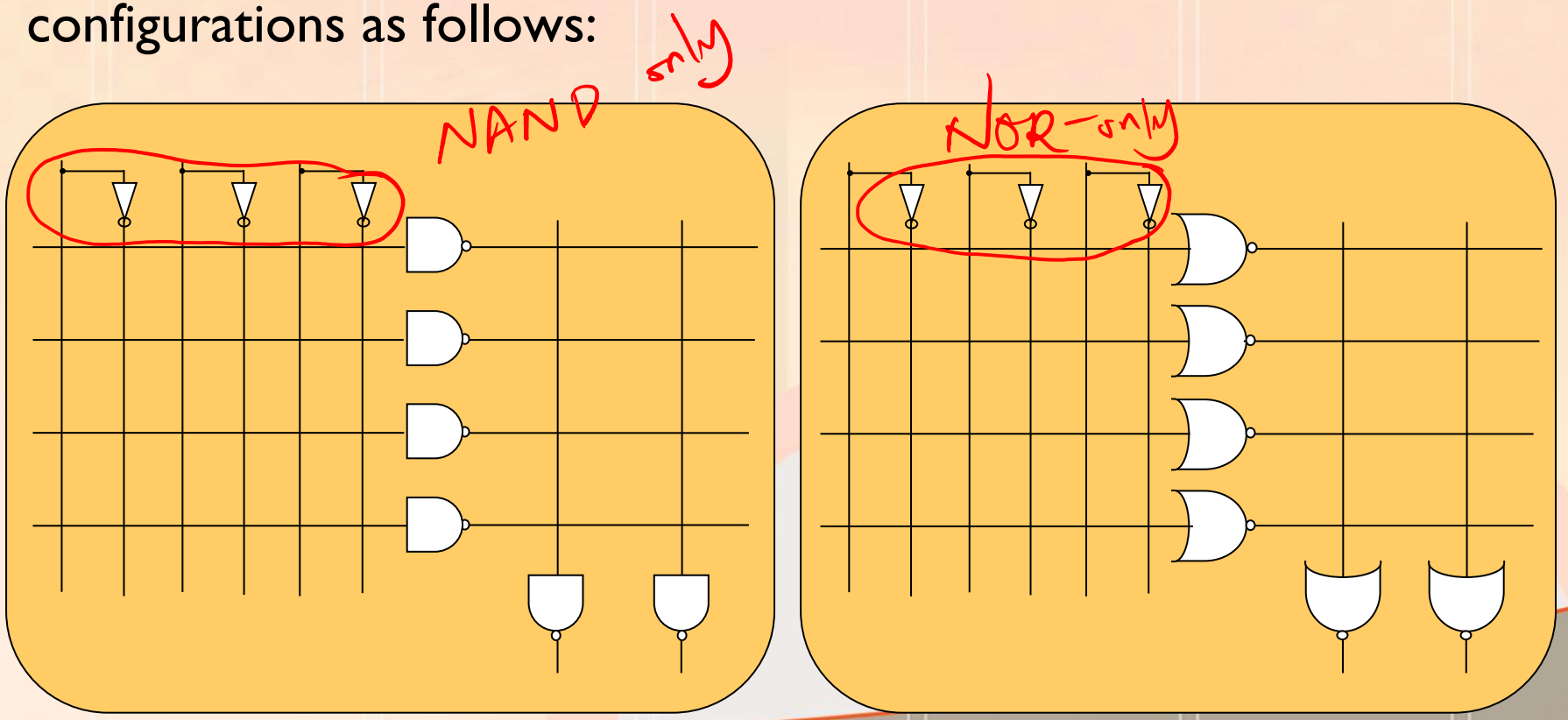
ROM	**No minimisation –** simply select the relevant minterms
PAL	**Individual minimisation**
PLA	**Multi-output minimisation**

- In terms of the PAL device, **we need to group the 1's or the 0's depending on which structure** we are given (i.e. 1's for the PAL 6H4 and 0's for the PAL 6L4).

- The PLA also offers a more flexible solution in terms of grouping either 1's or 0's. We will highlight this in more detail in the next section.

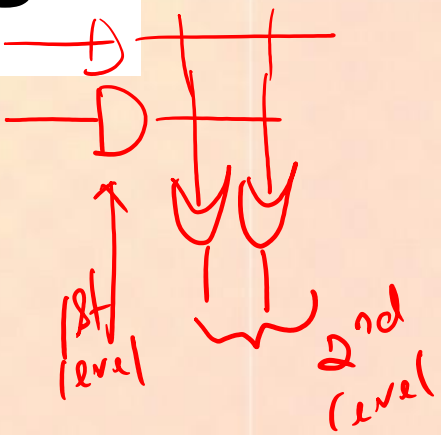
PLA – Further Implementations

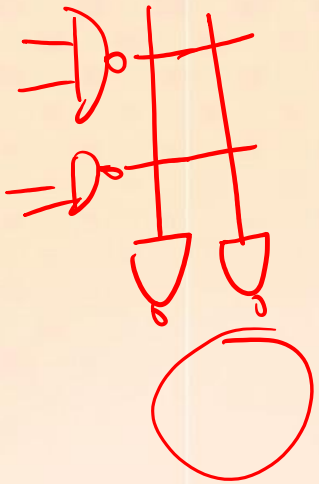
- In the previous sections we presented the PLA structure as a standard AND-OR configuration.
- We can easily present this structure in **NAND-only** and **NOR-only** configurations as follows:

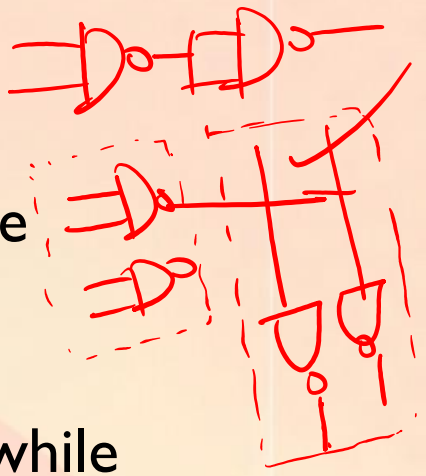


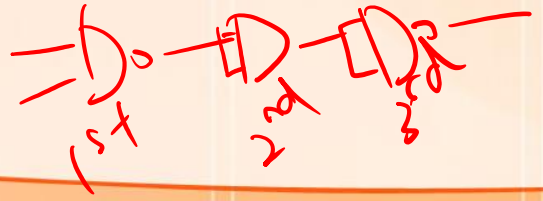
PLA – Further Implementations

- We know how to convert a sum-of-products (SOP) solution to NAND-only and NOR-only format using De Morgan's theorems.
- Hence, using the NAND-only and NOR-only PLA devices is very straightforward.
- However, in theory we could derive two different NAND-only solutions for the same function – one by grouping the 1's and the other by grouping the 0's.
- The former leads to a solution with two levels of NAND gates while the latter leads to a solution with 3 levels of NAND gates.







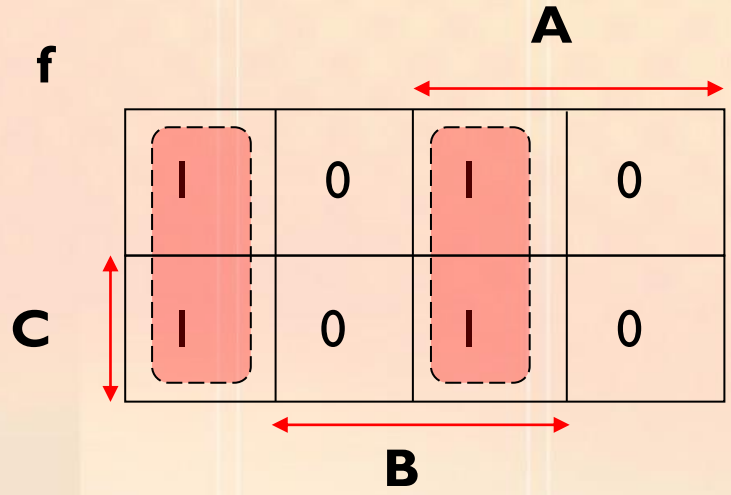


group 1

group 0

PLA – Further Implementations

- By way of illustration:



$$f = AB + \bar{A}\bar{B} = \overline{\overline{AB}\;\overline{\bar{A}\bar{B}}}$$

2 levels of NAND gates for a PLA

PLA – Further Implementations

- By way of illustration:

NAND

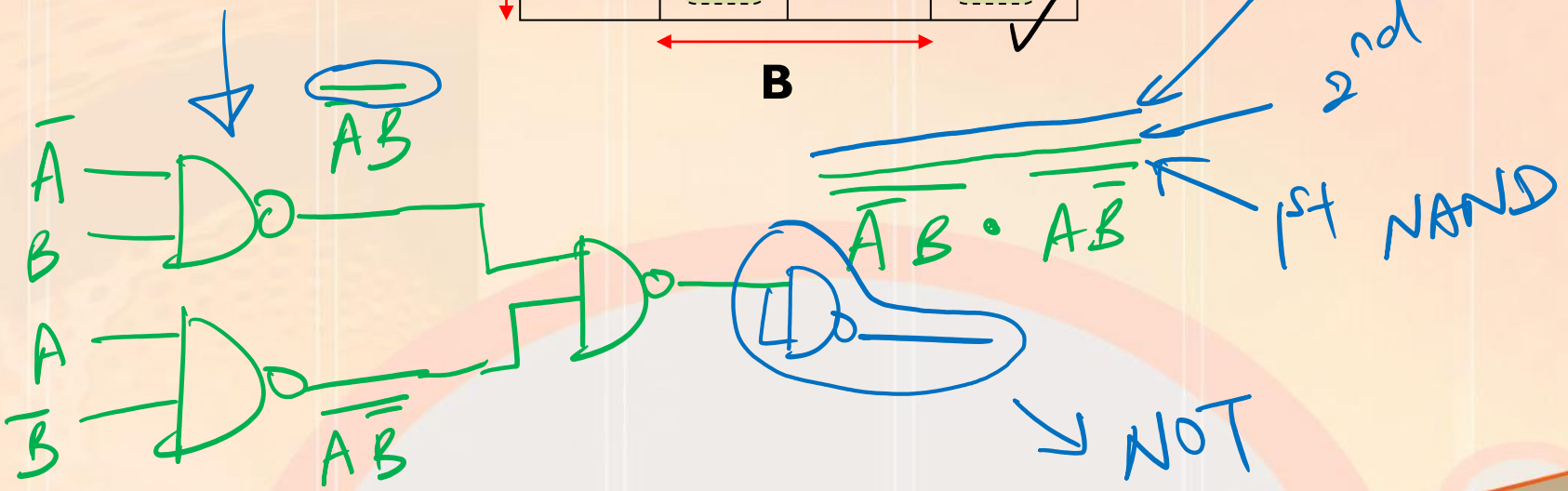
$\bar{f} = \bar{A}B + A\bar{B}$

$\bar{f} = \overline{\bar{A}B} \cdot \overline{A\bar{B}}$

$f = \overline{\bar{A}B} \cdot \overline{A\bar{B}}$

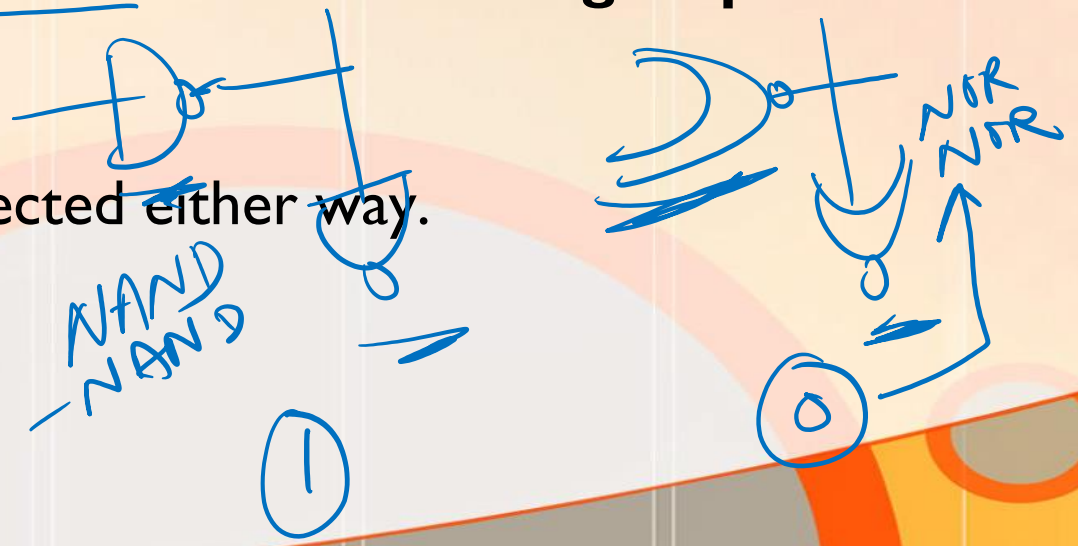
f

C \ A,B	00	01	11	10
0	1	0	1	0
1	1	0	1	0



PLA – Further Implementations

- The PLA structure given only has two levels of NAND gates and therefore the latter solution is not implementable.
- A similar argument exists for the NOR-only PLA.
- In essence, for a **NAND-NAND PLA** we need to **group the 1's** in a Karnaugh Map and for a **NOR-NOR PLA** we need to **group the 0's**.
- **Multi-output minimisation** is expected either way.



PLA – Further Implementations

- ***Ex. 9.4 Implement the following functions using a PLA with NOR-NOR architecture.***

$$f_1 = \Sigma(1, 5, 8, 10, 11, 12, 14, 15)$$

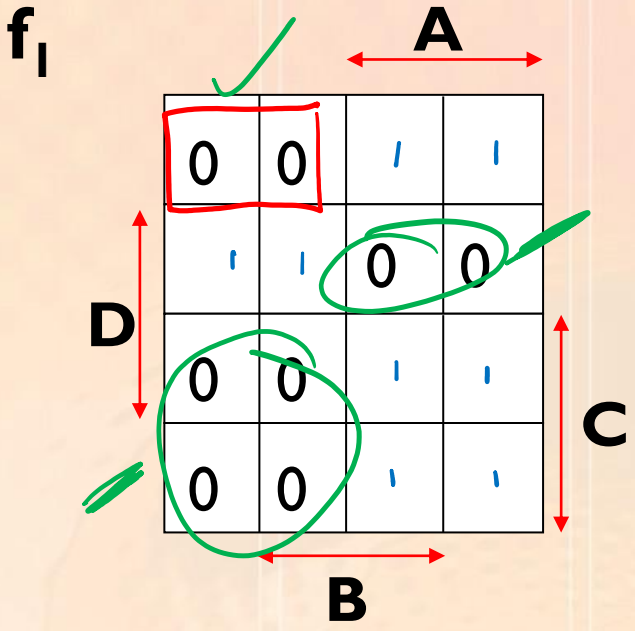
$$f_2 = \Sigma(1, 2, 3, 6, 8, 9, 10, 11, 12, 14)$$

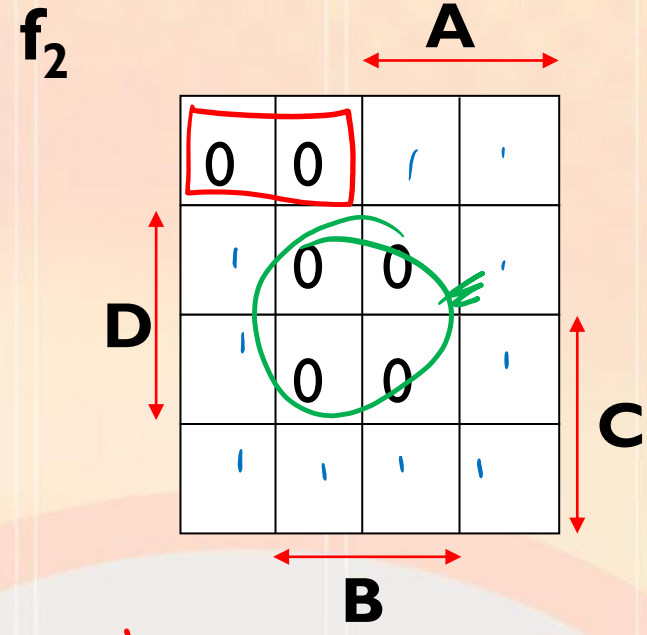
- Since we are using a PLA with NOR-NOR architecture, **we group the 0's in the Karnaugh of both functions and use multi-output minimisation.**

PLA – Further Implementations

NOR-NOR PLA → group 0

X → t





① X → t
② bar each term
③ bar everything
④ simplify

$\overline{f_1} = \bar{A}\bar{C}\bar{D} + A\bar{C}D + \bar{A}C$

$f_1 = \overline{A+C+D} + \overline{\bar{A}+C+\bar{D}} + \overline{A+\bar{C}}$

$\overline{f_2} = \bar{A}\bar{C}\bar{D} + BD$

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

2nd 1st 2nd 1st

PLA – Further Implementations

- This gives:

$$\overline{f_1} = \bar{A}\bar{C}\bar{D} + A\bar{C}D + \bar{A}C$$

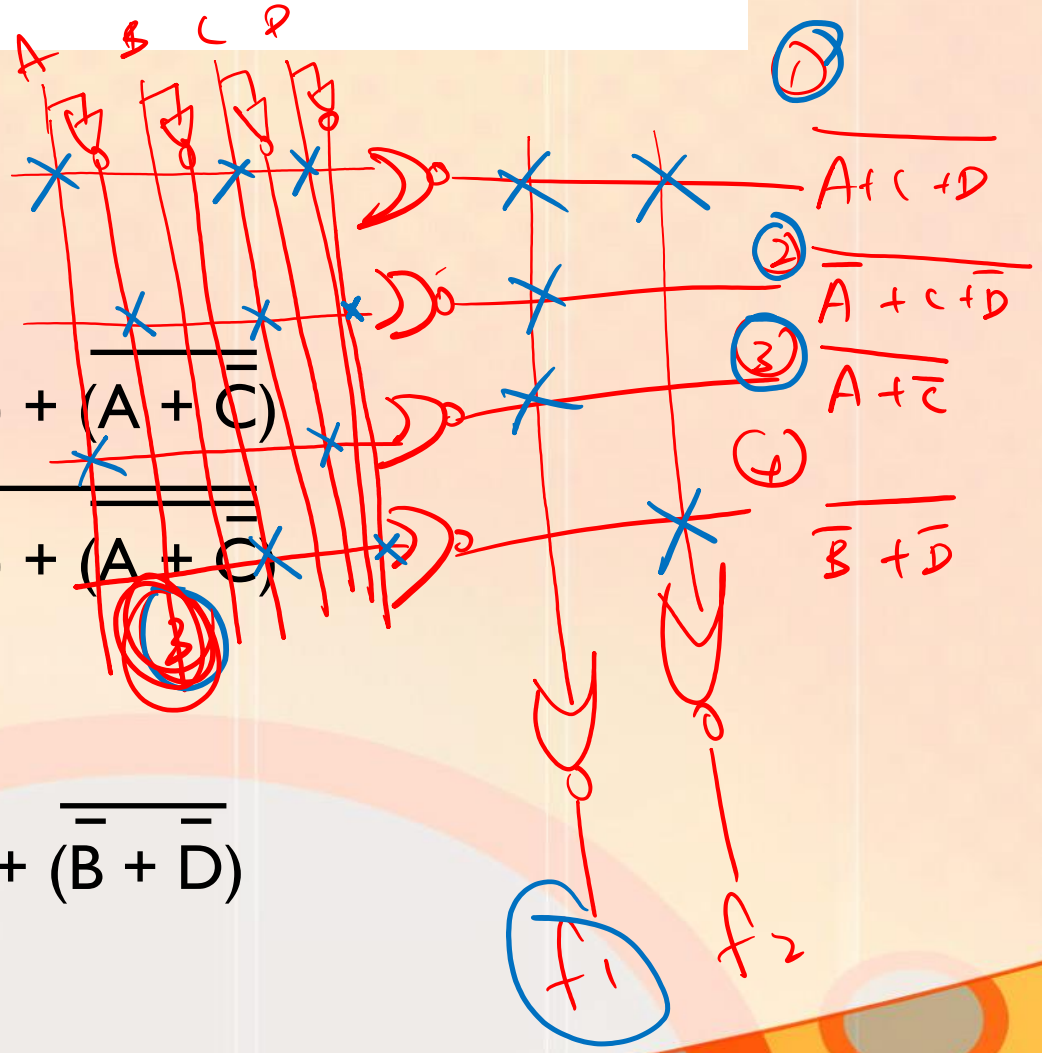
$$= \overline{(A + C + D)} + \overline{(\bar{A} + C + \bar{D})} + \overline{(A + \bar{C})}$$

$$\Rightarrow f_1 = \overline{(A + C + D)} + \overline{(\bar{A} + C + \bar{D})} + \overline{(A + \bar{C})}$$

- and

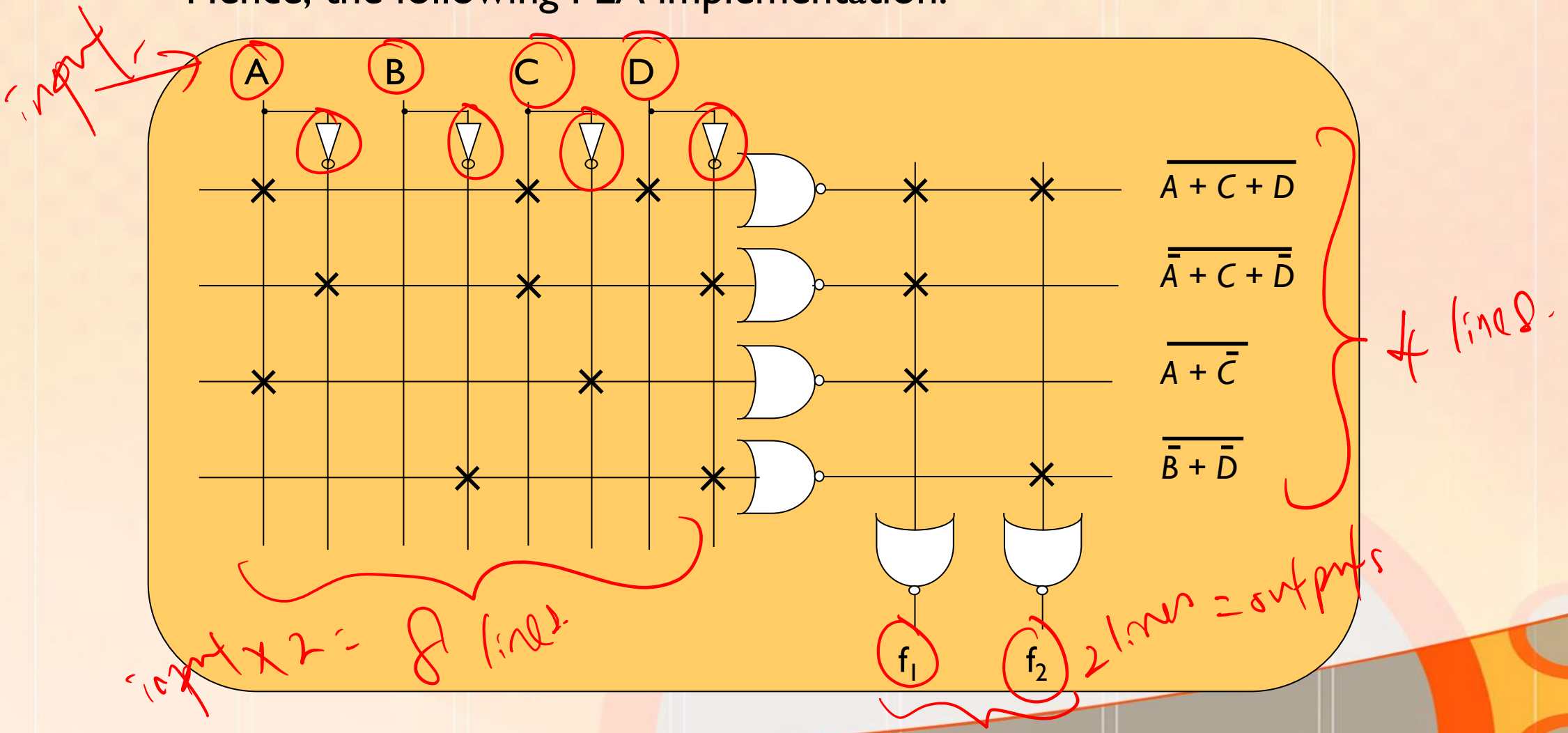
$$\overline{f_2} = \bar{A}\bar{C}\bar{D} + BD = \overline{(A + C + D)} + \overline{(B + D)}$$

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



PLA – Further Implementations

- Hence, the following PLA implementation:

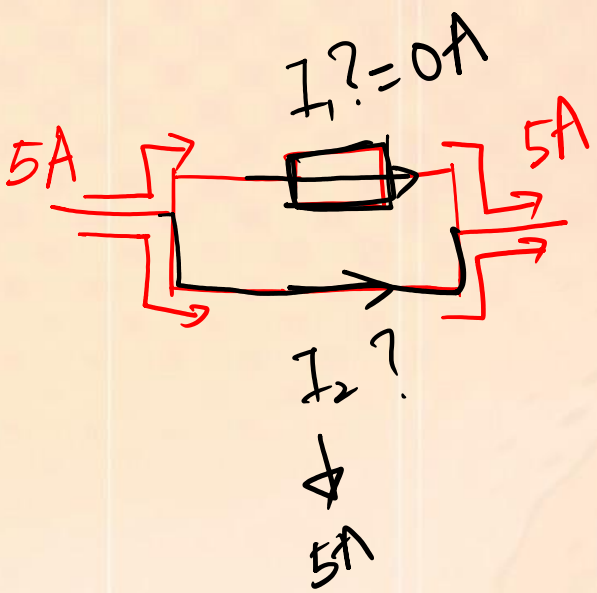


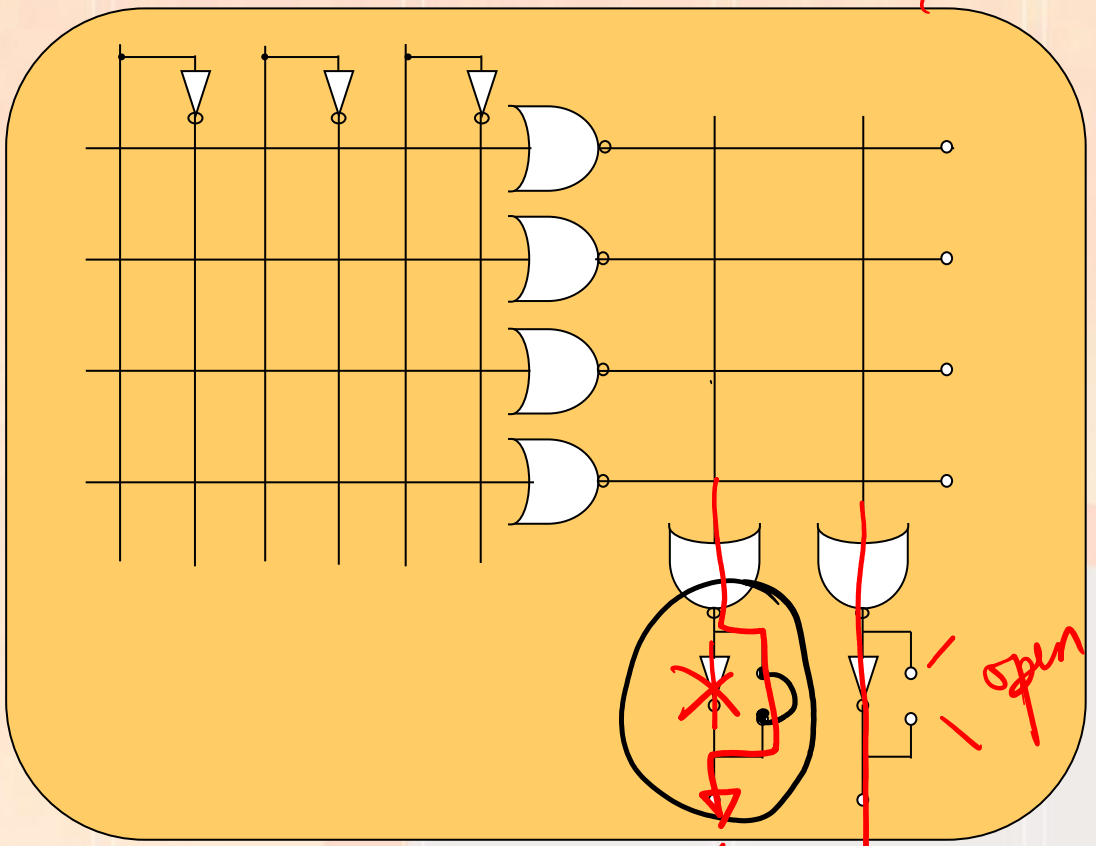
PLA – NOR-NOR-NOT

- Finally, a more flexible PLA implementation exists in the form of a NOR-NOR-NOT structure as follows:

NAND-NAND-NOT

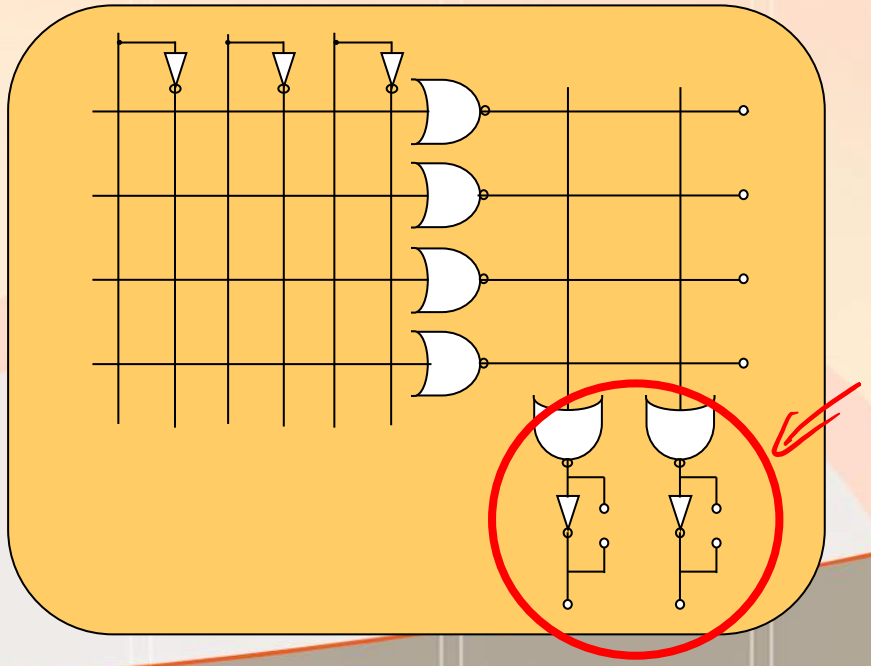
Allow 2 or 3 level





PLA – NOR-NOR-NOT

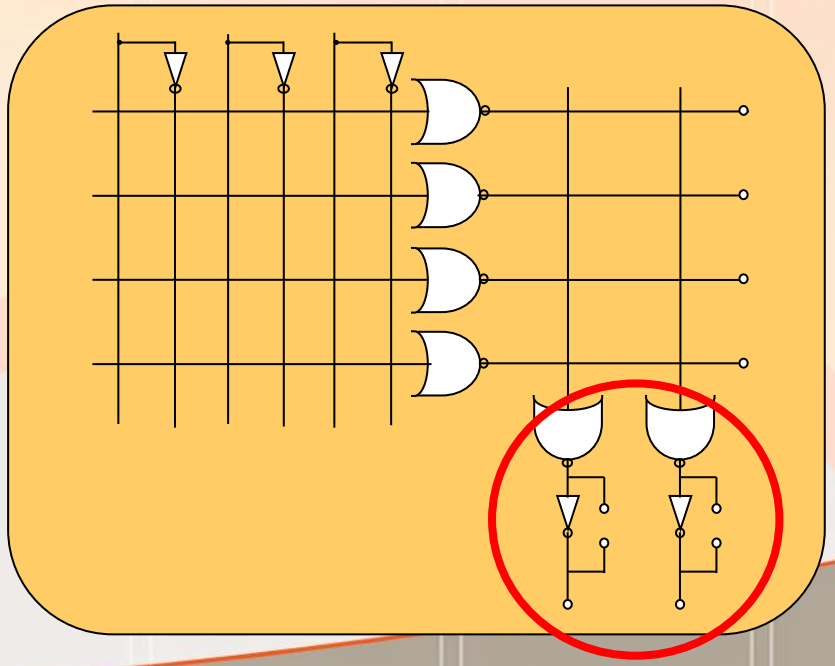
- With this structure we have the option of having an additional inverter on the output for each function.
- Hence, we now have the option of implementing functions as either NOR-NOR or NOR-NOR-NOT.
- In other words we can consider grouping both 0's and 1's in a Karnaugh Map if need be.



PLA – NOR-NOR-NOT

- Furthermore, as each function has both options available, we can now also consider grouping 0's for one function while grouping 1's for another function at the same time, to see if this produces an even better (and hence more minimal) solution.

- This structure gives us the most flexibility but also demands the most work in terms of the minimisation process.

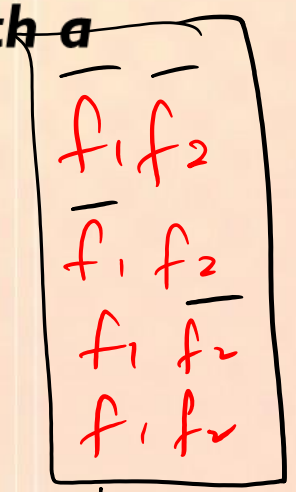


PLA – Further Implementations

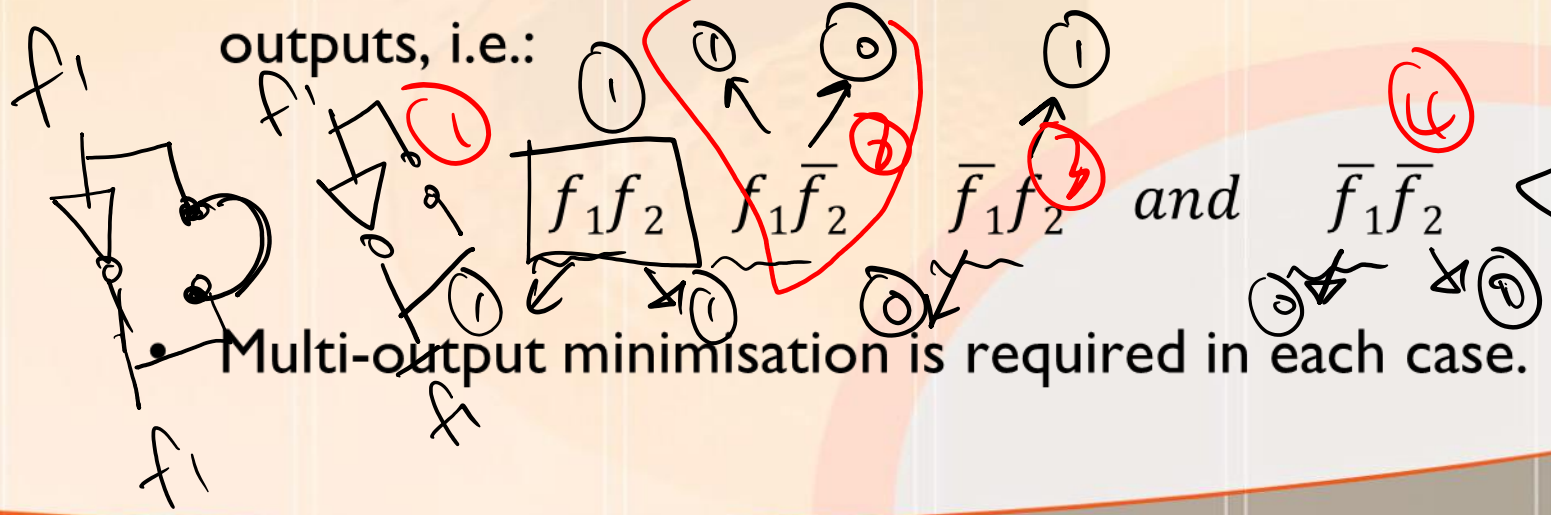
- ***Ex. 9.5 Implement the following functions using a PLA with a NOR-NOR-NOT architecture.***

$$f_1 = \Sigma(0, 1, 2, 3, 8, 9, 10, 11, 12)$$

$$f_2 = \Sigma(1, 3, 4, 6, 9, 11, 14)$$



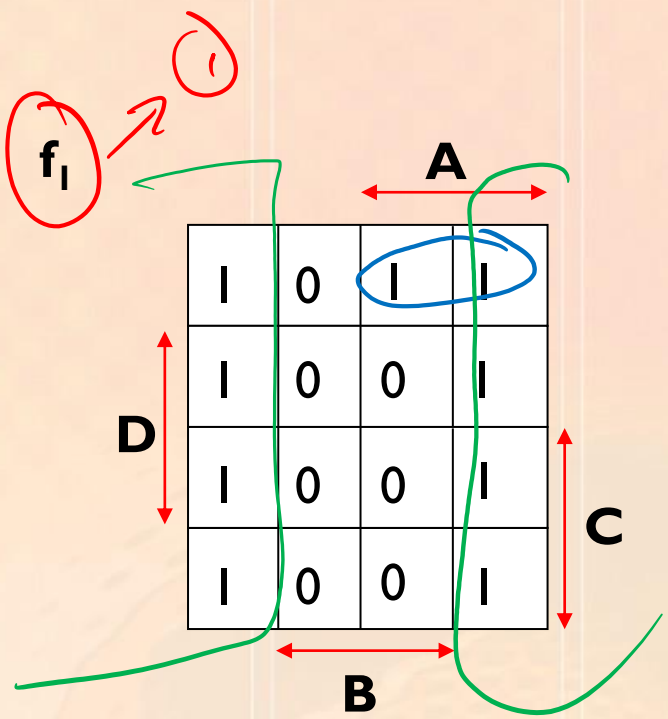
- For this structure, we have to consider all combinations of the outputs, i.e.:

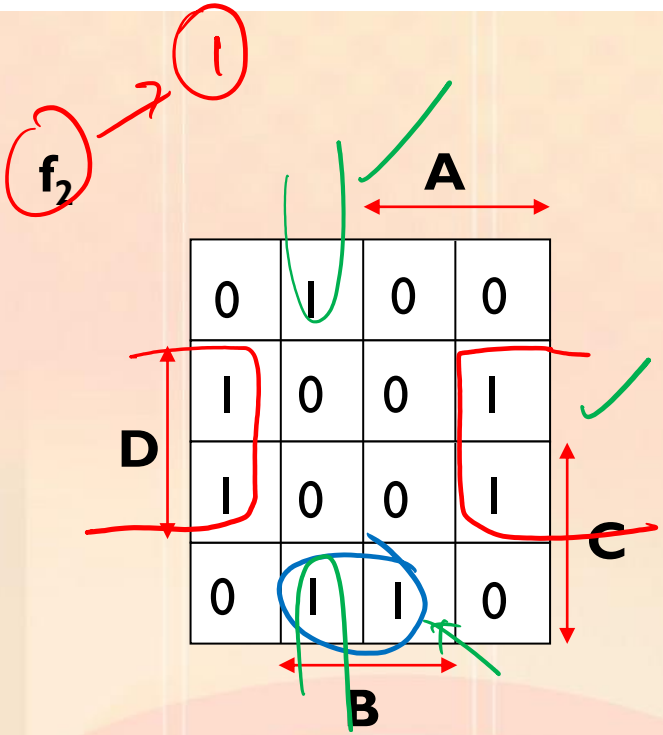


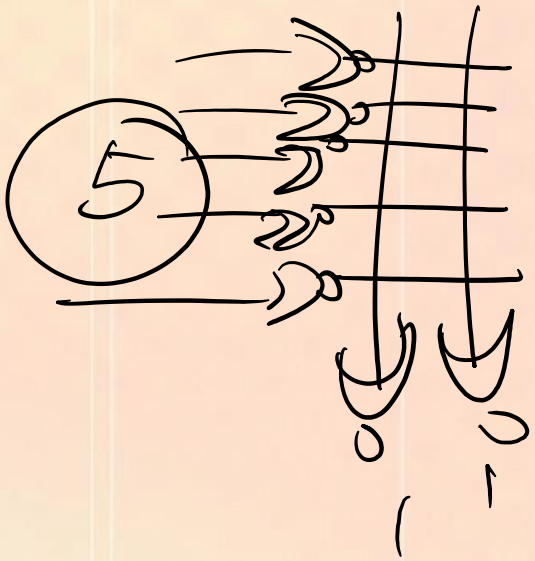
- Multi-output minimisation is required in each case.

PLA – Further Implementations

NOR-NOR-NOT







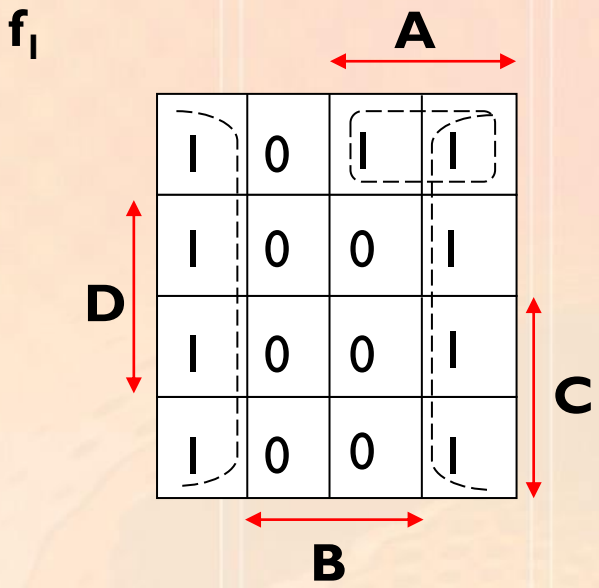
$f_1 = \bar{B} + A\bar{C}\bar{D}$

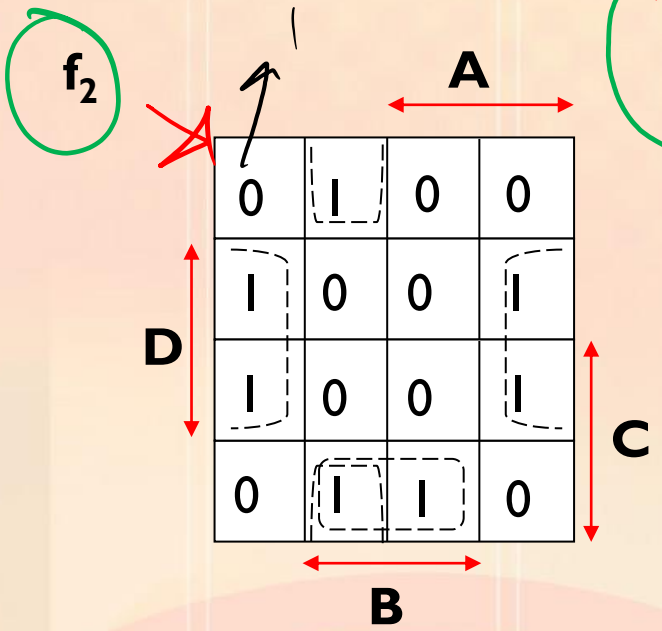
$= \overline{B} + \overline{\bar{A} + C + D}$

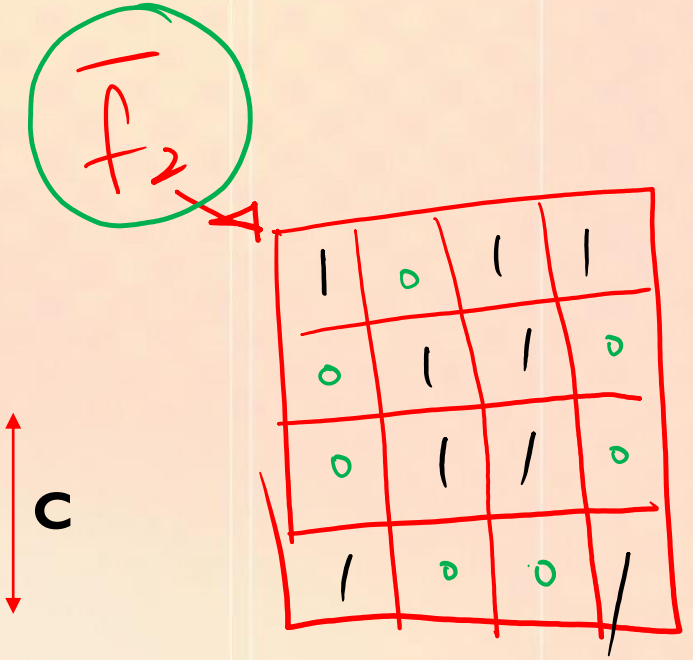
$f_2 = A\bar{B}\bar{D} + \bar{A}B\bar{D} + BC\bar{D}$

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

PLA – Further Implementations

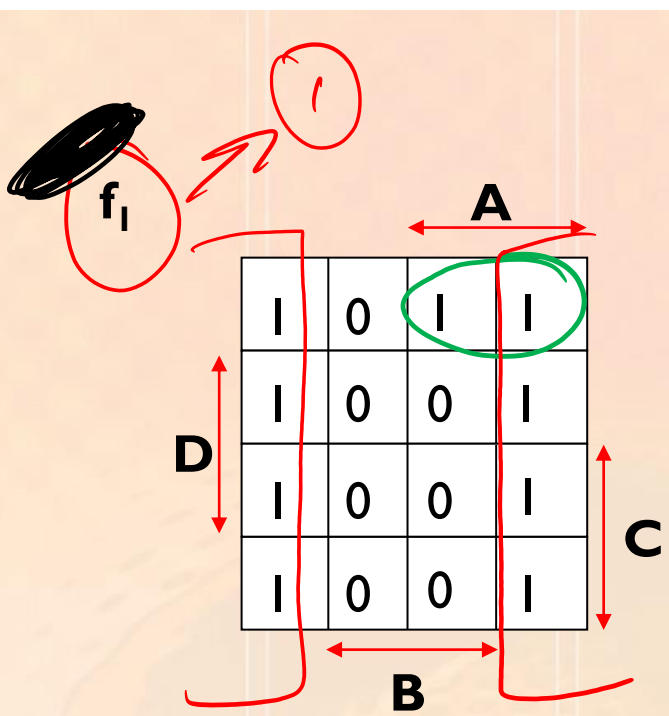


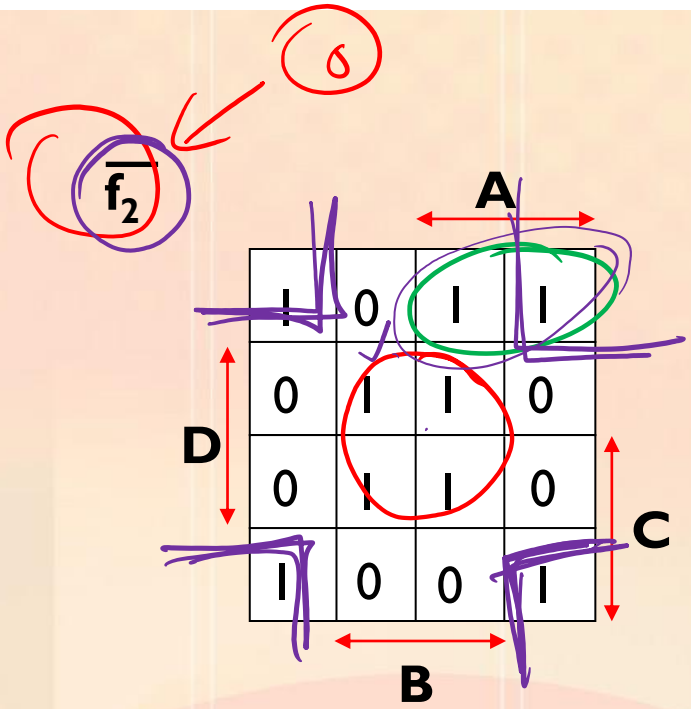


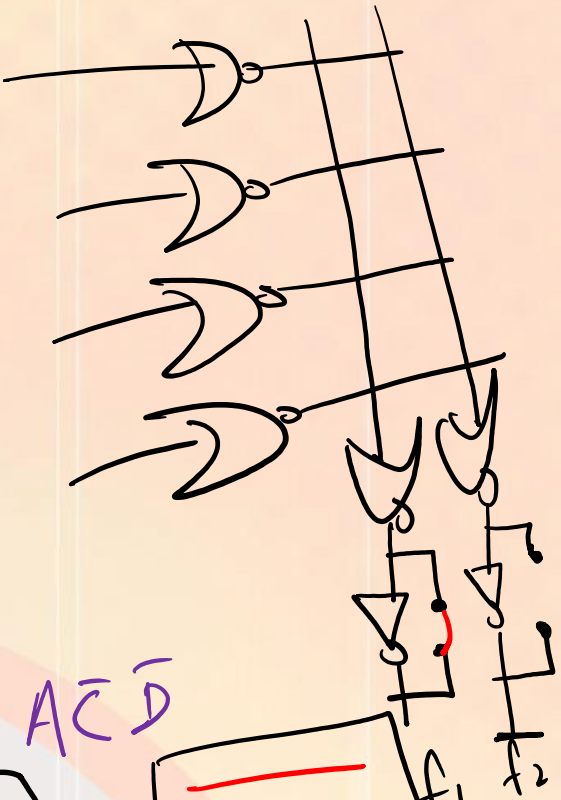


$f_1 f_2$ – needs 5 product terms (i.e. 5 groups)

PLA – Further Implementations







$$f_1 = \bar{B} + A\bar{C}\bar{D}$$

$$\bar{f_1} = \overline{\bar{B} + \overline{A+C+D}}$$

(1) (2)

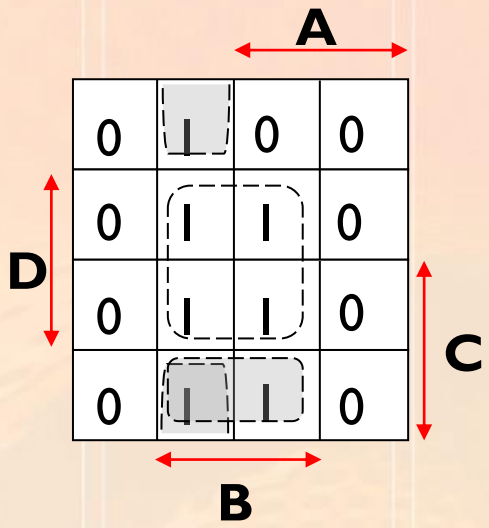
$$f_2 = BD + \bar{B}\bar{D} + A\bar{C}\bar{D}$$

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

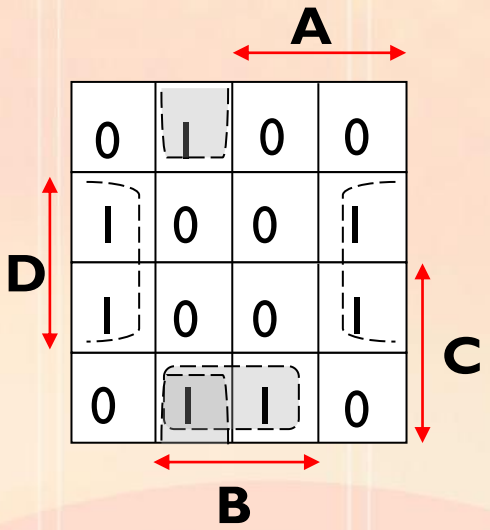
(3) (4) (2)

PLA – Further Implementations

$\overline{f_1}$

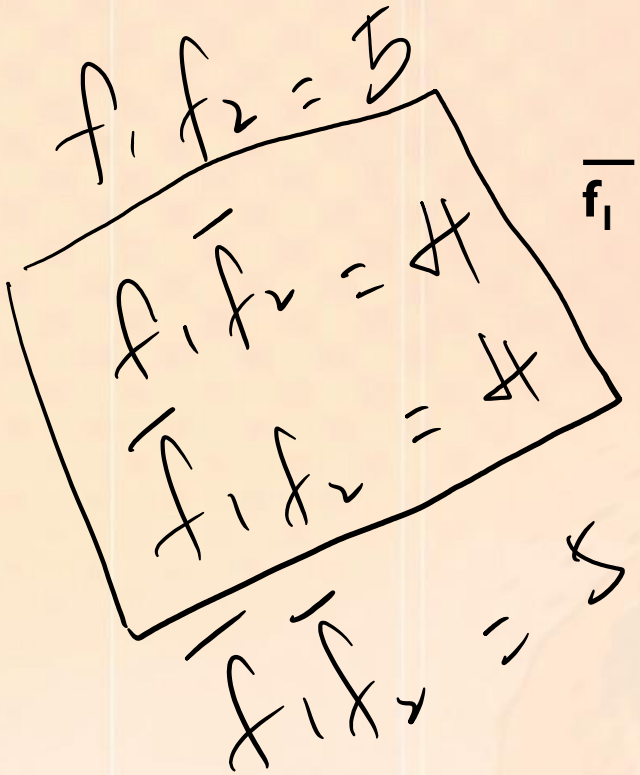


f_2

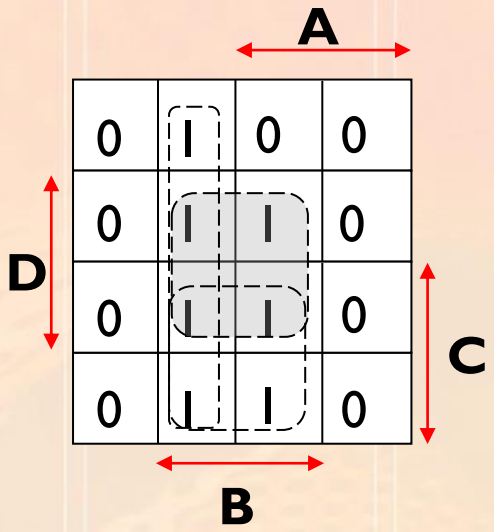


$\overline{f}_1 f_2$ – needs 4 product terms

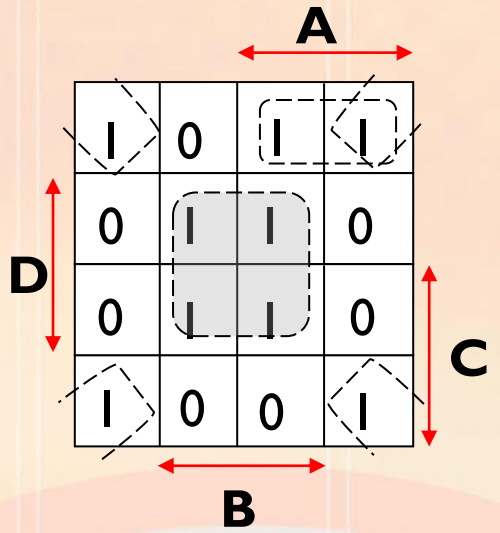
PLA – Further Implementations



$\overline{f_1}$



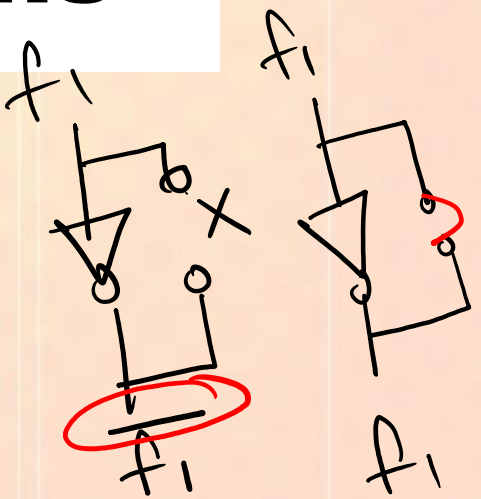
$\overline{f_2}$



$\overline{f_1}\,\overline{f_2}$ – needs 5 product terms

PLA – Further Implementations

- Both $f_1\bar{f}_2$ and $\bar{f}_1 f_2$ offer the best minimisation solutions.
- Taking $f_1\bar{f}_2$, we can write down the following solution:



$$f_1 = \bar{B} + A\bar{C}\bar{D} = \overline{\bar{B} + (\bar{A} + C + D)}$$

NOR-NOR-NOT format

and

$$\bar{f}_2 = BD + \bar{B}\bar{D} + A\bar{C}\bar{D} = (\bar{B} + \bar{D}) + (B + D) + (\bar{A} + C + D)$$

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

NOR-NOR format

PLA – Further Implementations

- Hence the PLA implementation is given as: