

5. Übungsblatt - zu bearbeiten bis 27.11.2023

Aufgabe 1 Programmierbare Logikarrays

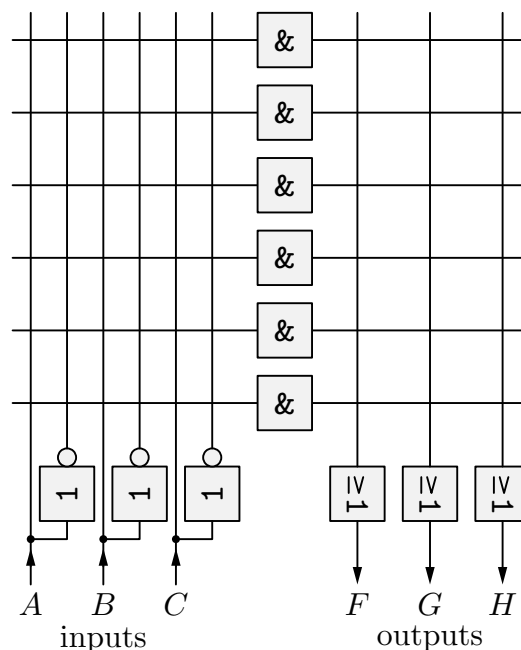
- a) Realisieren Sie die folgenden Funktionen in einem programmierbaren Logikarray (PLA, siehe rechts). Markieren Sie die Kreuzungspunkte in den beiden Matrizen (AND-Gitter und OR-Gitter), die verbunden werden müssen! Verwenden Sie dazu das rechts gezeigte Schema!

i) $F = \overline{A}C \vee AB$

ii) $G = \overline{A}C \vee \overline{B}C \vee A$

iii) $H = A \vee BC \vee \overline{C}$

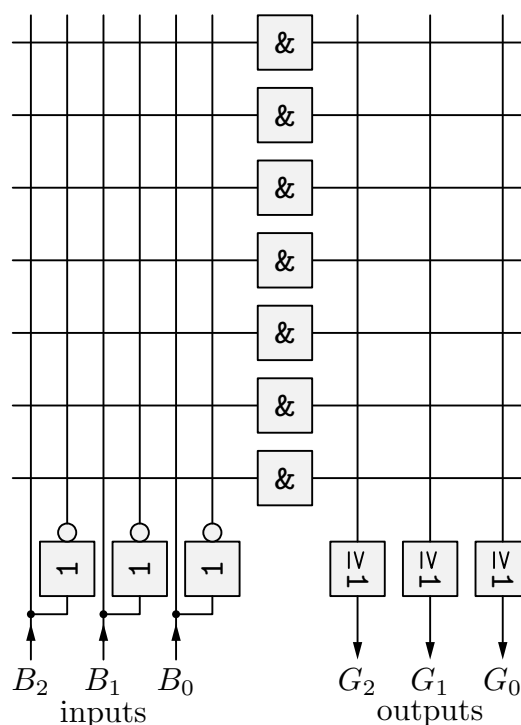
Warum reichen die im rechten Schema vorgesehenen AND-Gatter aus?



- b) Gegeben sei die folgende Übersetzungstabelle eines dreistelligen Binärkodes $[B_2 B_1 B_0]$ in einen dreistelligen Gray-Code $[G_2 G_1 G_0]$ (vergleiche Aufgabe 1 auf dem 4. Übungsblatt):

Binärkode			Gray-Kode		
B_2	B_1	B_0	G_2	G_1	G_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

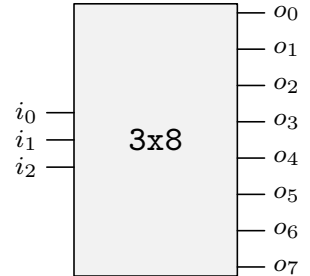
Implementieren Sie die Berechnung des Gray-Kodes (alle drei Spalten) mit Hilfe eines programmierbaren Logikarrays! Verwenden Sie dazu das rechts gezeigte Schema! Warum reichen die im rechten Schema vorgesehenen AND-Gatter aus?



Aufgabe 2 Hardwareschaltung: Dekodierer

Ein 3×8 -Dekodierer hat drei Eingänge i_2, i_1, i_0 und acht Ausgänge o_7, \dots, o_0 . Wenn die Eingabe $(i_2 i_1 i_0)_2 = x$ anliegt, wird der Ausgang o_x auf 1 und alle anderen Ausgänge auf 0 geschaltet. Der Dekodierer ist rechts als Schaltzeichen abgebildet.

- a) Gegeben sie einen boolschen Ausdruck für o_3 an.
- b) Skizzieren Sie eine Schaltung, die eine vierstellige vorzeichenlose Binärzahl als Eingabe erhält und 1 ausgibt, wenn die Zahl ein Vielfaches von 7 ist, ansonsten 0. Verwenden Sie hierzu nur
- einen 3×8 -Dekodierer
 - ein NOR-Gatter mit drei Eingängen
 $\downarrow (a, b, c) = \neg(a \vee b \vee c)$
 - ein NOR-Gatter mit zwei Eingängen
 - ein XOR-Gatter mit zwei Eingängen.



Aufgabe 3 Hardware Description Language (HDL)

Alle zur Arbeit mit HDL nötigen Informationen finden Sie auf

<http://www.nand2tetris.org/software.php>

Lesen Sie insbesondere das Tutorial zum Hardware-Simulator (auf der oben angegebenen Webseite verfügbar) und prüfen Sie Ihre Lösungen der folgenden Aufgaben mit diesem Hardware-Simulator!

- a) Implementieren Sie die primitiven Gatter NOT, AND und OR lediglich mit Hilfe von NAND-Gattern in HDL!
- b) Implementieren Sie ein NOR-Gatter und einen 2-Multiplexer mit Hilfe von NAND-Gattern in HDL!
- c) Schreiben Sie Tests, um Ihre Gatter auf Korrektheit zu überprüfen!
- d) Implementieren Sie die Funktion $(a \wedge \bar{b}) \vee (b \wedge c) \vee (\bar{a} \wedge \bar{b})$!