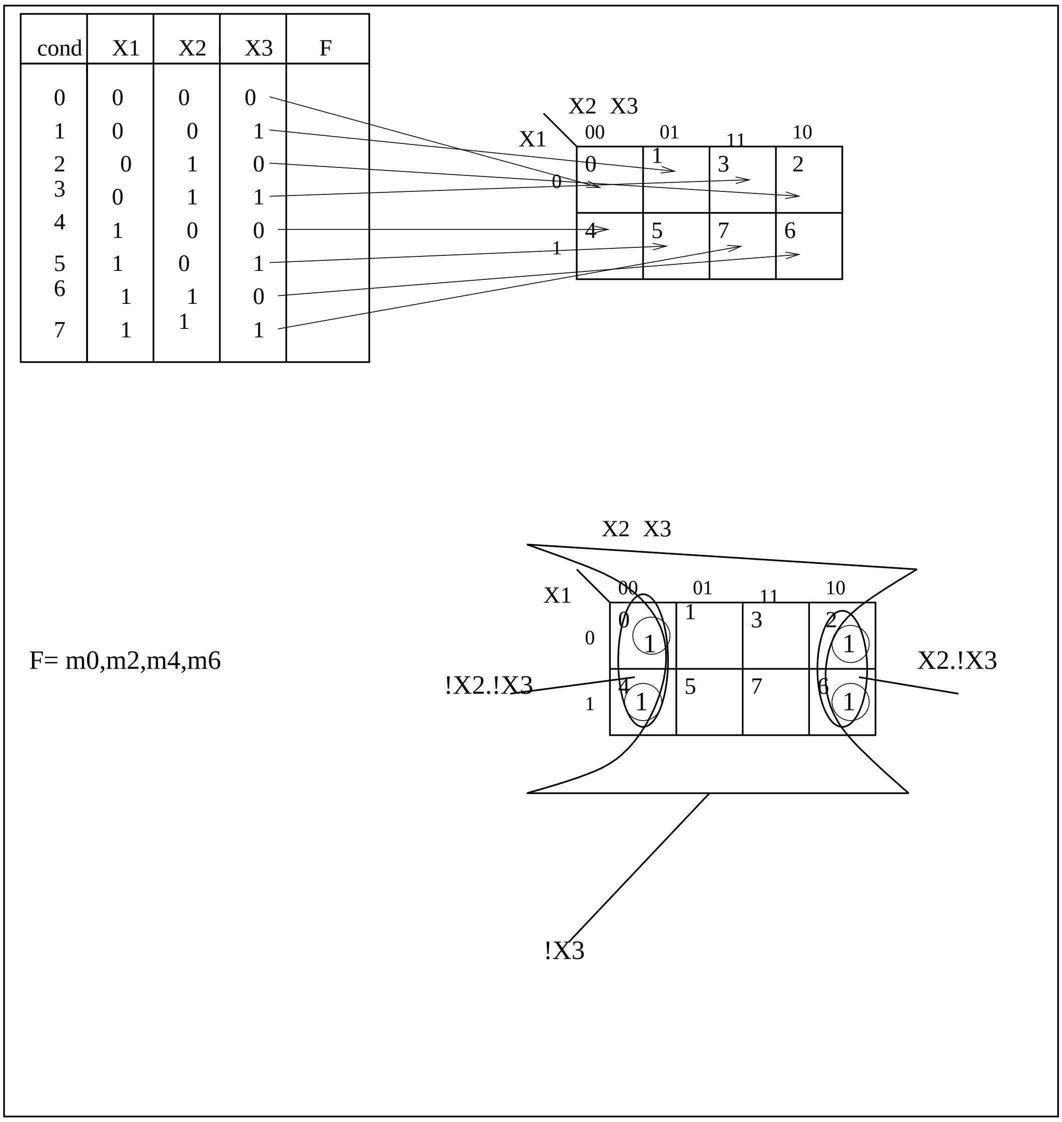
Logic Optimization and Implementation
Optimization Using Karnaugh Map

Concept: any two min or Max terms could be reduced to one term if they differ in one variable

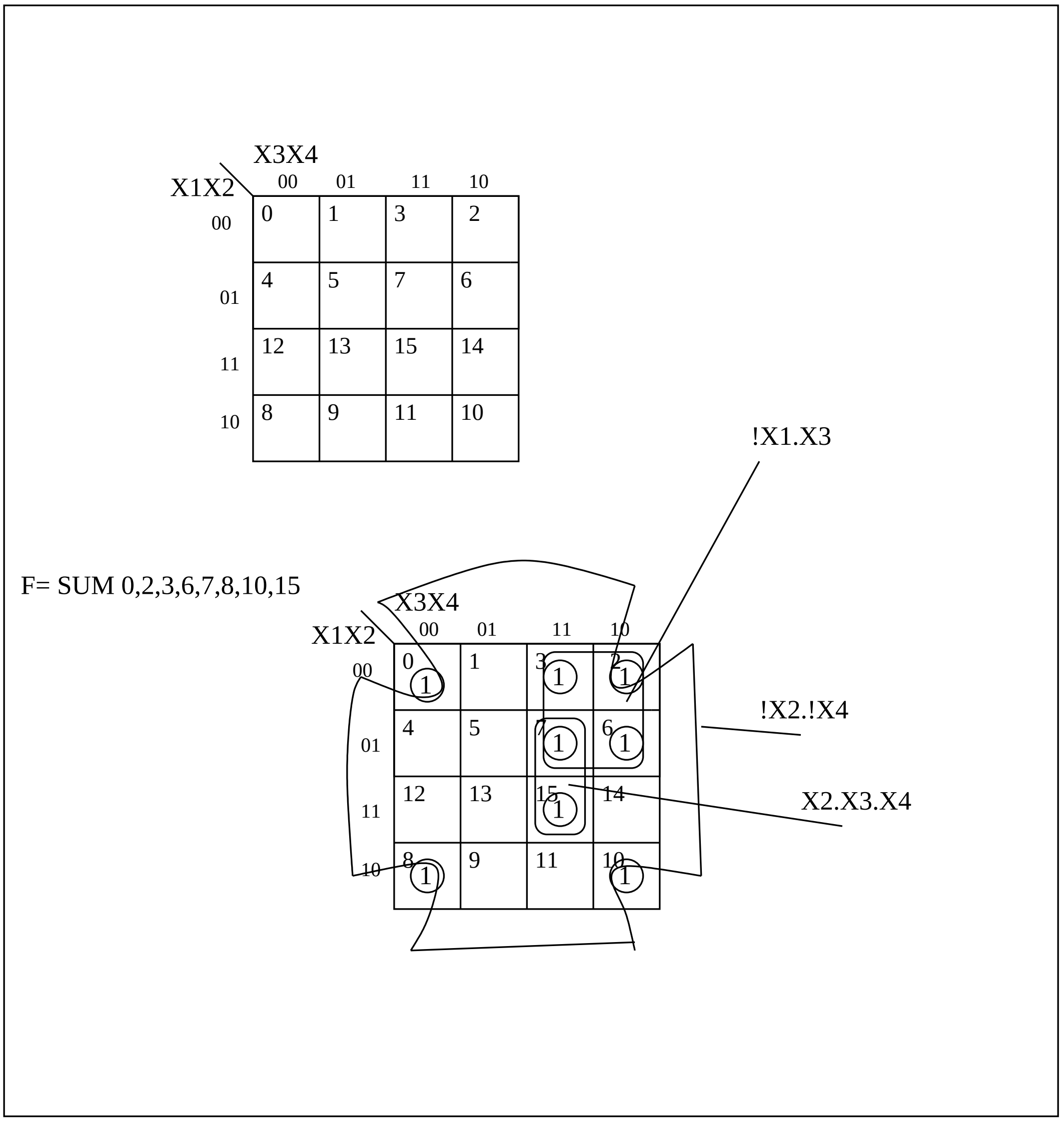
Example: X1.!X2.X3 + X1.!X2.!X3 = X1.!X2

(X1+!X2+X3).(X1+!X2+!X3)= X1+!X2



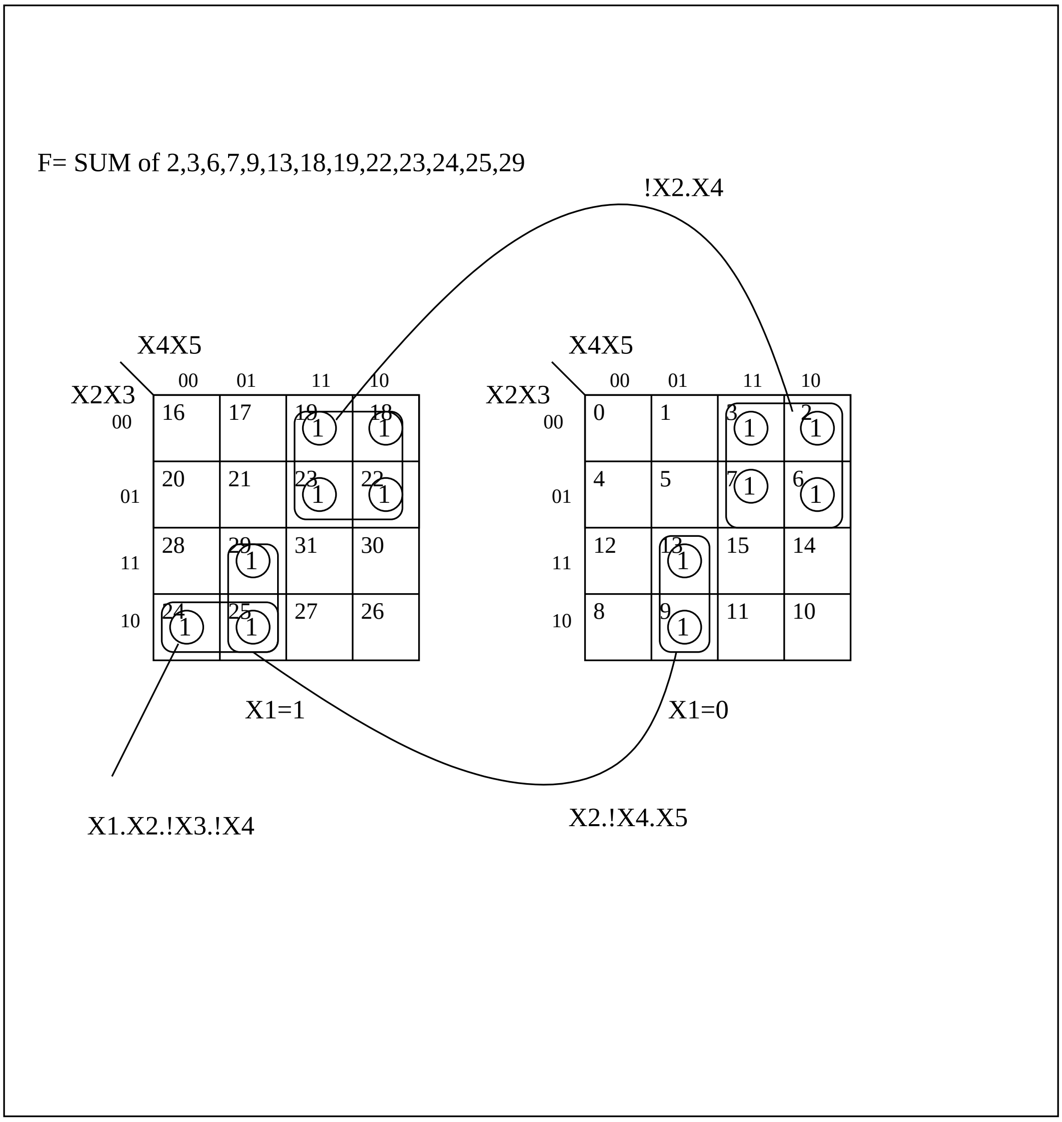
K-Map for 4 input variables

Example: $\sum 0, 2, 3, 6, 7, 8, 10, 15$



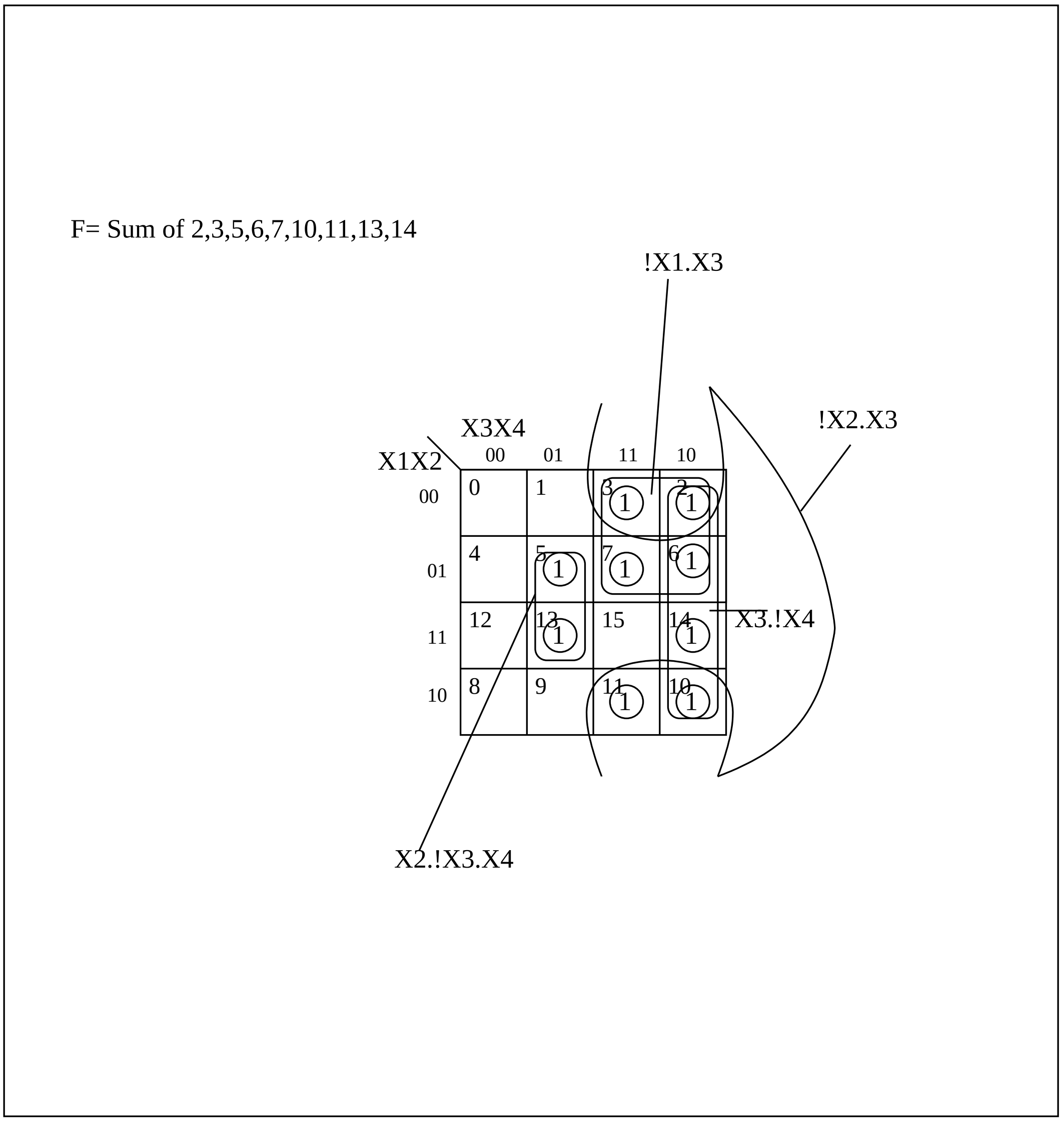
K-Map for 5 input variables

Example: $\sum 2, 3, 6, 7, 9, 13, 18, 19, 22, 23, 24, 25, 29$



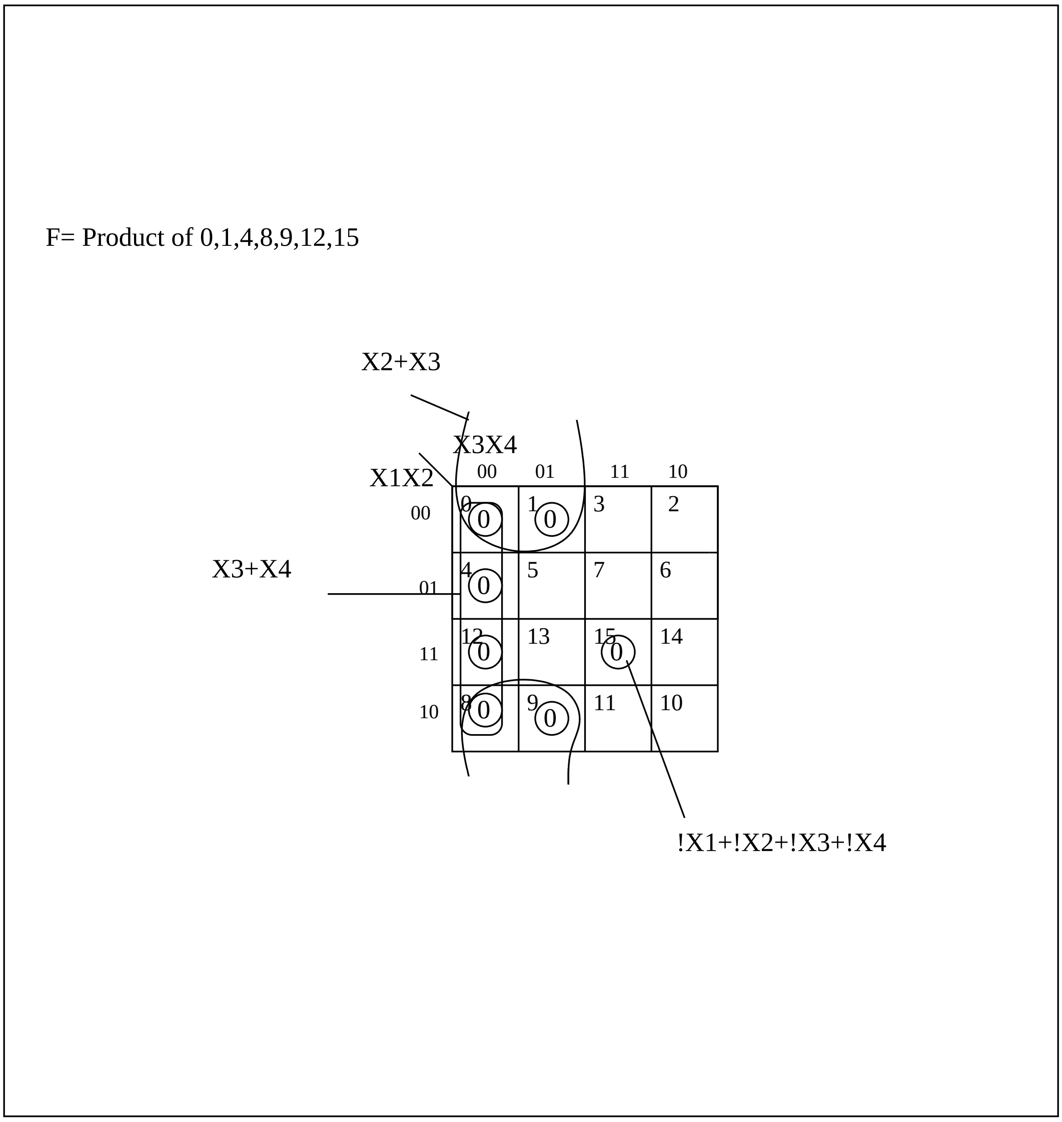
K-Map for 4 input variables

Example: $\sum 2, 3, 5, 6, 7, 10, 11, 13, 14$



K-Map for 4 input variables

Example: $\prod 0, 1, 4, 8, 9, 12, 15$



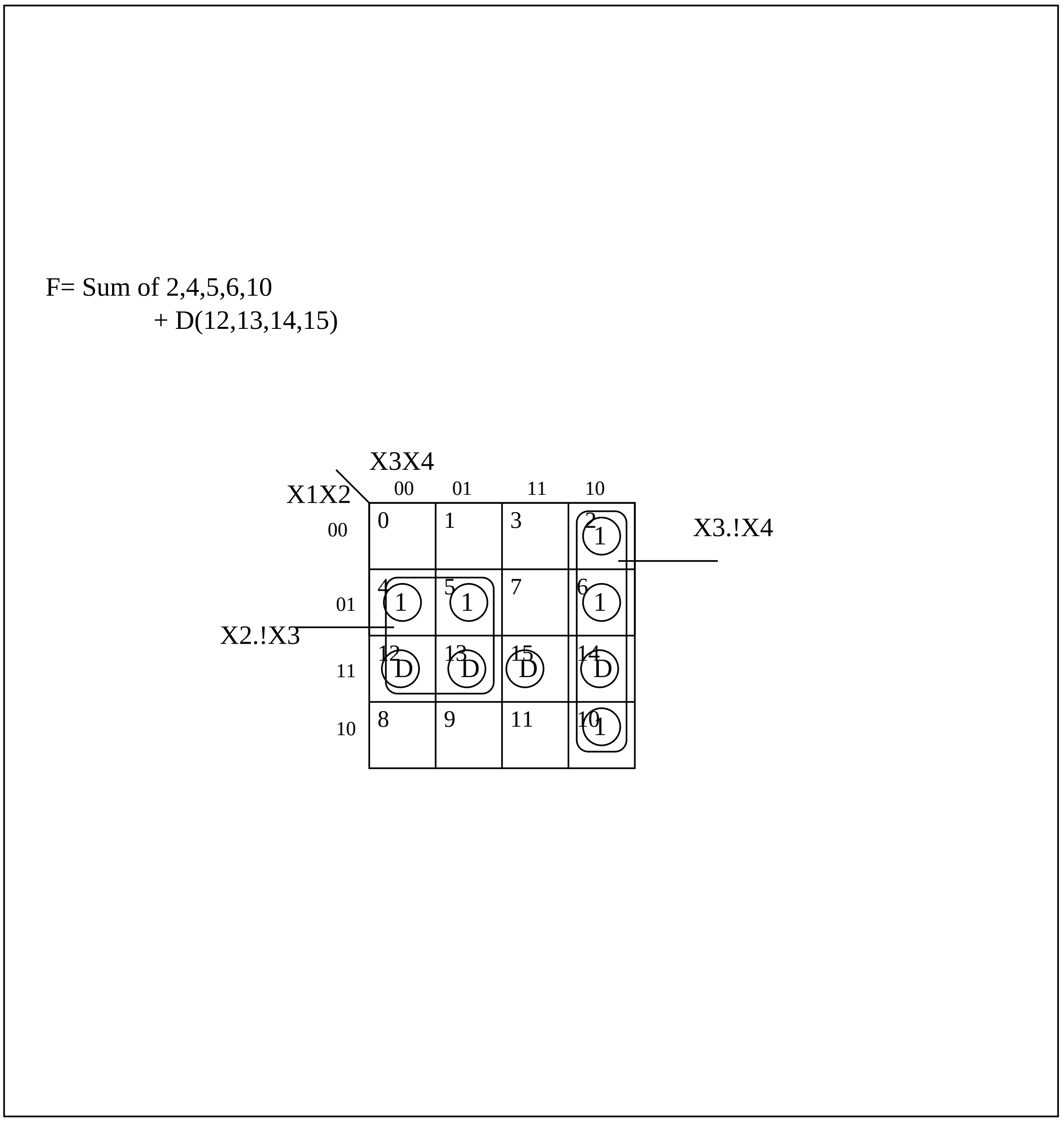
©N. Mekhiel

Optimization of Incompletely Specified Functions

Gives more flexibility and therefore better optimization

DO NOT CARE conditions could be 0 or 1

Example: $\sum 2, 4, 5, 6, 10 + D(12, 13, 14, 15)$

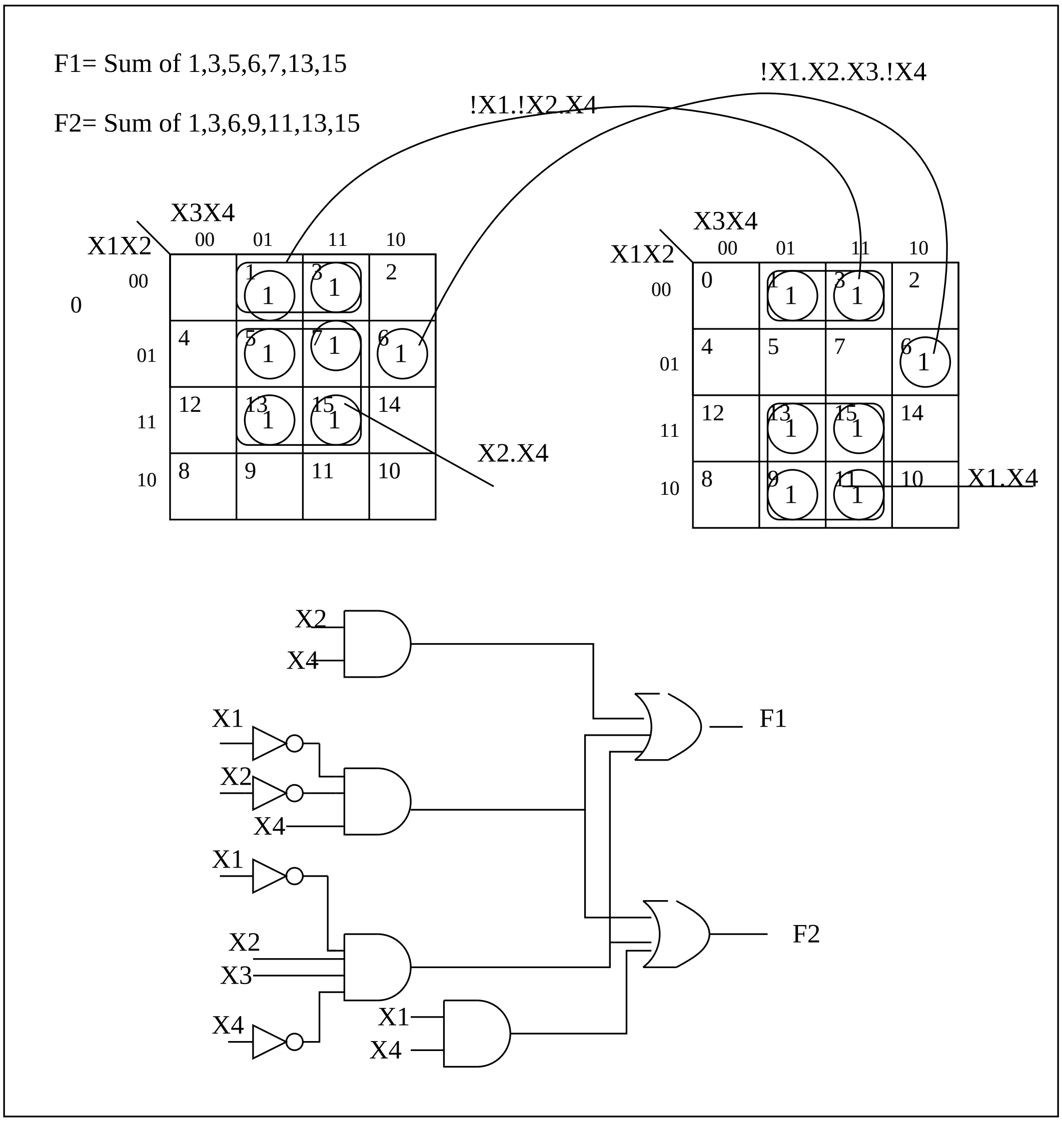


Multiple Output Circuits

Could share circuits, different from optimization of individual functions

Example:F1= $\sum 1, 3, 5, 6, 7, 13, 15$

F2= $\sum 1, 3, 6, 9, 11, 13, 15$

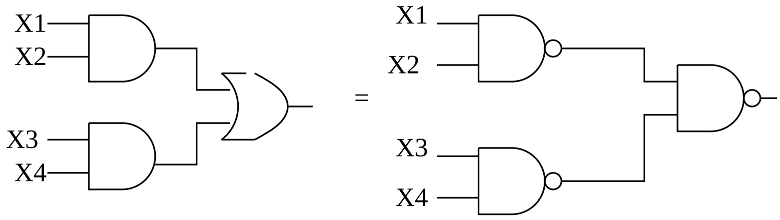


Using NAND and NOR ONLY

SUM OF PRODUCT COULD USE NAND ONLY
(NAND-NAND, rather than AND-OR)
PRODUCT OF SUMS COULD USE NOR ONLY
(NOR-NOR, rather than OR-AND)

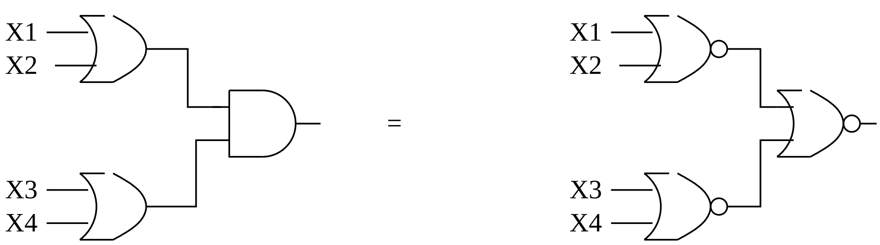
F=X1.X2+X3.X4 = !!(X1.X2+X3.X4)

= !(!(X1.X2).!(X3.X4)



F= (X1+X2).(X3+X4) = !!((X1+X2)(X3+X4))

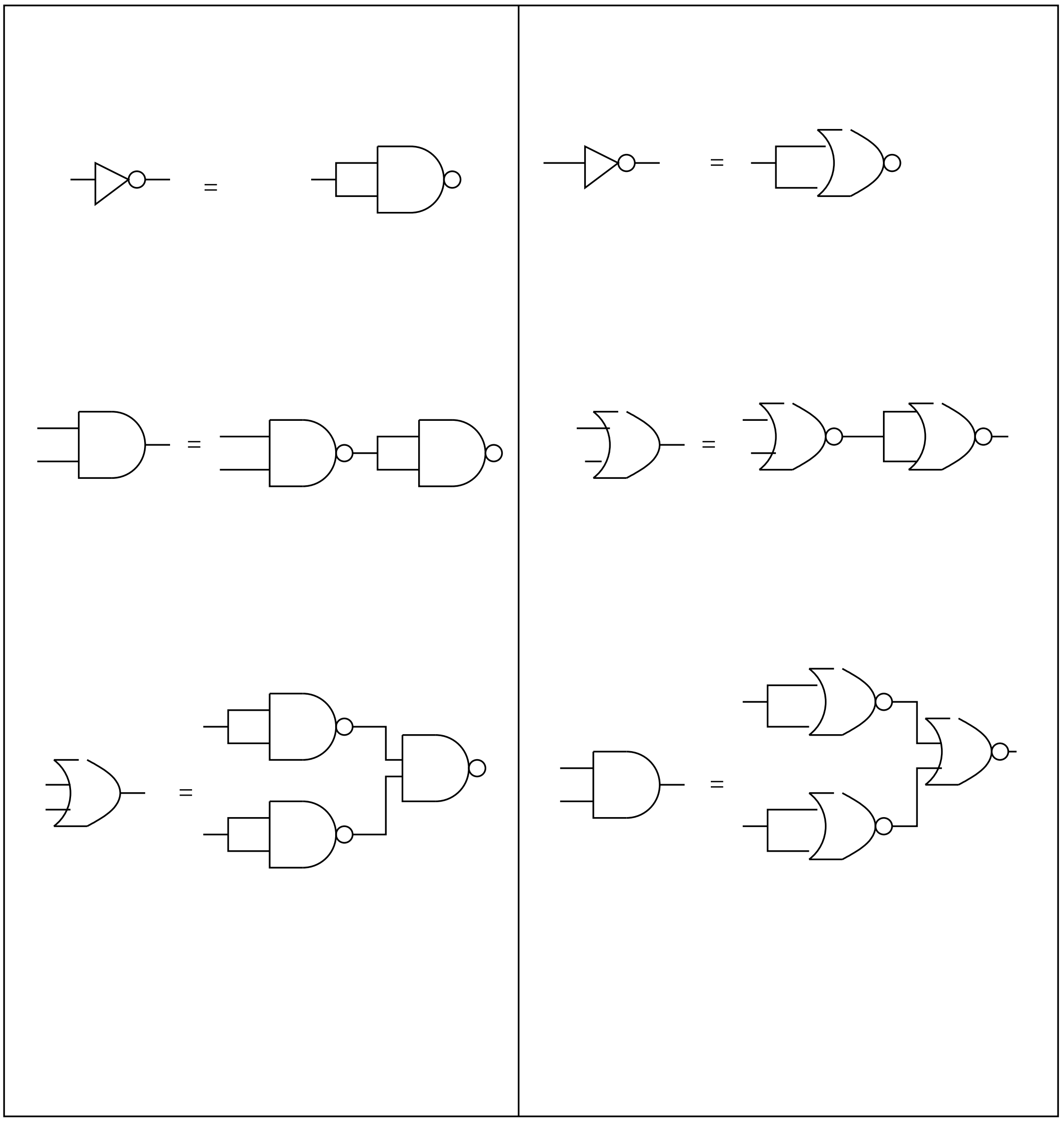
= !(!(X1+X2)+!(X3+X4))



Using NAND and NOR ONLY

ANY GATE COULD BE CONSTRUCTED USING NAND GATES

ANY GATE COULD BE CONSTRUCTED USING NOR GATES

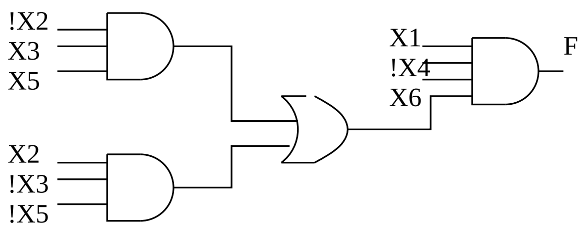


Multi-Level Synthesis

Using Factoring: For large number of inputs and to reduce wire complexity but it increases circuit delay

F= X1.!X2.X3.!X4.X5.X6 + X1.X2.!X3.!X4.!X5.X6

= X1.!X4.X6.(!X2.X3.X5 + X2.!X3.!X5)

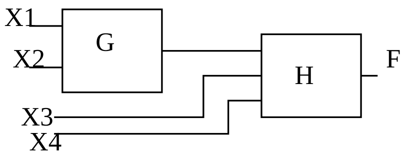


Factoring

Multi-Level Synthesis

Using Decomposition: Converts large circuit to several subcircuits to reduce circuit complexity but it increases circuit delay

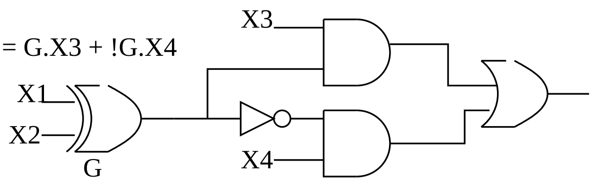
F(X1,X2,X3,X4) = H(G(X1,X2), X3,X4)



F= !X1.X2.X3 + X1.!X2.X3 + X1.X2.X4 + !X1.!X2.X4

= (!X1.X2 + X1.!X2).X3 + (X1.X2 + !X1.!X2).X4   assume G =!X1.X2 + X1.!X2

= G.X3 + !G.X4



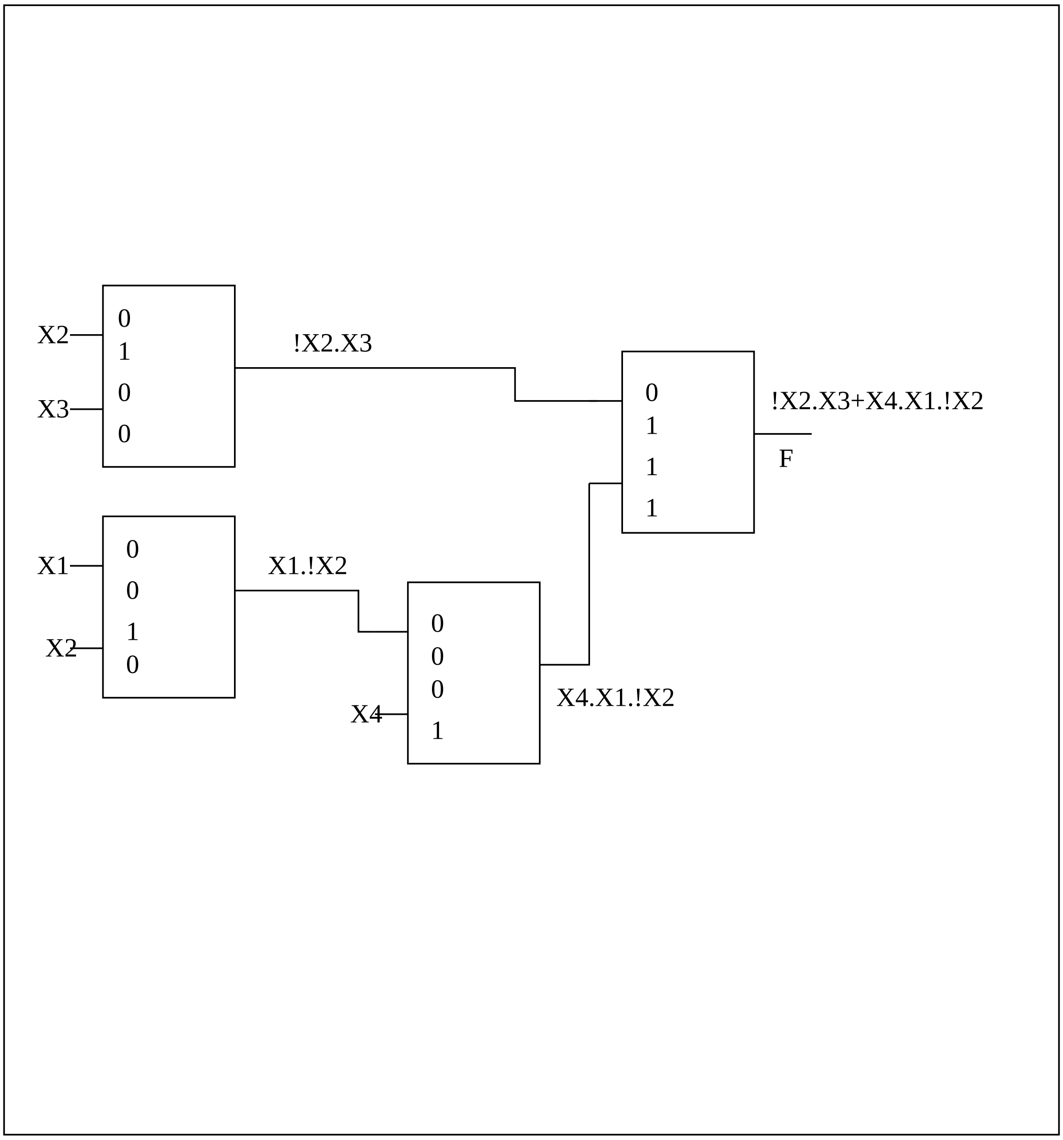
DECOMPOSITION

Implement $F = !X2.X3 + X1.!X2.X4$ Using VHDL IN FPGA

```
   ENTITY Funct IS
         PORT(X1,X2,X3,X4 : IN  STD_LOGIC;
              F           : OUT STD_LOGIC);
   END Funct;

ARCHITECTURE LogicFunction OF Funct IS
BEGIN
   F<=(NOT X2 AND X3) OR (X1 AND NOT X2 AND X4);
END LogicFunction;
```

Implement $F = !X2.X3 + X1.!X2.X4$ Using VHDL IN FPGA

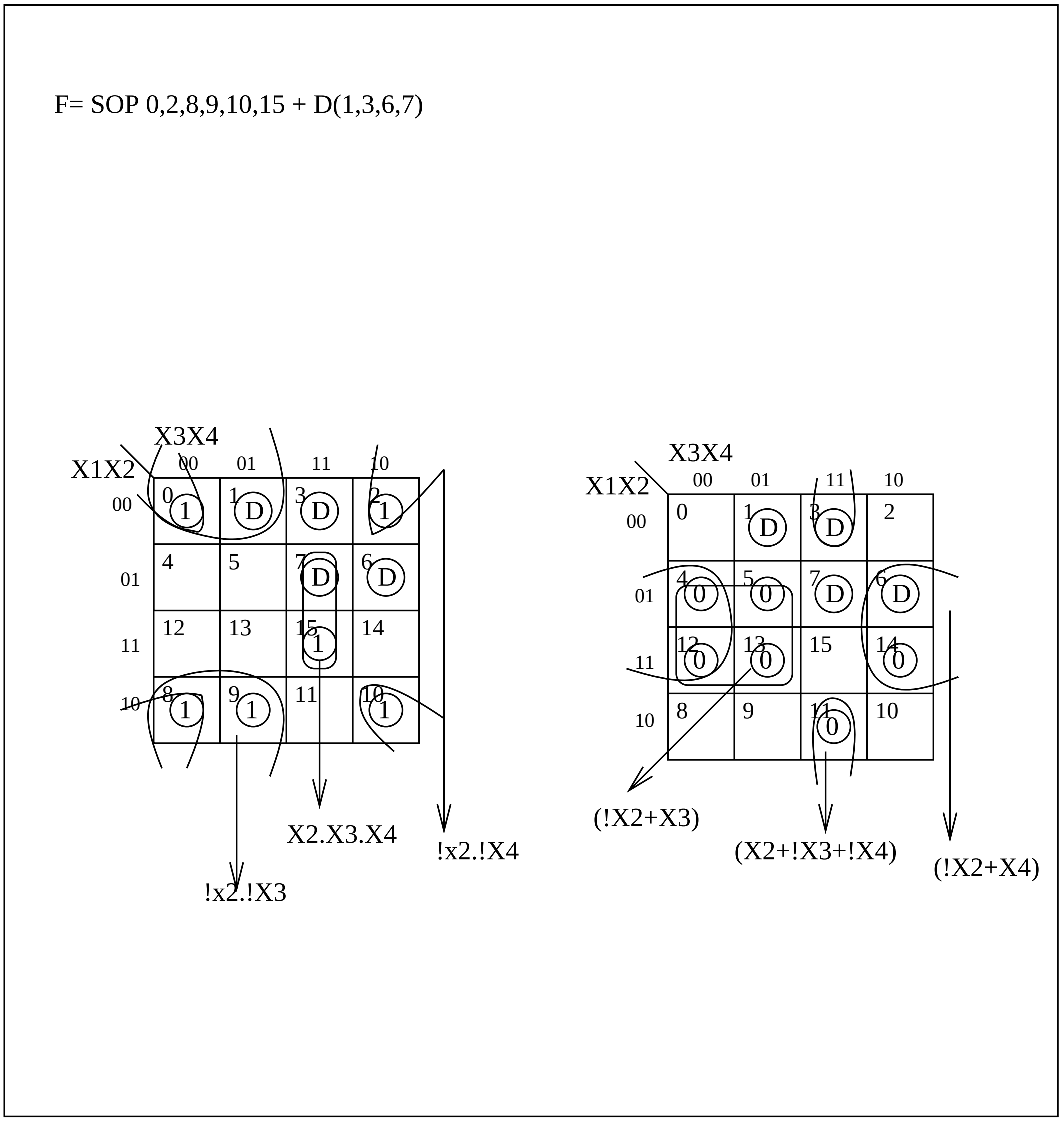


CH4 4-4, 410, 4-12, 4-14, 4-22

Find Optimal design for $\sum 0, 2, 8, 9, 10, 15 + D(1, 3, 6,$

F=!X2.!X3+!X2.!X4+X2.X3.X4

F=(!X2+X3).(!X2+X4).(X2+!X3+!X4)



4-10

F=X1.X2.!X3+!X1.X2.X4+X1.!X2.X4+X1.X3.!X4
+!X1.X3.X4+X1.X3.!X4

4-12 Two outputs: $f = \sum 0, 2, 4, 6, 7, 9 + D(10, 11)$
$g = \sum 2, 4, 9, 10, 15 + D(0, 13, 14)$

F=Sum of 0,2,4,6,7,9 + D(10,11)

G=Sum of 2,4,9,10,15 + D (0,13,14)