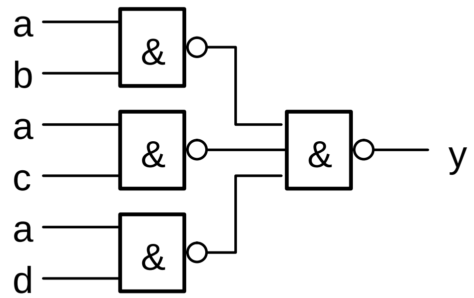
Solutions for Chapter 10

Exercise 10.1

a. By complementing the y function twice and applying De Morgan's theorem you get:

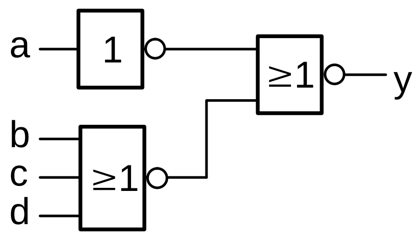
$$y = \overline{\overline{ab \vee ac \vee ad}} = \overline{\overline{ab} \wedge \overline{ac} \wedge \overline{ad}}$$



$$n_{Tr} = 3n_{Tr,NAND2} + n_{Tr,NAND3} = 18$$

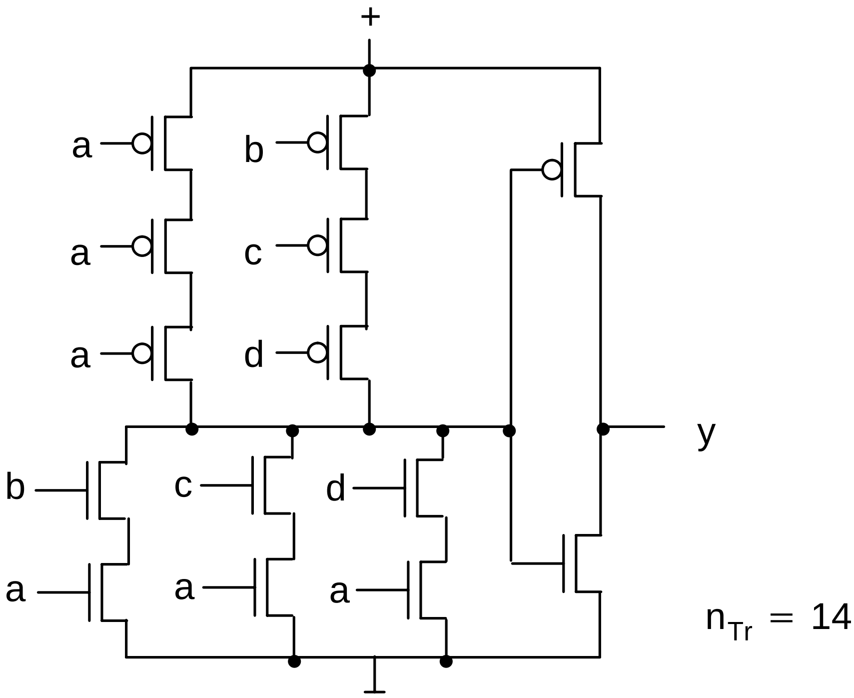
b. By applying the distributive law and the De Morgan's theorem you get following NOR–NOR structure:

$$\begin{aligned} y &= ab \vee ac \vee ad = a \wedge (b \vee c \vee d) \\ &= \overline{\overline{a \wedge (b \vee c \vee d)}} = \overline{\overline{a} \vee \overline{(a \vee b \vee c)}} \end{aligned}$$

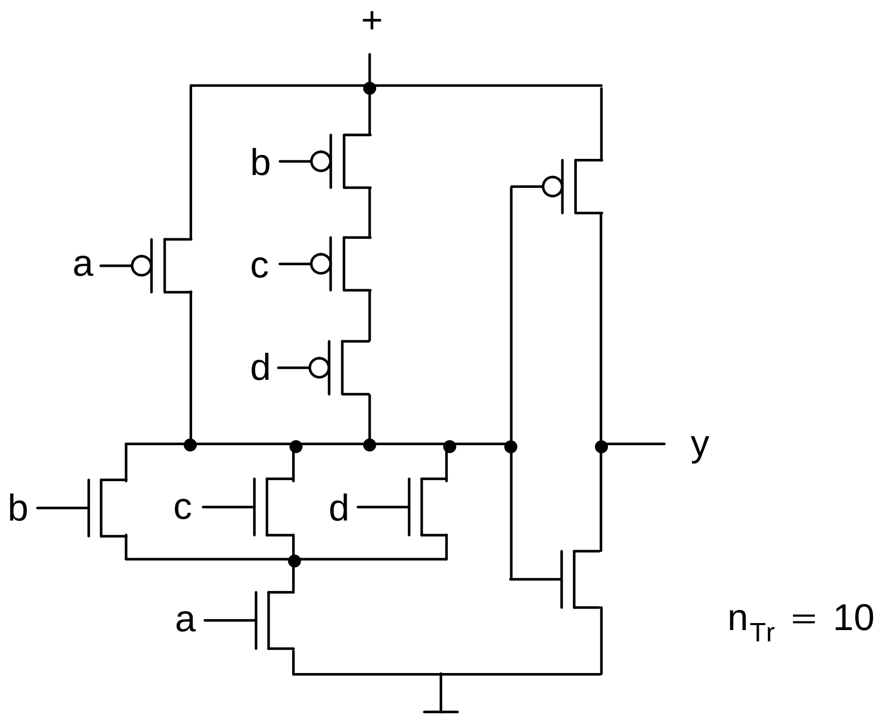


$$n_{Tr} = n_{Tr,INN} + n_{Tr,NOR3} + n_{Tr,NOR2} = 12$$

c. The following picture shows the direct conversion of the y function without factoring out of the variable a as complex gate.

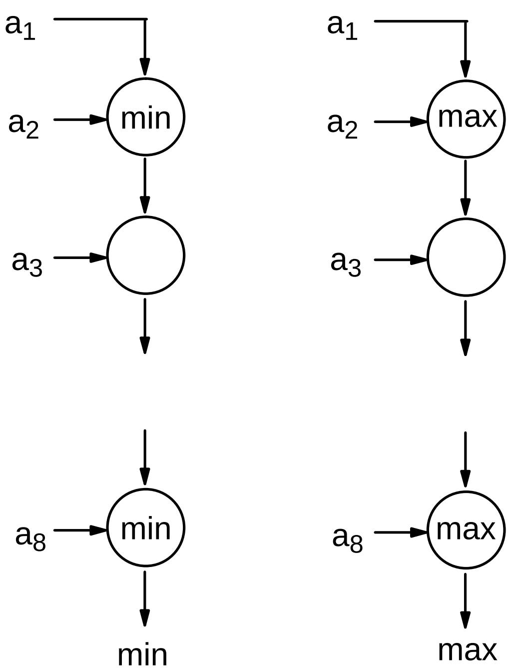


By factoring out of the variable a you can save transistors in the pull–up as well as in the pull–down path.

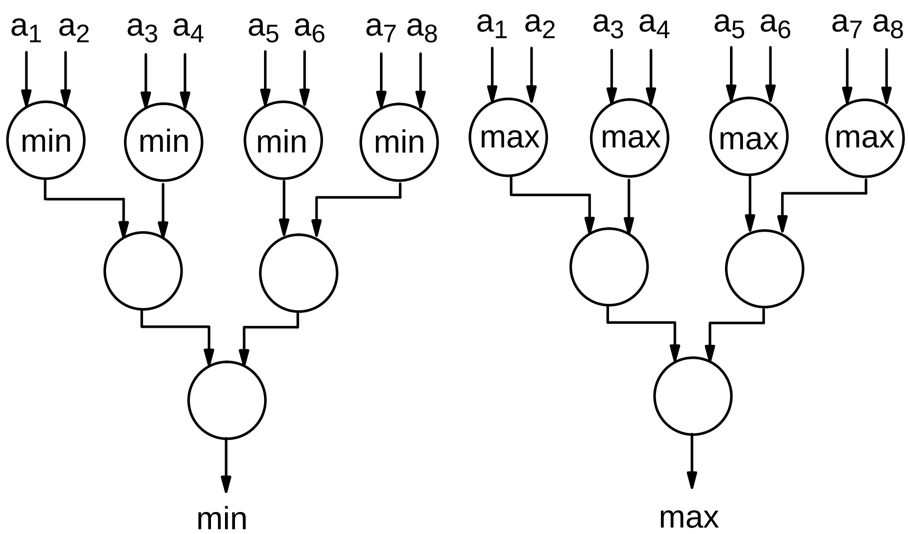


Exercise 10.2

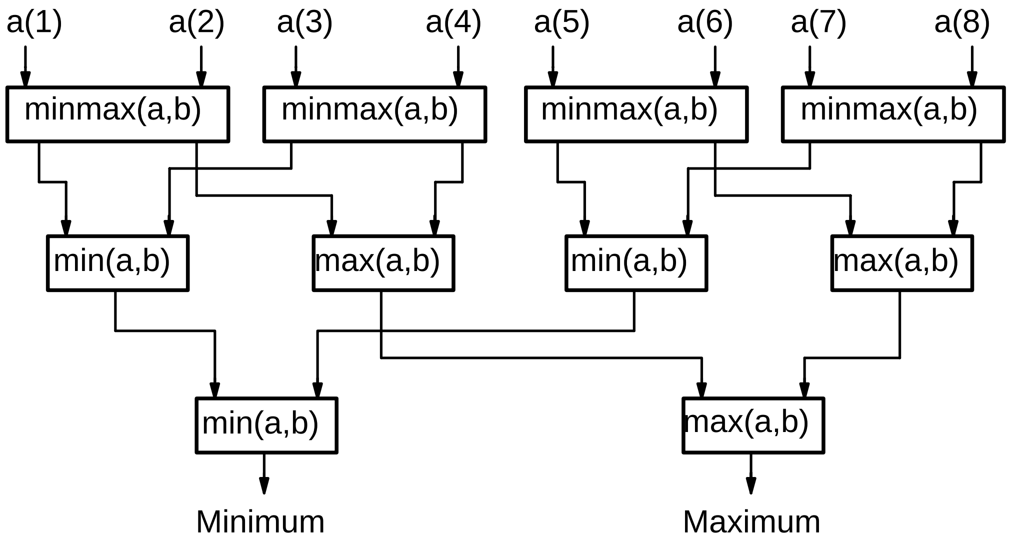
a. The following picture shows the dependency graph for the sequential extraction of the maximum and the minimum for 8 values. The nodes exist of a comparator and a following selector. For the minimum search the input operand with the smaller value is passed through and for the maximum search the operand with the greater value.



b. The hierarchical structure of the maximum and minimum search for 8 input operands with separated trees is shown in the following picture. The effort with 7 nodes for minimum and maximum search is the same like the sequential approach in a.) whereas the delay is reduced from 7 steps to 3 steps.

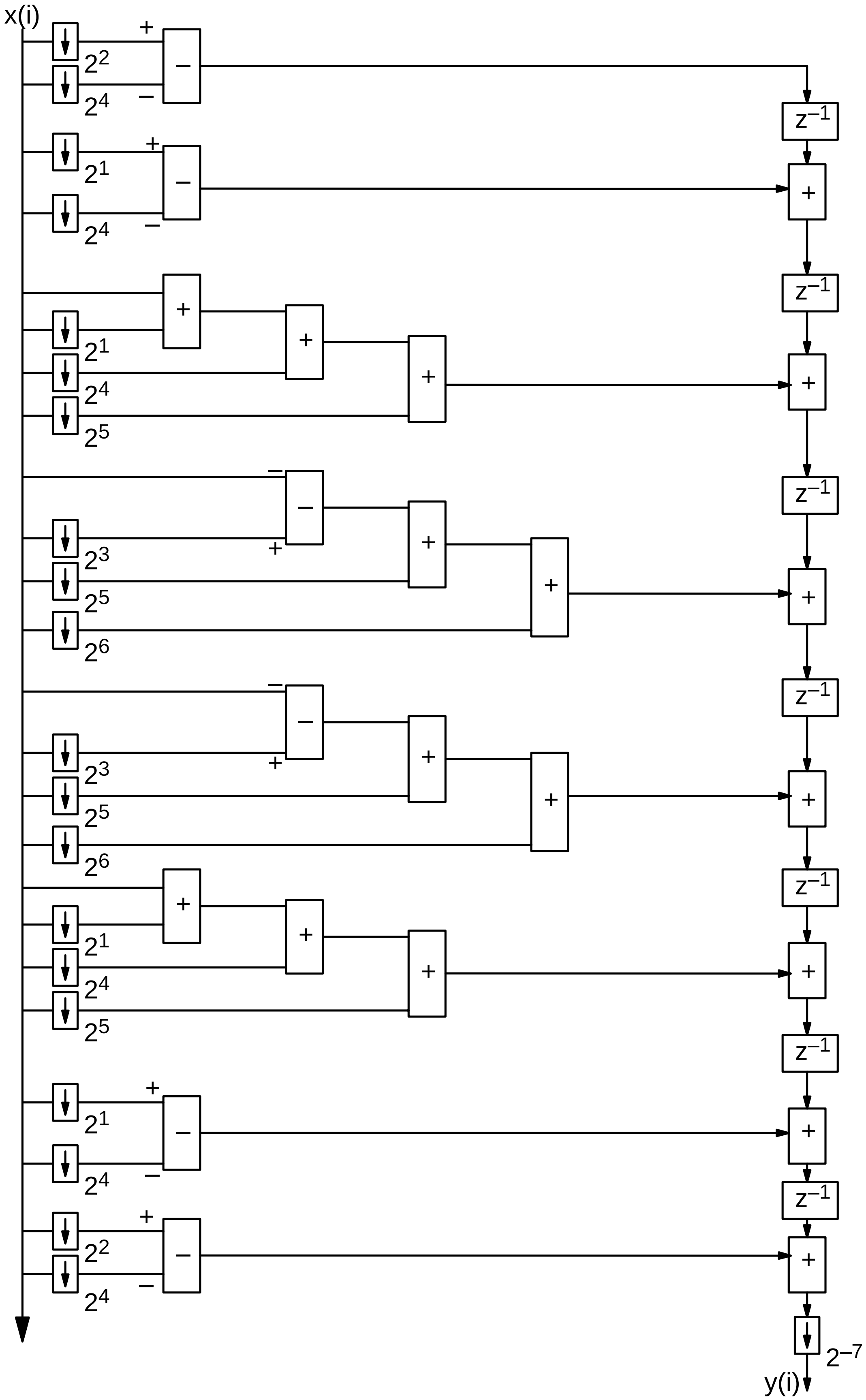


c. The following picture is showing the dependency graph for the minimum and maximum search by using of minmax–nodes. These minmax–nodes can simultaneously determine the minimum and maximum of two input operands. The number of required comparators can be reduced from 14 to 10. The number of selectors as well as the number of steps are the same like the solution in b.).



Exercise 10.3

a. The following picture shows the complete FIR filter when each coefficient is individually implemented by hardwired shifts and add/subtract. The number of adders/subtractors needed for the complete FIR filter is 23.



x(i)
2^2
2^4
2^1
2^4
2^1
2^4
2^5
2^3
2^5
2^6
2^3
2^5
2^6
2^1
2^4
2^5
2^1
2^4
2^2
2^4
z^{-1}
2^{-7}
y(i)

b. The following picture shows the complete FIR filter after factorization (see Figure 10.1.3) and utilization of the symmetric impulse response (see Figure 6.2.13). The number of adders/subtractors needed for the complete FIR filter can be reduced from 23 to 11.