

Multiplexer und Demultiplexer (MUX)

Vorlesung Digitales Design

Hes·so  **VALAIS
WALLIS**



Haute Ecole d'Ingénierie
Hochschule für Ingenieurwissenschaften

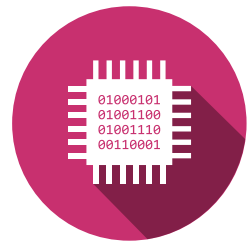
Orientierung: [Informatik und Kommunikationssysteme \(ISC\)](#)

Kurs: Digitales Design (DiD)

Verfasser: [Christophe Bianchi](#), [François Corthay](#), [Pierre Pompili](#), [Silvan Zahno](#)

Datum: 25. August 2022

Version: v2.1



Inhaltsverzeichnis

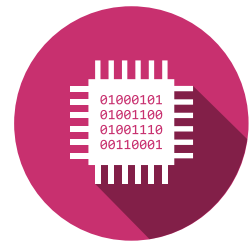
1 Einführung	2
2 Multiplexer	3
2.1 Universelle Logikfunktion	3
2.2 Realisierung von Multiplexern mittels Basislogikgatter	4
2.3 Realisierung von Multiplexern mit einer Baumstruktur	5
2.4 Vereinfachung von Funktionen mit Multiplexern als Grundlage	5
3 Demultiplexer	8
3.1 Realisierung von Demultiplexern mittels Basislogikgatter	9
3.2 Realisierung von Demultiplexern mit einer Baumstruktur	9
Literatur	11



1 Einführung

Ein Multiplexer übergibt die Information von seinen Eingängen an den einzigen Ausgang. Der **Multiplexer** ermöglicht unter anderem die Realisierung einer beliebigen Logikfunktion. Der **Demultiplexer** seinerseits übergibt die Information von seinem Eingang an einen der Ausgänge.

In diesem Kapitel befassen wir uns mit diesen beiden logischen Operatoren, ihrem inneren Schaltkreis und mit der Möglichkeit, Schaltkreise mit diesen Operatoren als Basis zu vereinfachen.



2 Multiplexer

Der Multiplexer überträgt einen seiner 2^n Dateneingänge auf seinen Ausgang und zwar gemäss der Kombination seiner n Auswahlwege.

Der Multiplexer arbeitet wie eine Weiche.

Beispiel: 2 zu 1 Multiplexer

Der 2 zu 1 Multiplexer in der Abbildung 1 übergibt die Information von einem seiner zwei Dateneingänge, x_0 und x_1 , an seinen Ausgang y , gemäss dem Auswahlsignal sel .



Abbildung 1: 2 zu 1 Multiplexer

Beispiel: 4 zu 1 Multiplexer

Der 4 zu 1 Multiplexer in der Abbildung 2 übergibt die Information von einem seiner 4 Dateneingänge, x_0 bis x_3 , an seinen Ausgang y , gemäss den beiden Auswahlbits, s_1 und s_0 .

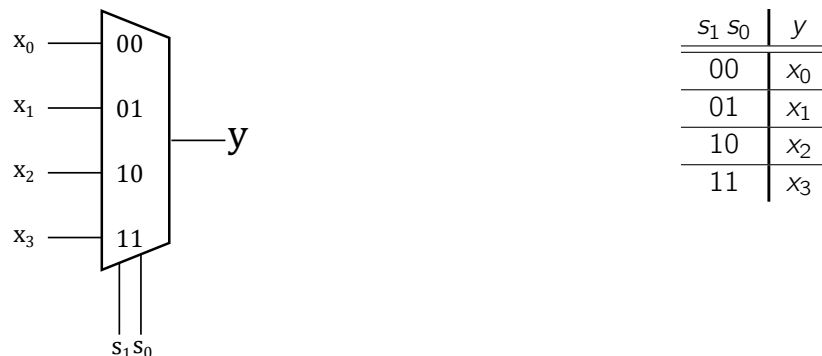


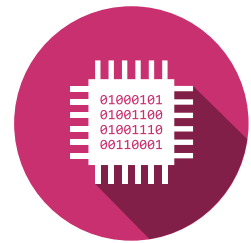
Abbildung 2: 4 zu 1 Multiplexer

2.1 Universelle Logikfunktion

Der Multiplexer kann für die Realisierung jeder beliebigen Funktion benutzt werden.

Es genügt hierfür, die Eingänge der Funktion mit der Auswahlleitung des Multiplexers zu verbinden und die Dateneingänge je nach Wert der Funktion für die angewählten Eingänge auf '0' oder '1' zu setzen.

Der Multiplexer ermöglicht so die elektronische Realisierung einer Wahrheitstabelle.



Beispiel: Funktion mit 2 Eingängen

Die Abbildung 3 zeigt, wie man die Exklusiv-ODER-Funktion von 2 Eingängen mit Hilfe eines Multiplexer mit 2 Auswahleingängen und somit $2^2 = 4$ Dateneingängen realisiert.

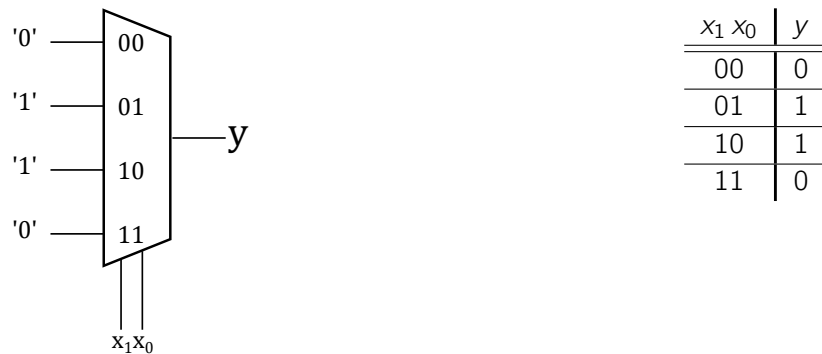


Abbildung 3: Exklusiv-ODER-Funktion mit 2 Eingängen

2.2 Realisierung von Multiplexern mittels Basislogikgatter

Ein 2^n zu 1 Multiplexer wird mit 2^n UND-Gattern realisiert, die alle einen der Dateneingänge und alle Auswahleingänge, invertiert oder nicht, erhalten. Die invertierten oder nichtinvertierten Auswahleingänge werden so verteilt, dass nur eines der ODER-Gatter seinen Ausgang nicht unbedingt auf '0' gesetzt hat. Die Ausgänge der UND-Gatter werden zu einem ODER-Gatter geführt, das den Ausgang des Multiplexers realisiert.

Beispiel: 4 zu 1 Multiplexer

Für einen 4 zu 1 Multiplexer benutzt man UND- und ODER-Gatter sowie Inverter, wie in der Abbildung 4 dargestellt.

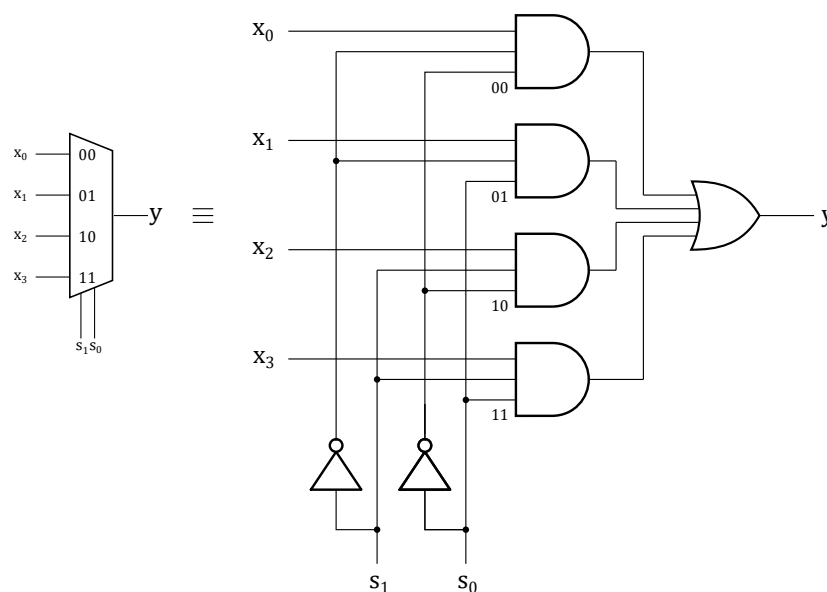


Abbildung 4: Realisierung eines 4 zu 1 Multiplexers mittels Basislogikgatter



2.3 Realisierung von Multiplexern mit einer Baumstruktur

Ein 2^n zu 1 Multiplexer wird mit einer Baumstruktur von n Multiplexerstufen 2 zu 1 realisiert. Die Multiplexer jeder Stufe werden mit einer der Auswahlvariablen kontrolliert.

Beispiel: 4 zu 1 Multiplexer

Der 4 zu 1 Multiplexer wird mittels Multiplexern 2 zu 1 realisiert, wie in der Abbildung 5 dargestellt.

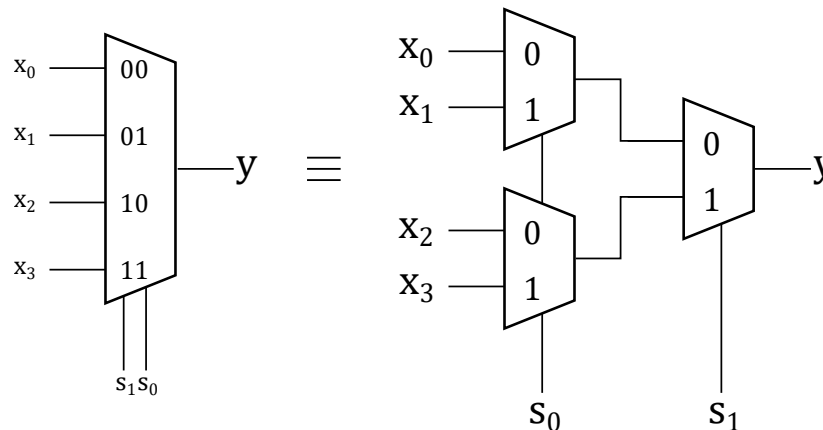


Abbildung 5: Realisierung eines 4 zu 1 Multiplexers mit einer Baumstruktur

2.4 Vereinfachung von Funktionen mit Multiplexern als Grundlage

Wenn der Multiplexer als Funktionsgenerator benutzt wird, kann die Grösse des Schaltkreises immer verringert werden, wenn man von der Darstellung des Multiplexers in einer Baumstruktur ausgeht. Diese Vereinfachung basiert auf folgenden Überlegungen:

- Ein 2 zu 1 Multiplexer mit zwei identischen Eingängen kann eliminiert werden und sein Ausgang mit einem der Eingänge verbunden werden.
- Ein 2 zu 1 Multiplexer mit den Eingängen $x_0 = 0$ und $x_1 = 1$ kann eliminiert werden und sein Ausgang mit dem Auswahleingang verbunden werden.
- Ein 2 zu 1 Multiplexer mit den Eingängen $x_0 = 1$ und $x_1 = 0$ kann eliminiert werden und sein Ausgang mit der Umkehrung des Auswahleingangs verbunden werden.

Mit dieser Vereinfachung kann die Tiefe der Baumstruktur um eine Einheit verringert werden.

Die Vereinfachung kann auf den nachfolgenden Stufen durchgeführt werden und dank der Beobachtung des resultierenden Schaltkreises können alle Multiplexer mit identischen Eingängen aufgezählt werden, um schliesslich nur einen zu berücksichtigen.

Beispiel: Verringerung um ein Niveau

In der Abbildung 6 ist die Realisierung der Mehrheitsfunktion mit 3 Eingängen mit Hilfe eines Multiplexers mit 3 Auswahleingängen und somit $2^3 = 8$ Dateneingängen dargestellt.

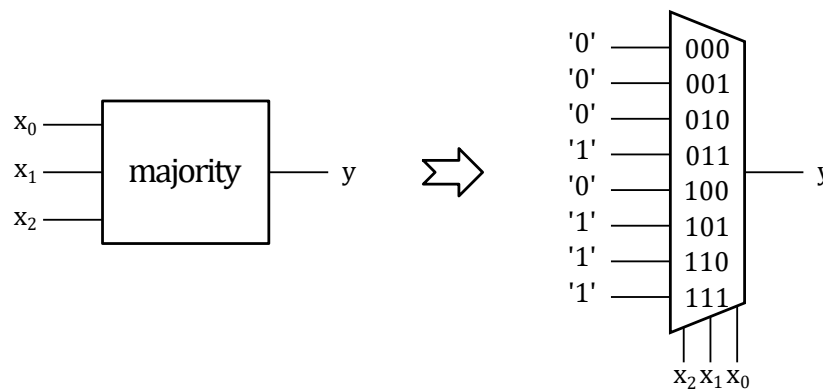
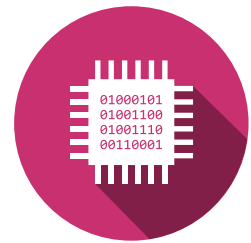


Abbildung 6: Realisierung der Mehrheitsfunktion mittels eines Multiplexers

In der Abbildung 7 ist die Vereinfachung dieses Schaltkreises auf einen 4 zu 1 Multiplexer dargestellt.

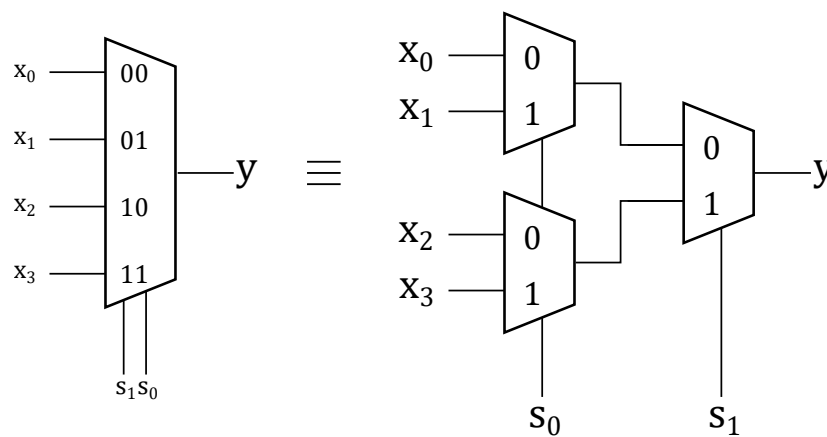


Abbildung 7: Vereinfachung eines 8 zu 1 Multiplexers auf einen 4 zu 1 Multiplexer

Beispiel: Vereinfachung um mehrere Niveaus

In der Abbildung 8 ist die Vereinfachung eines 16 zu 1 Multiplexers mit konstanten Eingängen mit Hilfe von 2 zu 1 Multiplexern dargestellt.

Die Eingänge von zwei Multiplexerpaaren sind identisch; für diese Gruppen wurde nur ein einziger Multiplexer behalten. Es sei darauf hingewiesen, dass der Multiplexer für die Umkehrung des Eingangs x_0 beibehalten wurde.

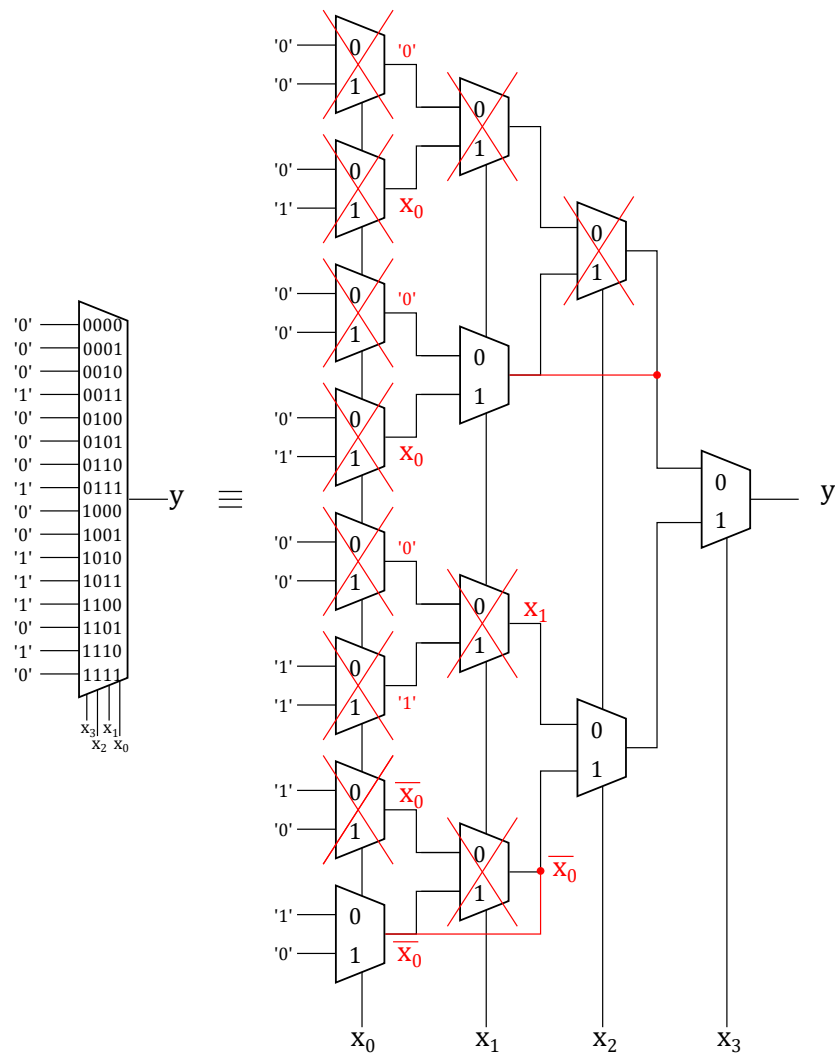
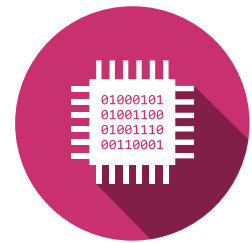
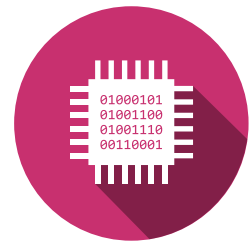


Abbildung 8: Vereinfachung eines 16 zu 1 Multiplexers mit 2 zu 1 Multiplexern



3 Demultiplexer

Der Demultiplexer übergibt die Information von seinem Dateneingang an einen seiner 2^n Ausgänge und zwar gemäss der Kombination seiner n Auswahlwege. Die nicht ausgewählten Ausgänge werden auf '0' gesetzt.

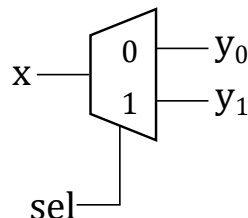
Der Demultiplexer arbeitet wie eine Weiche.



Es wird zu oft vergessen, dass die nicht angewählten Eingänge auf '0' gesetzt werden.

Beispiel: 1 zu 2 Demultiplexer

Der 1 zu 2 Demultiplexer der Abbildung 9 übergibt die Information seines Dateneingangs, x , an einen seiner beiden Ausgänge, y_0 oder y_1 , gemäss dem Auswahlsignal sel . Der andere Ausgang wird auf '0' gesetzt.

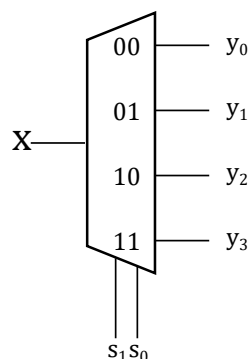


sel	y_1	y_0
0	0	x
1	x	0

Abbildung 9: 1 zu 2 Demultiplexer

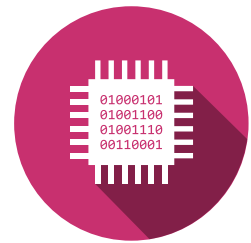
Beispiel: Demultiplexer 1 zu 4

Der 1 zu 4 Demultiplexer der Abbildung 10 übergibt die Information seines Dateneingangs, x , an einen seiner 4 Ausgänge, y_0 bis y_3 , gemäss den beiden Auswahlbit, s_1 und s_0 . Die anderen Ausgänge werden auf '0' gesetzt.



$s_1 s_0$	y_3	y_2	y_1	y_0
00	0	0	0	x
01	0	0	x	0
10	0	x	0	0
11	x	0	0	0

Abbildung 10: 1 zu 4 Demultiplexer



3.1 Realisierung von Demultiplexern mittels Basislogikgatter

Ein 1 zu 2^n Demultiplexer wird mit 2^n UND-Gattern realisiert, die alle einen der Dateneingänge sowie alle Auswahlwege, invertiert oder nicht, erhalten, analog zu dem, was wir bereits für den Multiplexer gesehen haben.

Beispiel: 1 zu 4 Demultiplexer

Der 1 zu 4 Demultiplexer wird mit UND-Gattern und Invertern realisiert, wie in der Abbildung 11 dargestellt.

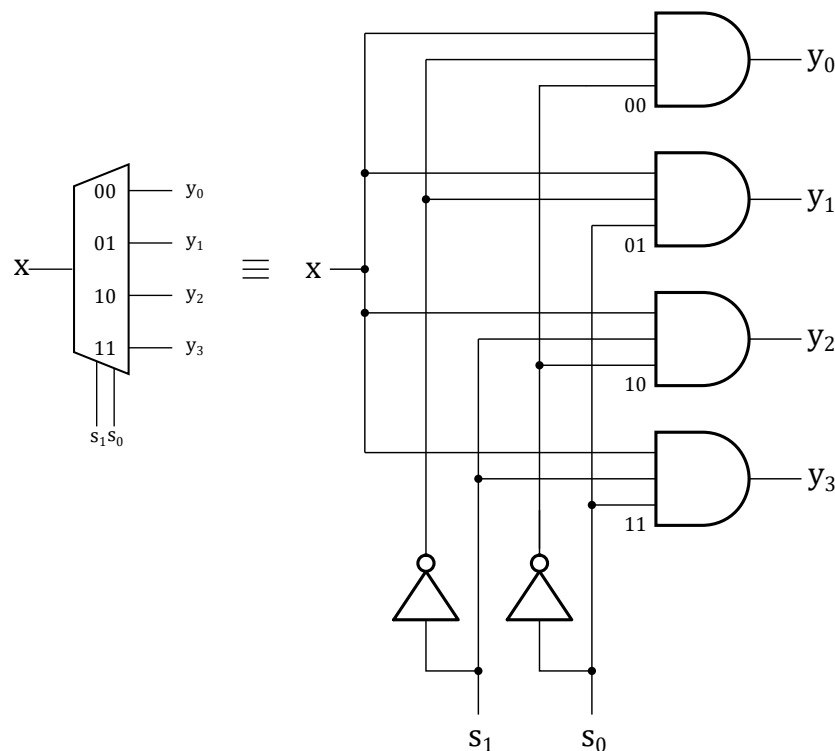


Abbildung 11: Realisierung eines Demultiplexers mittels Basislogikgatter

3.2 Realisierung von Demultiplexern mit einer Baumstruktur

Ein 1 zu 2^n Demultiplexer wird mit einer Baumstruktur von n Demultiplexerstufen 2 zu 1 realisiert, ähnlich wie beim Multiplexer.

Beispiel: 1 zu 4 Demultiplexer

Der 1 zu 4 Demultiplexer wird mit Hilfe von Demultiplexern 2 zu 1 realisiert, wie in der Abbildung 12 dargestellt.

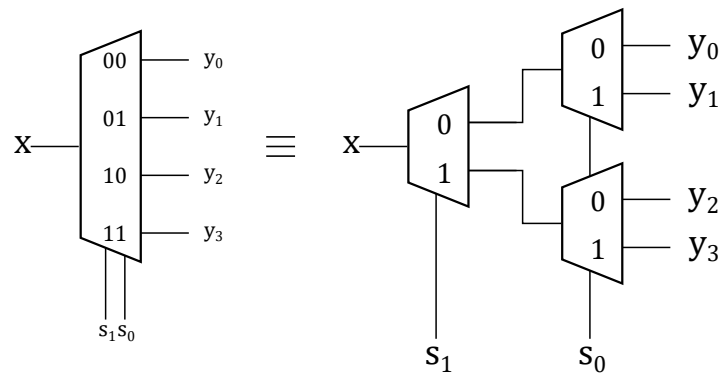
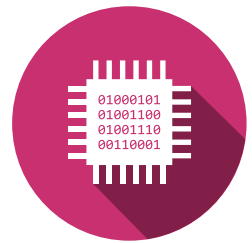


Abbildung 12: Realisierung eines 1 zu 4 Demultiplexers mit einer Baumstruktur



Literatur

- [1] Michael D. Ciletti und M. Morris Mano. *Digital Design*. second edition. New-Jersey: Prentice-Hall, 2007.
- [2] Marcel Gindre und Denis Roux. *Electronique Numérique, Logique Combinatoire et Technologie*. Paris: McGraw-Hill, 1987.
- [3] Daniel Mange. *Analyse et synthèse des systèmes logiques*. Editions Géorgi. Bd. Traité d'électricité, volume V. St Saphorin: PPUR presses polytechniques, 1995. 362 S. ISBN: 978-2-88074-045-0. Google Books: [5NSdD4GRl3cC](#).
- [4] John F. Wakerly. *Digital Design: Principles And Practices*. 3rd edition. Prentice-Hall, 2008. ISBN: 0-13-082599-9.