Logische Operationen, Logikschaltungen — Uebung

① a) NOT(A NAND B) =

b) NOT(A XOR B) =

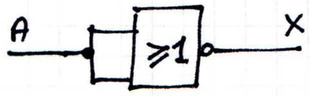
c) (NOT A) AND (NOT B) = NOT(...)

d) (NOT A) OR (NOT B) = NOT(...)

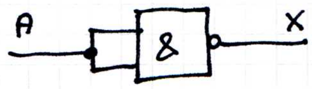
e) (A AND NOT(B)) OR (NOT(A) AND B) =

f) (A AND B) OR (NOT(A) AND NOT(B)) =

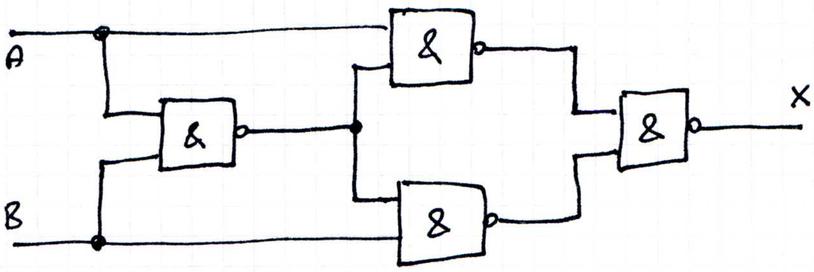
② a)



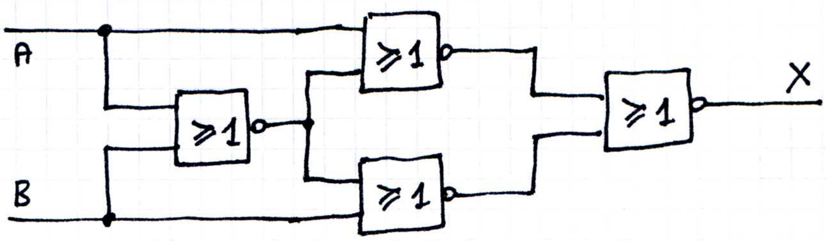
b)



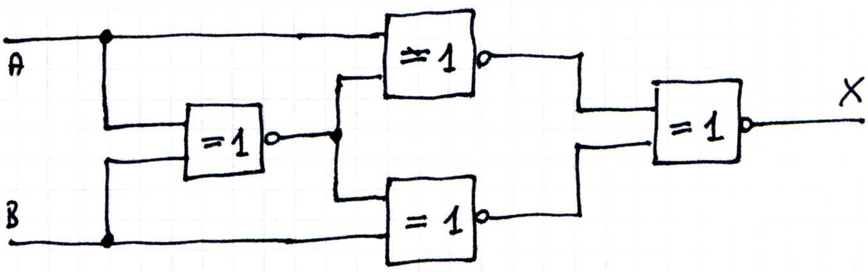
③



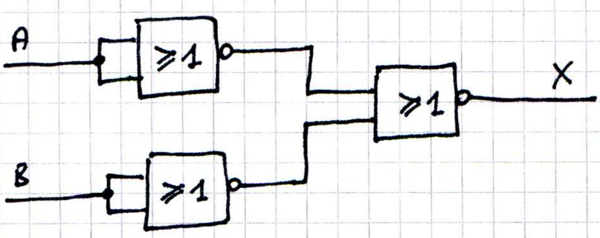
④



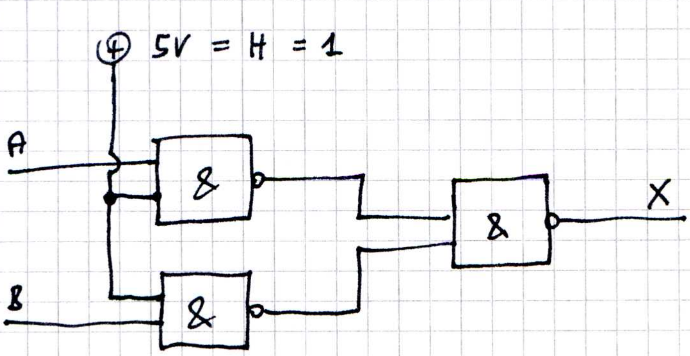
⑤



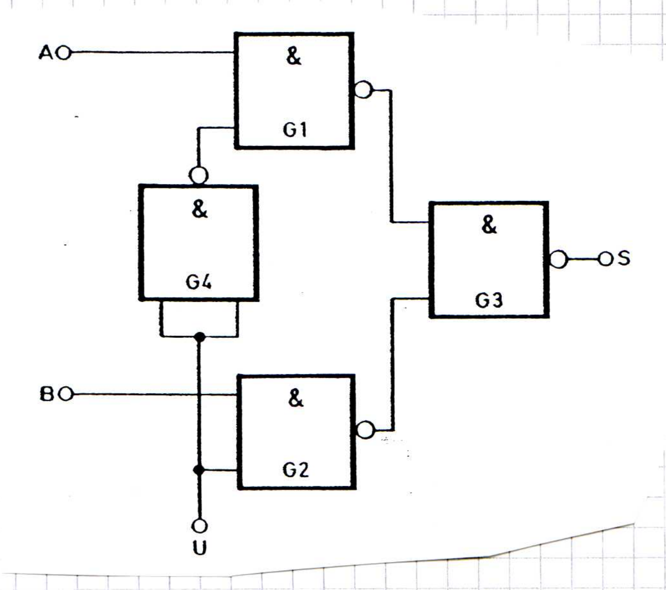
6



7



8

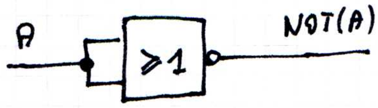


Der Eingang U ist unten gezeichnet statt links.
Er hat eine ganz andere Funktion als A und B.
Was leistet die Schaltung?

(B5) Eigentlich würde NOR genügen ...

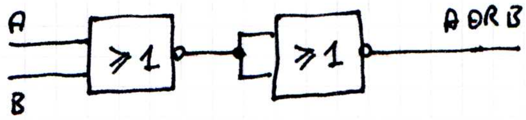
Tatsächlich kann man alle unsere Logik-Glieder aus NOR-Bausteinen aufbauen! (Es geht auch mit NAND!)

- NOT(A)

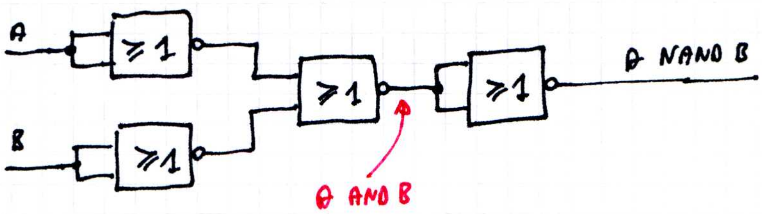


Uebung 2a

- A OR B = NOT(A NOR B)



- A EQUI B siehe Uebung 4 !
- NOT(A EQUI B) = A XOR B also Uebung 4 mit zusätzlichem Inverter
- A AND B siehe Uebung 6 !
- A NAND B somit aus Uebung 6 + Inverter :

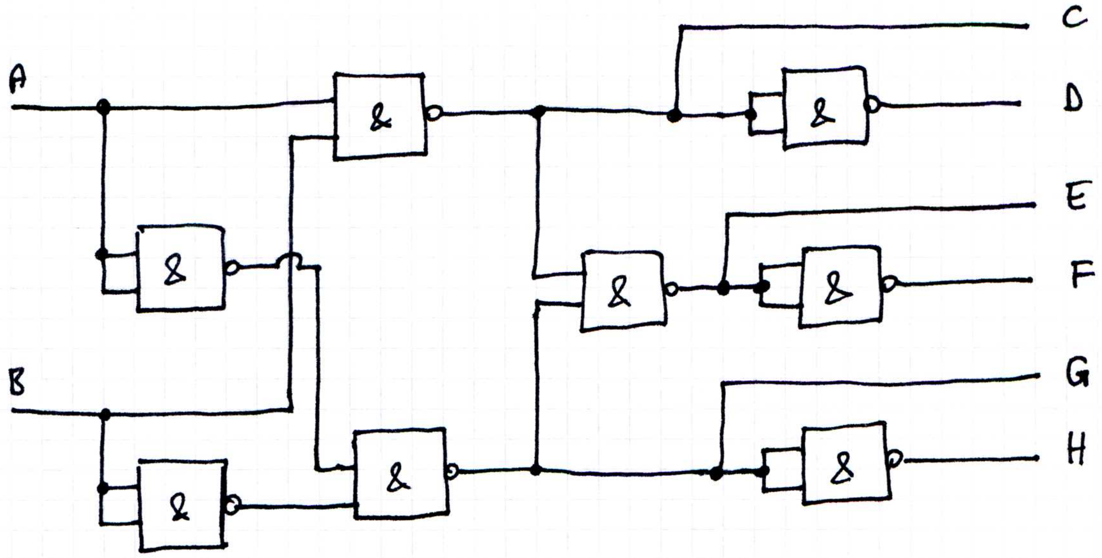


Genauso kann man zeigen, dass man auch aus NAND-Gliedern alle anderen Logikfunktionen zusammensetzen kann.
Davon ~~noch~~ die folgende Übungsaufgabe :

((wiki Shefferscher Strich ~ NAND
nach Henry Maurice Sheffer, 1882 - 1964, US-Logiker,
geboren als polnischer Jude in der Ukraine,
emigriert (mit Eltern) 1892
1916 - 1952 Prof in Harvard))

wiki Charles Sanders Peirce
schon um Charles Sanders ~~Pierce~~ ~~Pierce~~ Peirce entdeckte das schon viel früher, hat es aber nicht publiziert
(1839 - 1914)

Uebung zu NAND



Welche Funktionen C bis H stehen am Ausgang zur Verfügung?

→ wiki peirce !!

Charles Sanders Peirce, 1839 - 1914