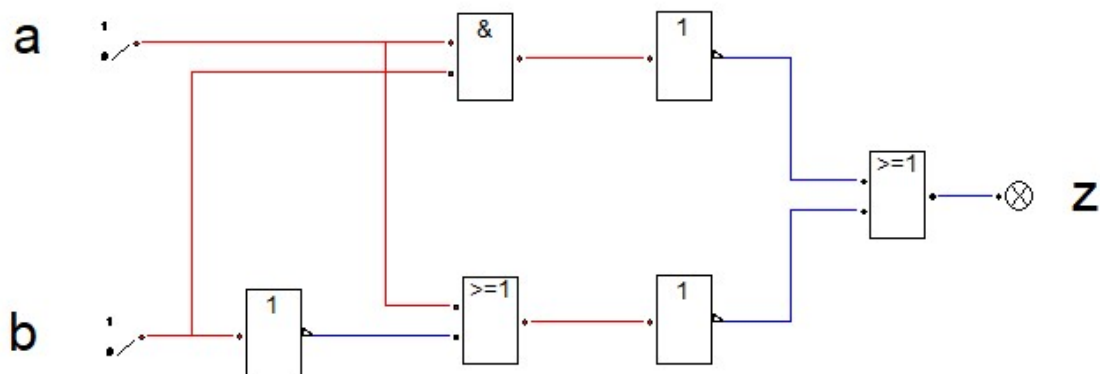


1. **MergeSort:** Eine Liste **a** mit den **n** Komponenten **a[0]**, , **a[n-1]**, für die die Ordnungsrelationen **<**, **>**, **≤**, **≥** erklärt sind, ist aufsteigend zu sortieren, so daß gilt: **a[0] ≤ a[1] ≤ ≤ a[n-1]**
 - a) Was versteht man unter einer rekursiv formulierten Funktion?
 - b) Formuliere die Schritte, gemäß denen der Algorithmus MergeSort vorgeht, in Worten.
An welcher Stelle zeigt sich die rekursive Definition des Algorithmus?
Welche Vereinbarung stellt sicher, daß der Algorithmus terminiert?
 - c) Schreibe den Quellcode der Funktion **sort** als Python-Programm; erläutere auch, was die Funktion **sort(a, l, r)** und die als gegeben vorausgesetzte Funktion **merge(a, l, m, r)** bewirken.
Wie lautet der Aufruf, um die Liste **a** mit **n** Datenelementen zu sortieren?
 - d) Führe den Sortieralgorithmus MergeSort auf dem Arbeitsblatt exemplarisch durch; ergänze auch die jeweiligen Parameter der Funktionen **sort** und **merge**.
Verdeutliche die durch **sort** jeweils veranlaßten Funktionsaufrufe durch Pfeile.
2. a) Für Boolesche Variable gilt bekanntlich das Distributivgesetz: **a (b + c) = a b + a c**
Formuliere das zur vorstehenden Rechenregel gehörende duale Distributivgesetz und beweise die Gültigkeit dieses dualen Gesetzes.
b) Vereinfache: $\bar{a} \cdot b + a + \bar{b}$

3. Die Boolesche Funktion **z = f(a,b,c)** ist durch folgende Wahrheitstafel gegeben:

a	b	c	z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

- a) Ermittle die disjunktive Normalform (DNF) für **z**.
 - b) Vereinfache den Funktionsterm unter Anwendung der Rechenregeln für Boolesche Terme.
 - c) Zeichne den Schaltplan für die optimierte Funktion **z**.
4. Gegeben ist folgende digitale Schaltung:



- a) Notiere am Ausgang jedes Gatters den jeweils entstandenen Booleschen Term.
- b) Optimierte den für z erhaltenen Term.
- c) Zeichne die digitale Schaltung für den optimierten Term.

sort(0,7)							
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
3	7	9	5	6	4	2	7

sort(,)				sort(,)			
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]

sort(,)		sort(,)		sort(,)		sort(,)	
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]

sort(,)	sort(,)	sort(,)	sort(,)	sort(,)	sort(,)	sort(,)	sort(,)
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
merge(, ,)		merge(, ,)		merge(, ,)		merge(, ,)	

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
merge(, ,)				merge(, ,)			

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
merge(, ,)							

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]