

Elektronikpraktikum Auswertung: Versuchstag 7

Digitalelektronik

Gruppe 01
Patrick Heuer
Benjamin Lotter

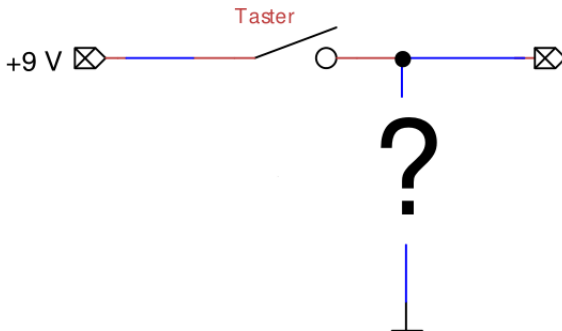
Übersicht

- 1 Logikgatter
 - Digitale Ein- und Ausgabe: Hilfsschaltung
 - CMOS-Schaltung
 - NAND-Gatter mit integriertem Schaltkreis (IC)
 - Exklusiv-Oder-Gatter (XOR) aufgebaut aus NAND-Gattern
- 2 Addierer
 - Halbaddierer
 - Volladdierer
- 3 Sequentielle Logik
 - RS-Flipflop
 - Taktgesteuertes RS-Flip-Flop
 - D-Latch
 - Flankengetriggertes D-Latch
- 4 Zähler
- 5 Binary code decimal und 7-Segment Anzeige

Übersicht

- 1 Logikgatter
 - Digitale Ein- und Ausgabe: Hilfsschaltung
 - CMOS-Schaltung
 - NAND-Gatter mit integriertem Schaltkreis (IC)
 - Exklusiv-Oder-Gatter (XOR) aufgebaut aus NAND-Gattern
- 2 Addierer
 - Halbaddierer
 - Volladdierer
- 3 Sequentielle Logik
 - RS-Flipflop
 - Taktgesteuertes RS-Flip-Flop
 - D-Latch
 - Flankengetriggertes D-Latch
- 4 Zähler
- 5 Binary code decimal und 7-Segment Anzeige

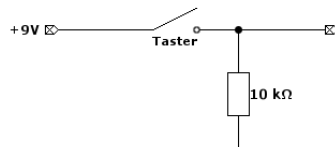
Logikpegel



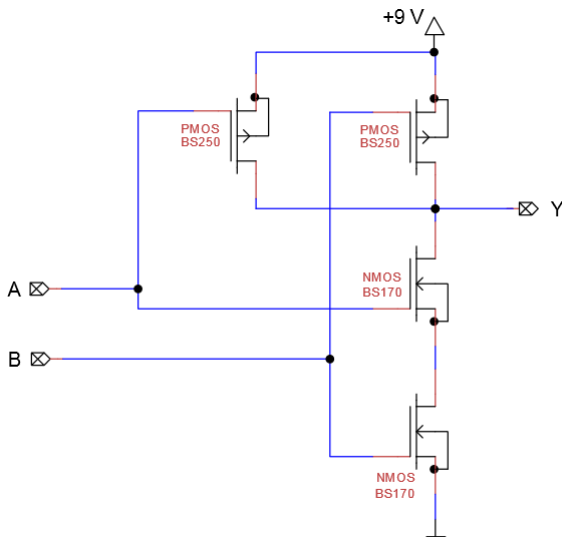
Logikpegel

Pull-Down Widerstand von $R = 10k\Omega$
liefert High und Low:

H	0.002 V
L	8.99 V



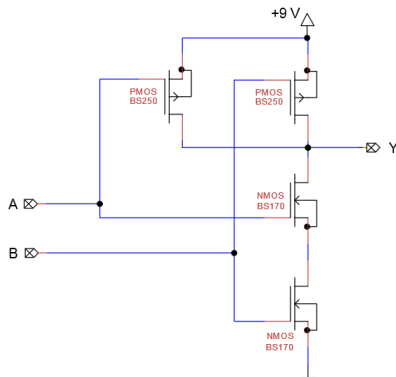
CMOS-Schaltung



CMOS-Schaltung

A	B	Y
1	0	1
0	1	1
0	0	1
1	1	0

NAND-Schaltung

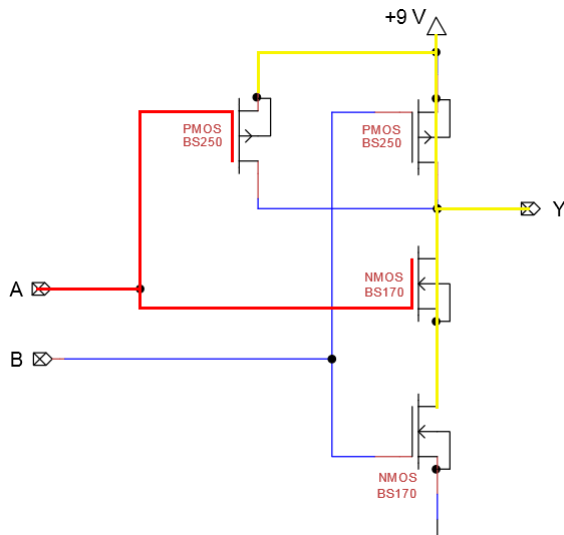


Funktionsweise

- PMOS sperren mit H an Basis
- NMOS leiten mit H an Basis

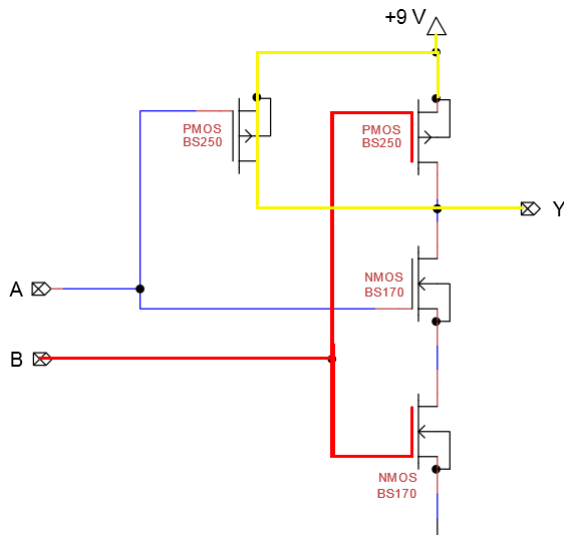
Funktionsweise

A	B	Y
1	0	1



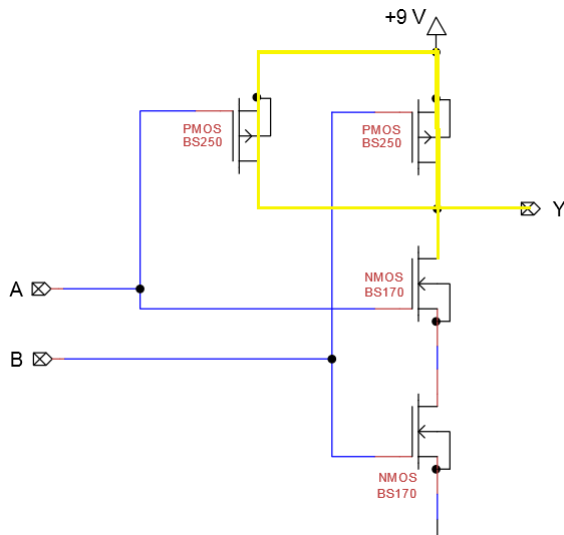
Funktionsweise

A	B	Y
1	0	1
0	1	1



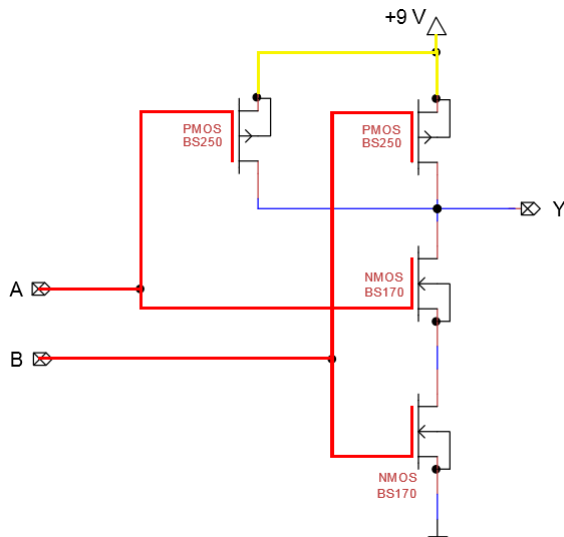
Funktionsweise

A	B	Y
1	0	1
0	1	1
0	0	1



Funktionsweise

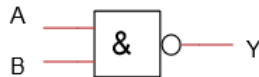
A	B	Y
1	0	1
0	1	1
0	0	1
1	1	0



NAND-Gatter mit IC

A	B	Y
1	0	1
0	1	1
0	0	1
1	1	0

	Messung	Datenblatt
H	8.54V	$> 8.5V$
L	0.0006V	$< 0.05V$



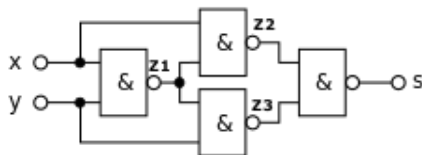
XOR aus NAND-Gattern

- NAND Schaltung:

A	B	Y
1	0	1
0	1	1
0	0	1
1	1	0

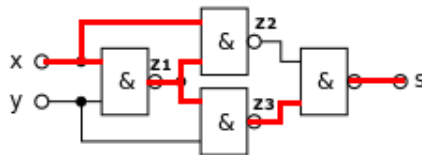
- XOR Schaltung:

A	B	Y
1	0	1
0	1	1
0	0	0
1	1	0



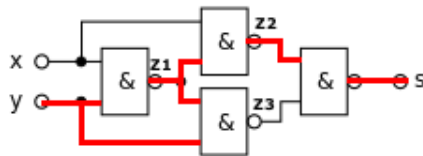
Funktionsweise

A	B	Z1	Z2	Z3	S
1	0	1	0	1	1



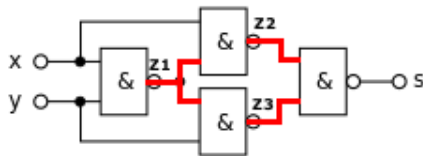
Funktionsweise

A	B	Z1	Z2	Z3	S
1	0	1	0	1	1
0	1	1	1	0	1



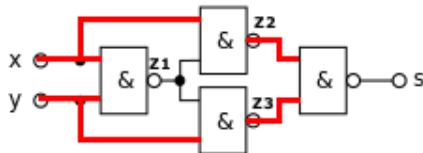
Funktionsweise

A	B	Z1	Z2	Z3	S
1	0	1	0	1	1
0	1	1	1	0	1
0	0	1	1	1	0



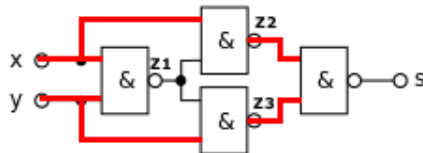
Funktionsweise

A	B	Z1	Z2	Z3	S
1	0	1	0	1	1
0	1	1	1	0	1
0	0	1	1	1	0
1	1	0	1	1	0



Funktionsweise

A	B	Z1	Z2	Z3	S
1	0	1	0	1	1
0	1	1	1	0	1
0	0	1	1	1	0
1	1	0	1	1	0



→ XOR-Schaltung

Übersicht

- 1 Logikgatter
 - Digitale Ein- und Ausgabe: Hilfsschaltung
 - CMOS-Schaltung
 - NAND-Gatter mit integriertem Schaltkreis (IC)
 - Exklusiv-Oder-Gatter (XOR) aufgebaut aus NAND-Gattern
- 2 Addierer
 - Halbaddierer
 - Volladdierer
- 3 Sequentielle Logik
 - RS-Flipflop
 - Taktgesteuertes RS-Flip-Flop
 - D-Latch
 - Flankengetriggertes D-Latch
- 4 Zähler
- 5 Binary code decimal und 7-Segment Anzeige

Halbaddierer

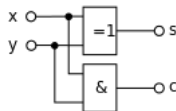
X	Y	C	S	Dezimal
0	1	0	1	1
1	0	0	1	1
0	0	0	0	0
1	1	1	0	2

- addiert binär X und Y ($X, Y \in \{0, 1\}$)
- Carry (=Übertrag) wie beim schriftlichen Addieren

$$1 + 1 = 0 \quad C = 1$$

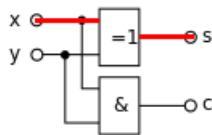
analog dezimal:

$$9 + 1 = 0 \quad C = 1$$



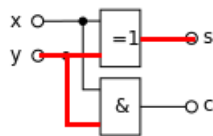
Funktionsweise

X	Y	C	S	Dezimal
1	0	0	1	1



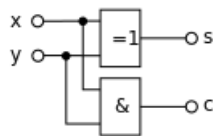
Funktionsweise

X	Y	C	S	Dezimal
1	0	0	1	1
0	1	0	1	1



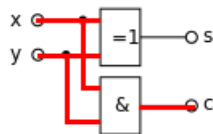
Funktionsweise

X	Y	C	S	Dezimal
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0



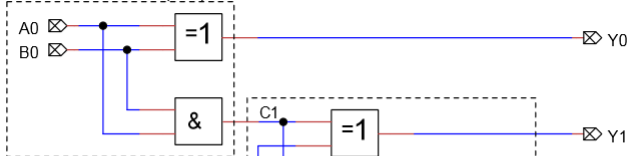
Funktionsweise

X	Y	C	S	Dezimal
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0
1	1	1	0	2

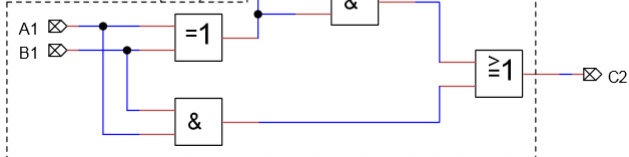


Volladdierer

Halbaddierer für bit 0 (LSB)



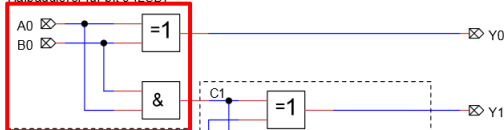
Volladdierer für bit 1 (MSB)



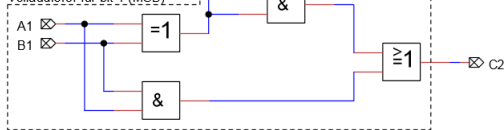
Funktionsweise

- Die ersten Bits werden addiert

Halbaddierer für bit 0 (LSB)

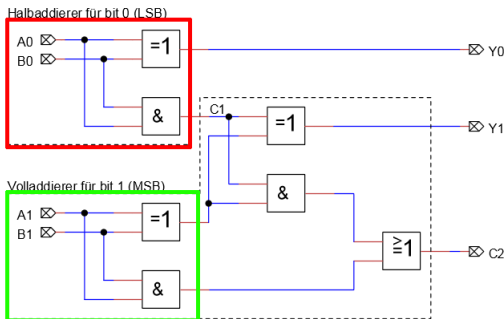


Volladdierer für bit 1 (MSB)



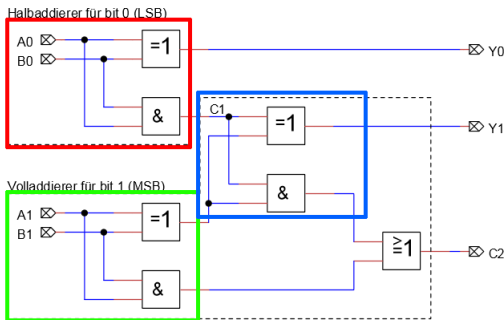
Funktionsweise

- Die ersten Bits werden addiert
- Die zweiten Bits werden addiert



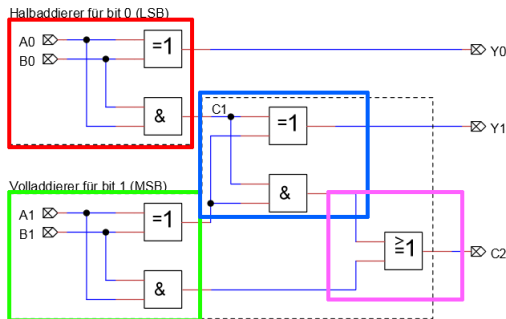
Funktionsweise

- Die ersten Bits werden addiert
- Die zweiten Bits werden addiert
- Das Carry der ersten Bits wird mit der Stelle der zweiten addiert



Funktionsweise

- Die ersten Bits werden addiert
- Die zweiten Bits werden addiert
- Das Carry der ersten Bits wird mit der Stelle der zweiten addiert
- entsteht bei den letzten beiden Schritten ein Übertrag auf dem zweiten Bit wird dieser auf die dritte Stelle geschoben



A0	B0	C1	A1	B1	Y0	Y1	C2	=
0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0	1
1	0	0	0	0	0	1	0	1
1	1	1	0	0	0	1	0	2
0	0	0	0	1	0	1	0	2
0	1	0	0	1	1	1	0	3
1	0	0	0	1	1	1	0	3
1	1	1	0	1	0	0	1	4
0	0	0	1	0	0	1	0	2
0	1	0	1	0	1	1	0	3
1	0	0	1	0	1	1	0	3
1	1	1	1	0	0	0	1	4
0	0	0	1	1	0	0	1	4
0	1	0	1	1	1	0	1	5
1	0	0	1	1	1	0	1	5
1	1	1	1	1	0	1	1	6

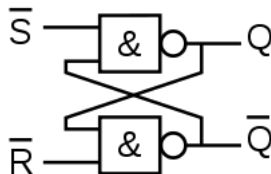
Bemerkungen

- Volladdierer nimmt zwei 2-Bit Zahlen (dezimal 0-3) und gibt 3-bit Zahl zurück (dezimal 0-6)
- C_2 OR kann durch XOR ersetzt werden, da niemals beide Eingänge gleichzeitig angesteuert werden
- Addition von n -bit Zahlen: n -tes Bit ist immer eine 1, d.h es entsteht immer ein Carry $\rightarrow n + 1$ -bit Zahl

Übersicht

- 1 Logikgatter
 - Digitale Ein- und Ausgabe: Hilfsschaltung
 - CMOS-Schaltung
 - NAND-Gatter mit integriertem Schaltkreis (IC)
 - Exklusiv-Oder-Gatter (XOR) aufgebaut aus NAND-Gattern
- 2 Addierer
 - Halbaddierer
 - Volladdierer
- 3 Sequentielle Logik
 - RS-Flipflop
 - Taktgesteuertes RS-Flip-Flop
 - D-Latch
 - Flankengetriggertes D-Latch
- 4 Zähler
- 5 Binary code decimal und 7-Segment Anzeige

RS-Flipflop

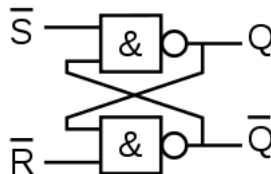


Funktionsweise

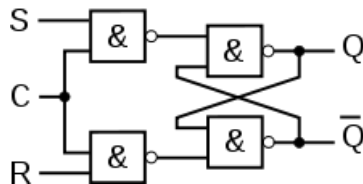
\bar{S}	\bar{R}	Q
1	1	unverändert
0	1	1 (gesetzt)
1	0	0 (zurückgesetzt)
0	0	Glitch ($Q = \bar{Q}$)

Gleizeitiges Auslößen

• ????

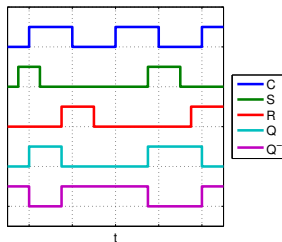
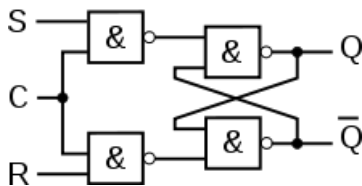


Taktgesteuertes RS-Flip-Flop

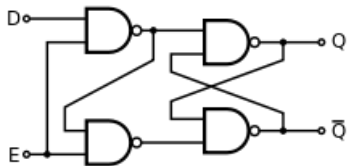


Funktionsweise

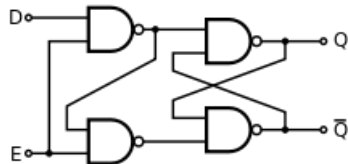
S	R	C	Q
0	0	0	unverändert
0	1	0	unverändert
1	0	0	unverändert
1	1	0	unverändert
0	0	1	unverändert
0	1	1	0 (zurückgesetzt)
1	0	1	1 (gesetzt)
1	1	1	Glitch ($\bar{Q} = Q$)



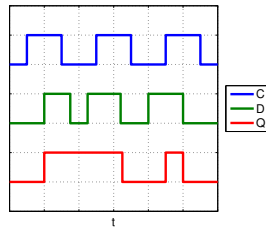
D-Latch



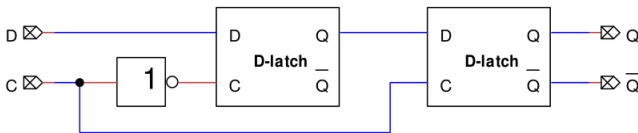
Funktionsweise



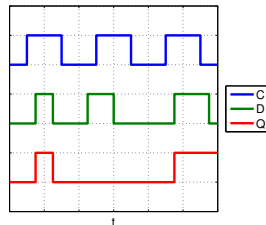
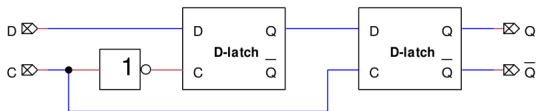
bla



Flankengetriggertes D-Latch



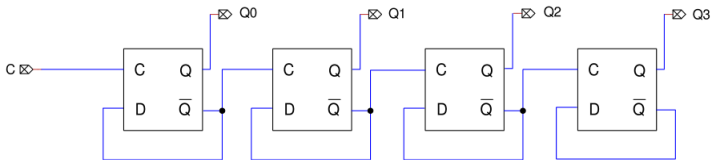
Flankengetriggertes D-Latch



Übersicht

- 1 Logikgatter
 - Digitale Ein- und Ausgabe: Hilfsschaltung
 - CMOS-Schaltung
 - NAND-Gatter mit integriertem Schaltkreis (IC)
 - Exklusiv-Oder-Gatter (XOR) aufgebaut aus NAND-Gattern
- 2 Addierer
 - Halbaddierer
 - Volladdierer
- 3 Sequentielle Logik
 - RS-Flipflop
 - Taktgesteuertes RS-Flip-Flop
 - D-Latch
 - Flankengetriggertes D-Latch
- 4 Zähler
- 5 Binary code decimal und 7-Segment Anzeige

Zähler



Entwicklung

Funktionsweise

- D-Latch flipt aktuelles Bit
- falls Q gesetzt wird \bar{Q} und somit der nächste D-Latch gesetzt
- Zahl muss 'rückwärts' gelesen werden

0000

1000

0100

1100

0010

1010

1110

0001

1001

0101

1101

0011

1011

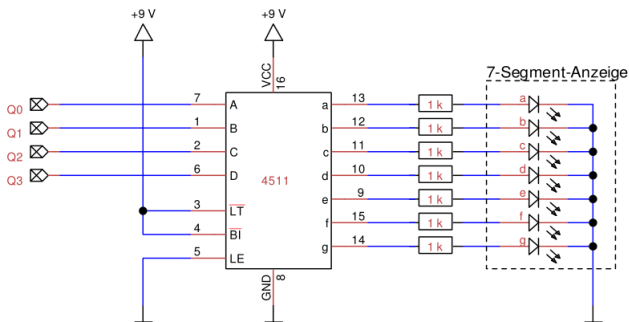
0111

1111

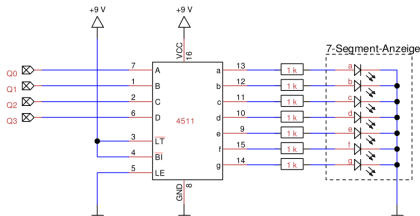
Übersicht

- 1 Logikgatter
 - Digitale Ein- und Ausgabe: Hilfsschaltung
 - CMOS-Schaltung
 - NAND-Gatter mit integriertem Schaltkreis (IC)
 - Exklusiv-Oder-Gatter (XOR) aufgebaut aus NAND-Gattern
- 2 Addierer
 - Halbaddierer
 - Volladdierer
- 3 Sequentielle Logik
 - RS-Flipflop
 - Taktgesteuertes RS-Flip-Flop
 - D-Latch
 - Flankengetriggertes D-Latch
- 4 Zähler
- 5 Binary code decimal und 7-Segment Anzeige

BCD und 7-Segment Anzeige



Funktionsweise



Inputs						Outputs								
LE	BI	LT	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	B
X	0	1	X	X	X	X	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	0	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	0	0	0	0	0	0	
0	1	1	1	1	0	0	0	0	0	0	0	0	0	
0	1	1	1	1	0	1	0	0	0	0	0	0	0	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	
1	1	1	X	X	X	X				*				*

IC konvertiert 4-bit Nummer in ein zur 7-Segment Anzeige passendes Signal

Bemerkungen

Problem

- Zähler zählt bis 15, 7-Seg kann aber nur 0-9 anzeigen
- größte Zahl ist $(1001)_2 = (9)_{10}$
- AND Verbindung von 2. und 4. Bit auf alle Resets "löscht" alle Zahlen größer als 9

Würfel

- Rechtecksspannung kann als Input verwenden werden
- hochfrequentiges Zählen wird vom Auge nicht mehr wahrgenommen
- → Psudo-Zufallszahlen