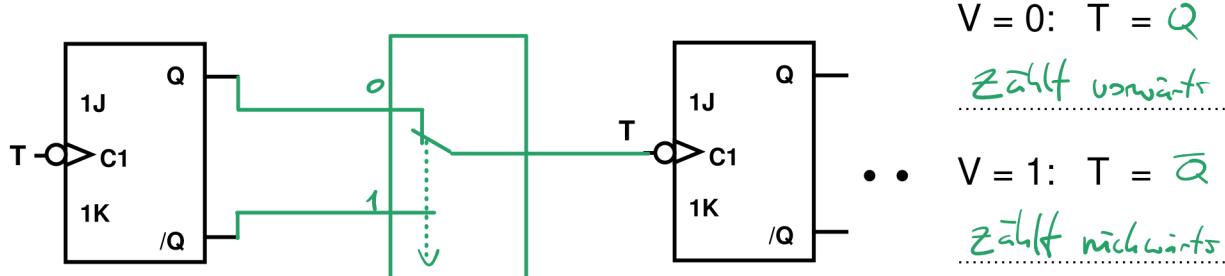


Zähler mit umschaltbarer Zählrichtung

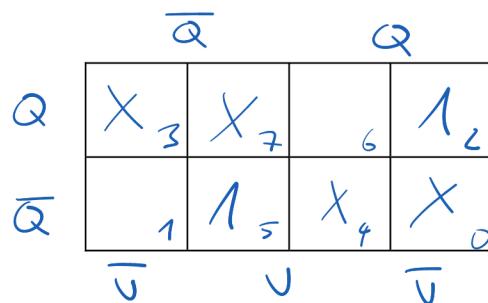
Prinzipschaltung:



Wertetabelle :

	V	Q	\bar{Q}	T
0	0	0	0	X
1	0	0	1	0
2	0	1	0	1
3	0	1	1	X
4	1	0	0	X
5	1	0	1	1
6	1	1	0	0
7	1	1	1	X

KV-Diagramm :

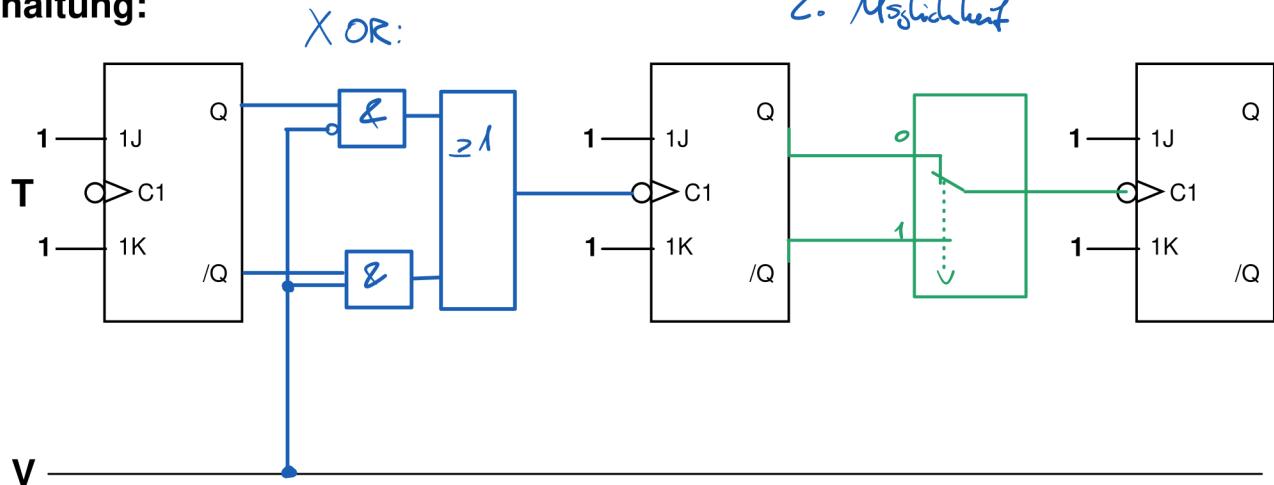


Funktionsgleichung :

$$(U_1 \bar{Q}) \vee (\bar{U}_1 Q) = T$$

XOR / Antivalenz

Schaltung:



Bauen Sie die Schaltung anschließend mit den entsprechenden TTL-ICs auf und prüfen Sie die Zählweise nach.

Ram

Massenspeicher / Optische (HDD / SSD)
Schnittstellen (SATA, PCIe)

Alle FlipFlop (SK
SK-Ms D)

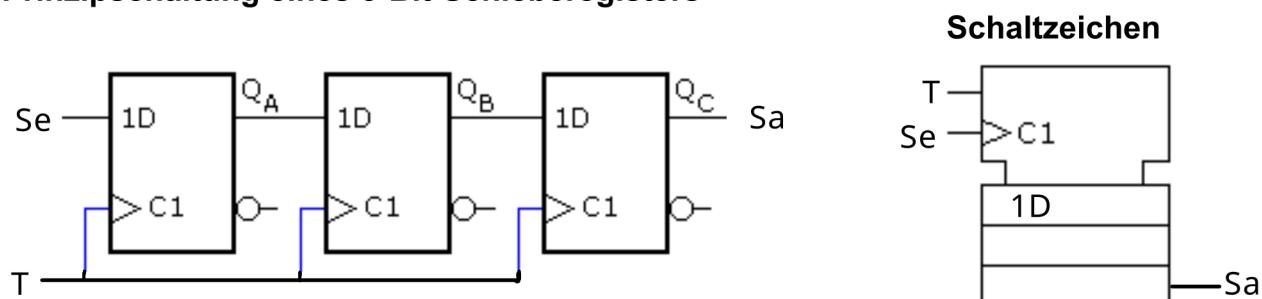
Asynchrone Zähler - Prinzip

Prinzip und Aufbau

Schieberegister: Kleine in Reihe geschaltete 1-bit-Speicher, deren Inhalt gleichzeitig durch ein Taktsignal verschoben wird.

Schieberegister werden mit D-FFs, JK-MS-FFs aufgebaut

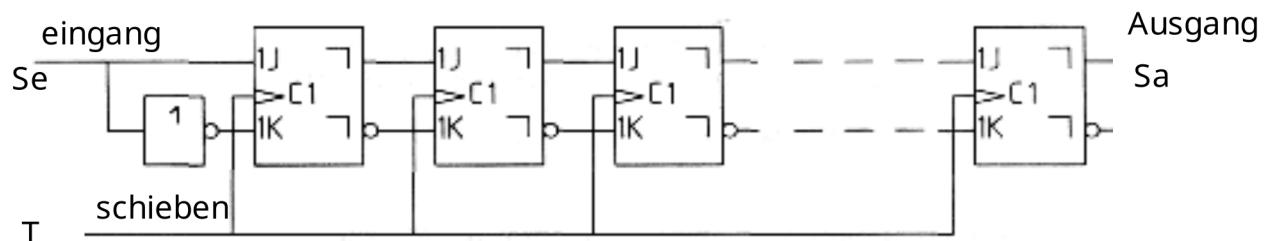
Prinzipschaltung eines 3-Bit-Schieberegisters



Arbeitsweisen der Schieberegister:

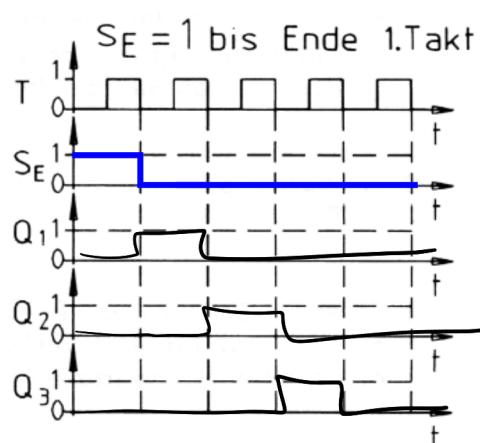
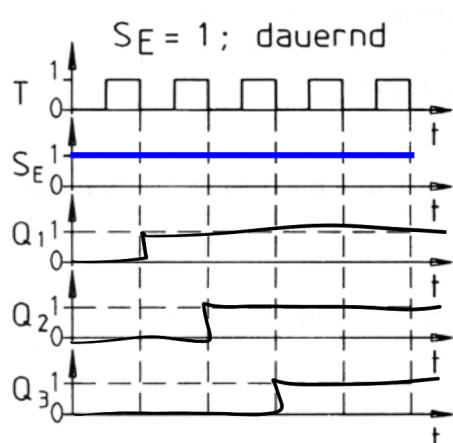
- Zahl der Stufen = Zahl der zu verarbeitenden Binärstellen/Bits
- Alle Stufen werden synchron vom Takt angesteuert.
Die Aufnahme der Information geschieht seriell (nacheinander).

1. Serieneingabe, Serienausgabe

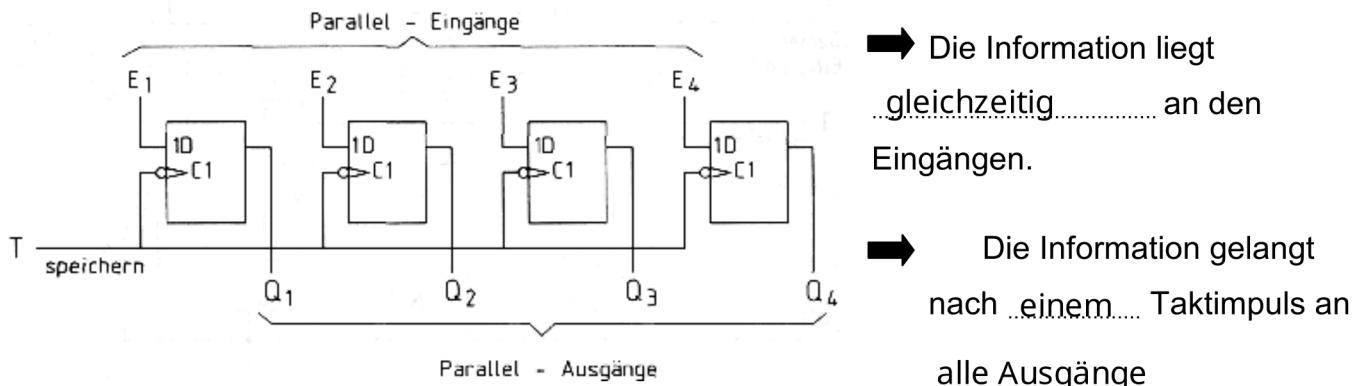


- Die Information wird nacheinander ein- und ausgeben
Mit jedem Takt wandert sie um ein FF weiter .

Ergänzen Sie die Zeitablaufdiagramme eines 3-bit-SRG mit serieller Eingabe nach Vorgabe.



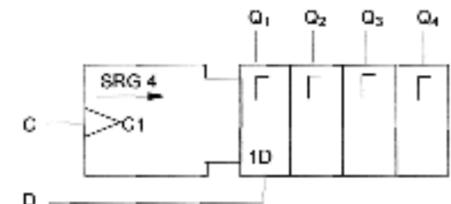
2. Parallele Eingabe, parallele Ausgabe



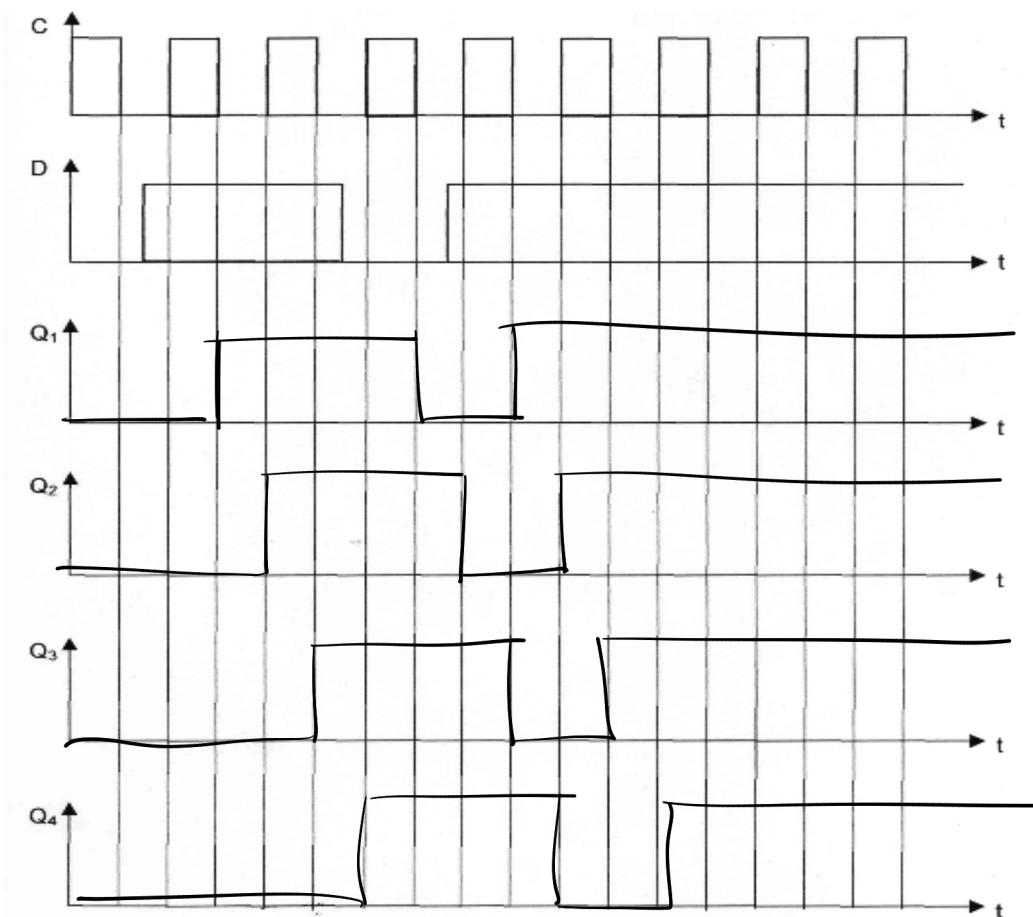
Aufgabe

An das Schieberegister werden die abgebildeten Daten gelegt.

1. Wie viele Bit werden in welcher Richtung verschoben? → 4
2. Zeichnen Sie den Verlauf der Ausgangssignale an Q₁ bis Q₄.



Zeitablaufdiagramme



Synchronzähler-Grundbausteine: JK-MS-FlipFlop, D-FlipFlop

4-Bit-Synchron-Dualvorwärtszähler

T	2^3 Q _D	2^2 Q _C	2^1 Q _B	2^0 Q _A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Die JK-Eingänge müssen so beschaltet werden,
dass die FF's im richtigen Augenblick kippen:

FF A muss nach jedem Takt..... kippen.

$$J_A = \dots 1 \dots K_A = \dots 1 \dots$$

FF B darf nur kippen, wenn $Q_A = 1$ ist.

$$J_B = \dots Q_A \dots K_B = \dots Q_B \dots$$

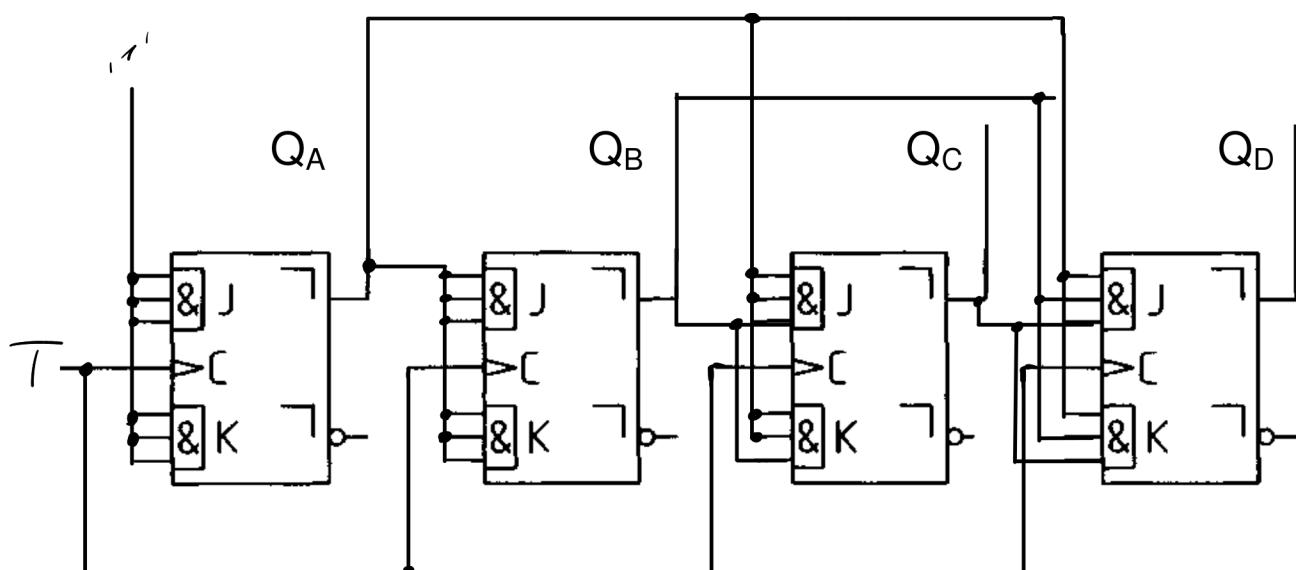
FF C darf nur kippen, wenn $Q_A = Q_B = 1$ ist.

$$J_C = \dots Q_A \wedge Q_B \dots K_C = \dots Q_A \wedge Q_B \dots$$

FF D darf nur kippen, wenn $Q_A = Q_B = Q_C = 1$ ist.

$$J_D = \dots Q_A \wedge Q_B \wedge Q_C \dots K_D = \dots Q_A \wedge Q_B \wedge Q_C \dots$$

Schaltung (realisiert mit IC SN 7472)



Asynchronzähler

- Der Ausgang des vorgehenden Flipflops erzeugt den Takt für das folgende Flipflop

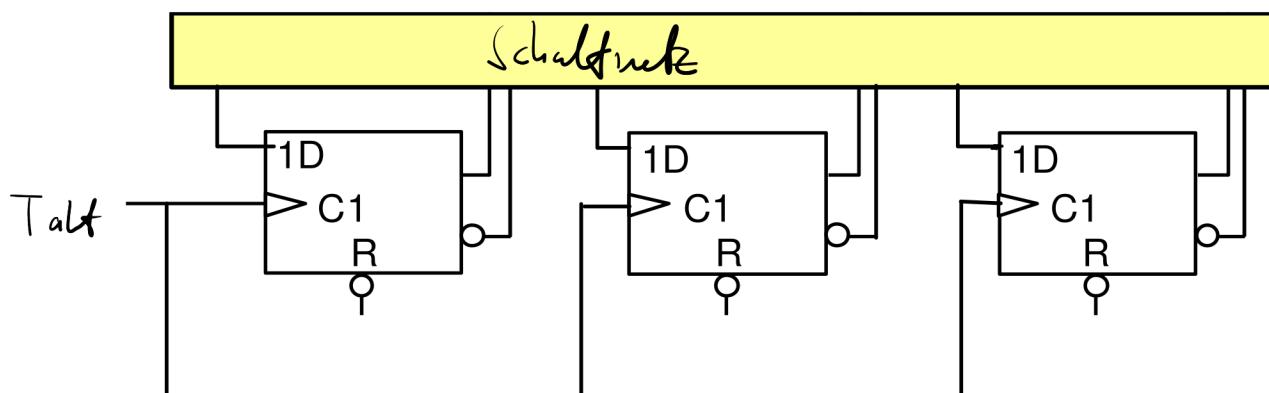
→ Vorteil: einfacher Aufbau

→ Nachteil: Zählerfrequenz begrenzt durch Signallaufzeiten
Flipflops schalten nicht gleichzeitig, dadurch entstehen falsche Zwischenzustände

Synchronzähler

- Alle Flipflops erhalten den gleichen Takt und schalten somit auch gleichzeitig
- Ein Schaltkreis legt fest, ob ein Flipflop bei einem Takt signal umschaltet oder nicht.
- Als FF werden vorwiegend positiv-taktfallende gesteuerte D-FF verwendet

Schaltungsprinzip:



→ Vorteile : Zählerfrequenz höher als bei Asynchronzählern Setzen und Rücksetzen ebenfalls taktsynchron

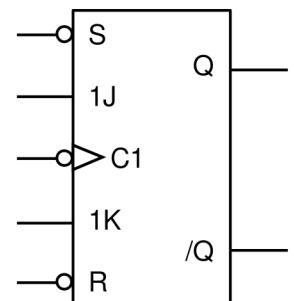
→ Nachteil : Komplexer Aufbau

Aufgabe 2: Asynchroner Zähler

Aus nebenstehendem Flipflop soll ein Vorwärtzzähler für 4 Bits aufgebaut werden, der mit einem gemeinsamen, taktunabhängig wirkenden Rückstelleingang versehen ist.

Das Zählergebnis soll mit den LED's binär angezeigt werden.

Anmerkung: taktunabhängig wirkende Rücksetzung wird auch als asynchrone Rücksetzung bezeichnet.

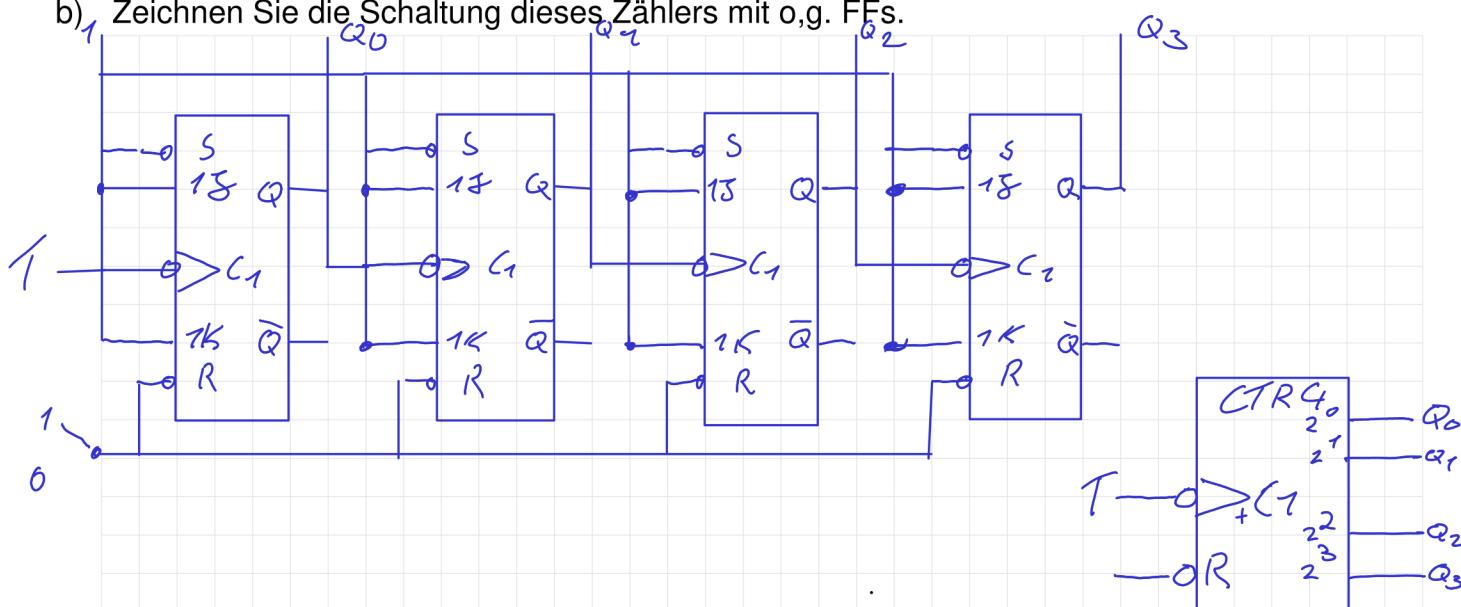


- a) Beschreiben Sie den Zähler in wenigen Worten (Zählername, Flipflopart).

Der binäre Asynchrone 4-bit Vorwärtzzähler aus vier JK Flipflops mit Vier eingängen

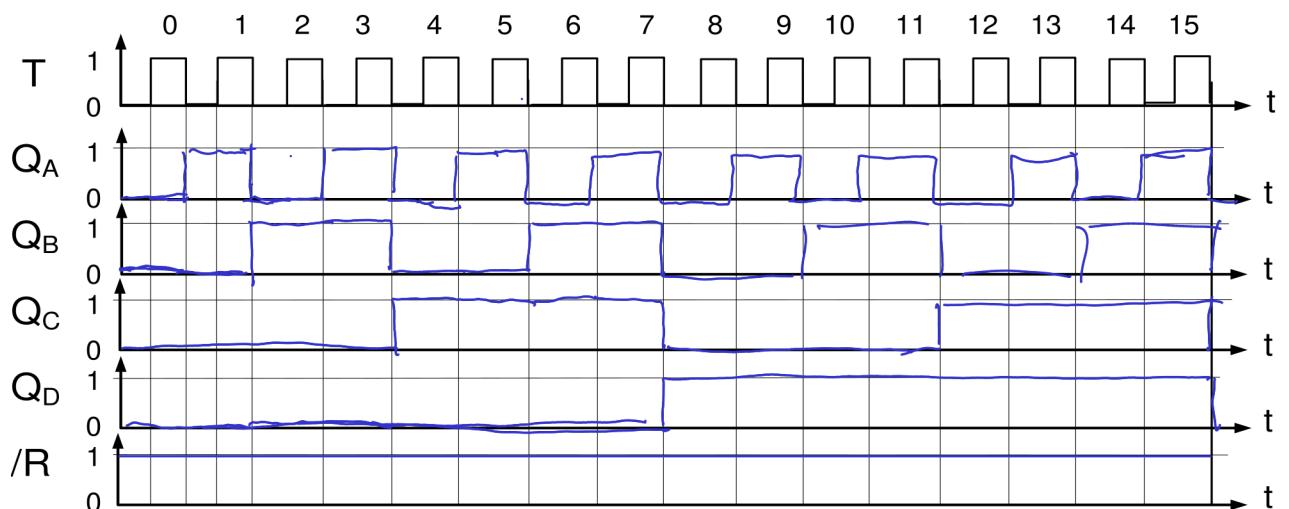
Zählvorgang: ab 0 bis 15 16 Zählmöglichkeiten

- b) Zeichnen Sie die Schaltung dieses Zählers mit o.g. FFs.



Bauen Sie die Schaltung auf und prüfen Sie die Funktion.

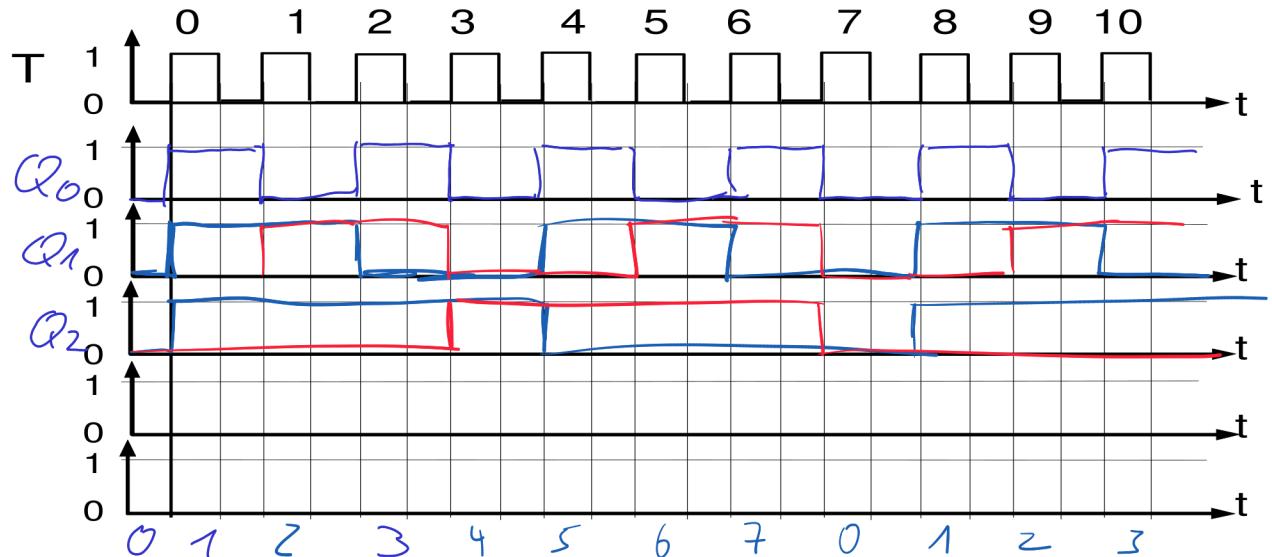
- c) Zeichnen Sie das Blockschaltbild und ergänzen Sie das Zeitablaufdiagramm des Zählers.



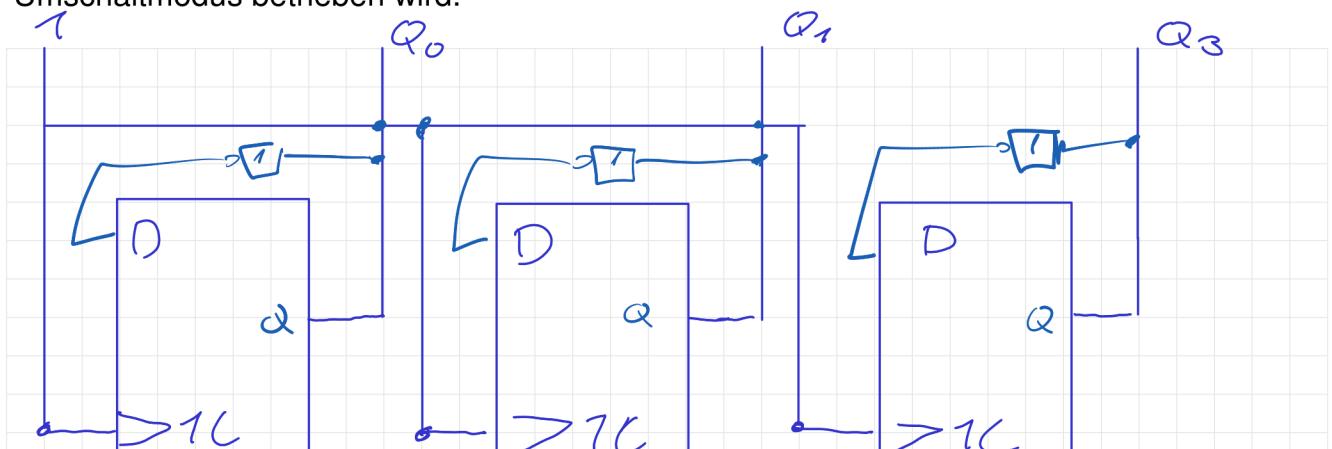
Aufgabe 3: Asynchroner 3-Bit-Dual-Vorwärtszähler

Aus positiv-taktflankengesteuerten D-Flipflops soll ein asynchroner 3-bit-Vorwärtszähler entworfen werden.

- a) Zeichnen Sie das Zeitablaufdiagramm des Zählers.



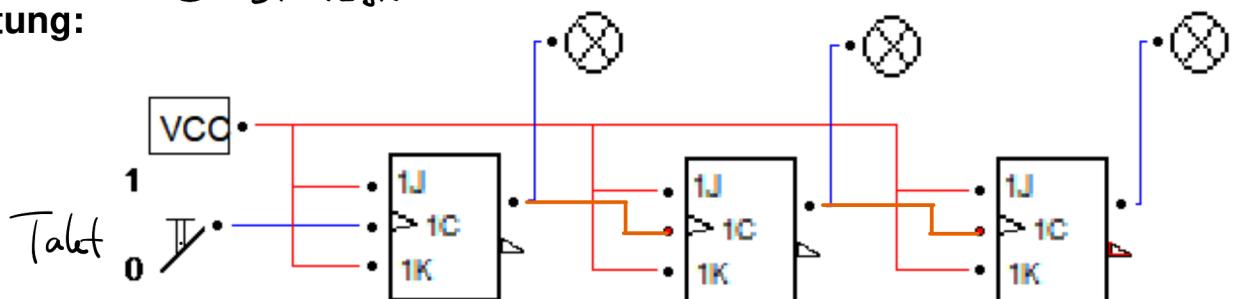
- b) Zeichnen Sie die Schaltung. Beschalten Sie das D-Flipflop dabei so, dass es im Umschaltmodus betrieben wird.



3-bit-dual-Rückwärtszähler aus positiv-taktflankengesteuerten JK-Flipflops

CTR 3 nach-JK

Schaltung:



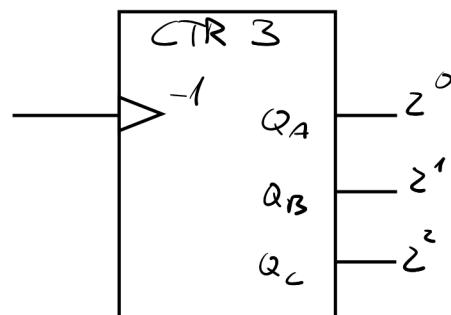
Q -Ausgänge zur Anzeige $\rightarrow Q$ -Ausgang am nachfolgenden Takteingang legen.

Alternativ: \bar{Q} -Ausgang bleibt am Takteingang, \bar{Q} -Ausgänge zur Anzeige

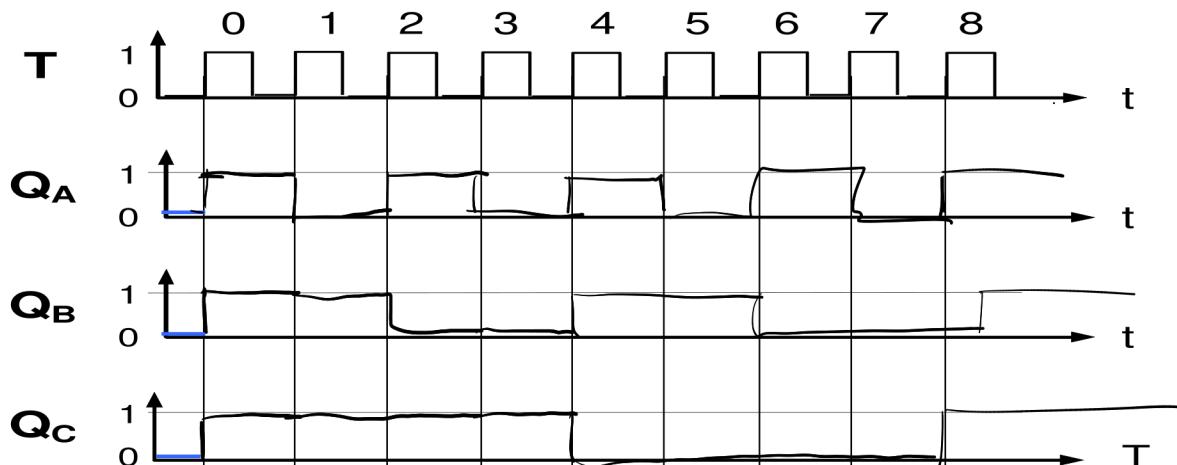
Wertetabelle :

T	2^2 Q _C	2^1 Q _B	2^0 Q _A	Dezimal
0	0	0	0	0
1	1	1	1	7
2	1	1	0	6
3	1	0	1	5
4	1	0	0	4
5	0	1	1	3
6	0	1	0	2
7	0	0	1	1

Blockschaltbild :



Zeitablaufdiagramm:



Fortführung Aufgabe 5: D-Flipflop

c) Zustandsgesteuertes (pulsgetriggertes) D-Latch

(Latch [engl.] = Raste, Klinke)

→ Ausgang Q „rastet“ auf das Signal am D-Eingang ein (vgl. Wertetabelle 1).

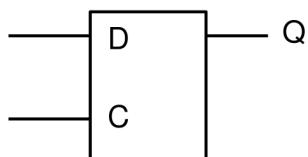


Bild 1: Schaltsymbol des statischen D-Flipflops

Wertetabelle 1:

C	D	Q_{n+1}
0	X	Q_n Speichern
1	0	0
1	1	1



Stat. D-Flipflop
https://www.youtube.com/watch?v=dnj1F_B08xA

d) Dynamisches D-Flipflop

→ meist positiv flankengetriggert.

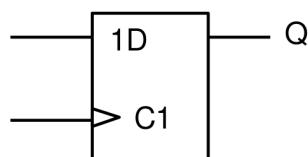


Bild 2: Schaltsymbol des dynamischen D-Flipflops

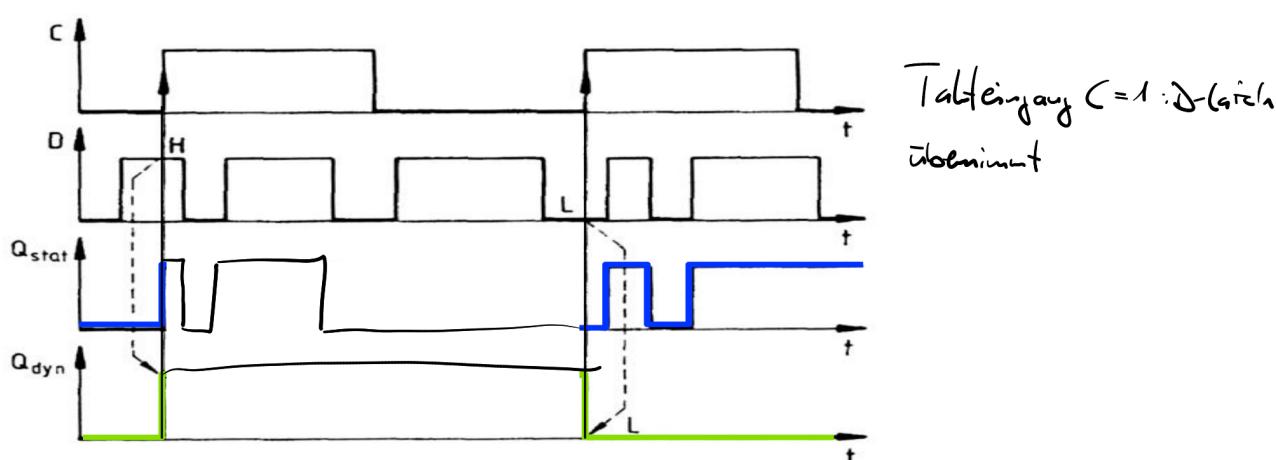
Wertetabelle 2:

C	D	Q_{n+1}
↓	X	Q_n Speichern
↑	0	0
↑	1	1

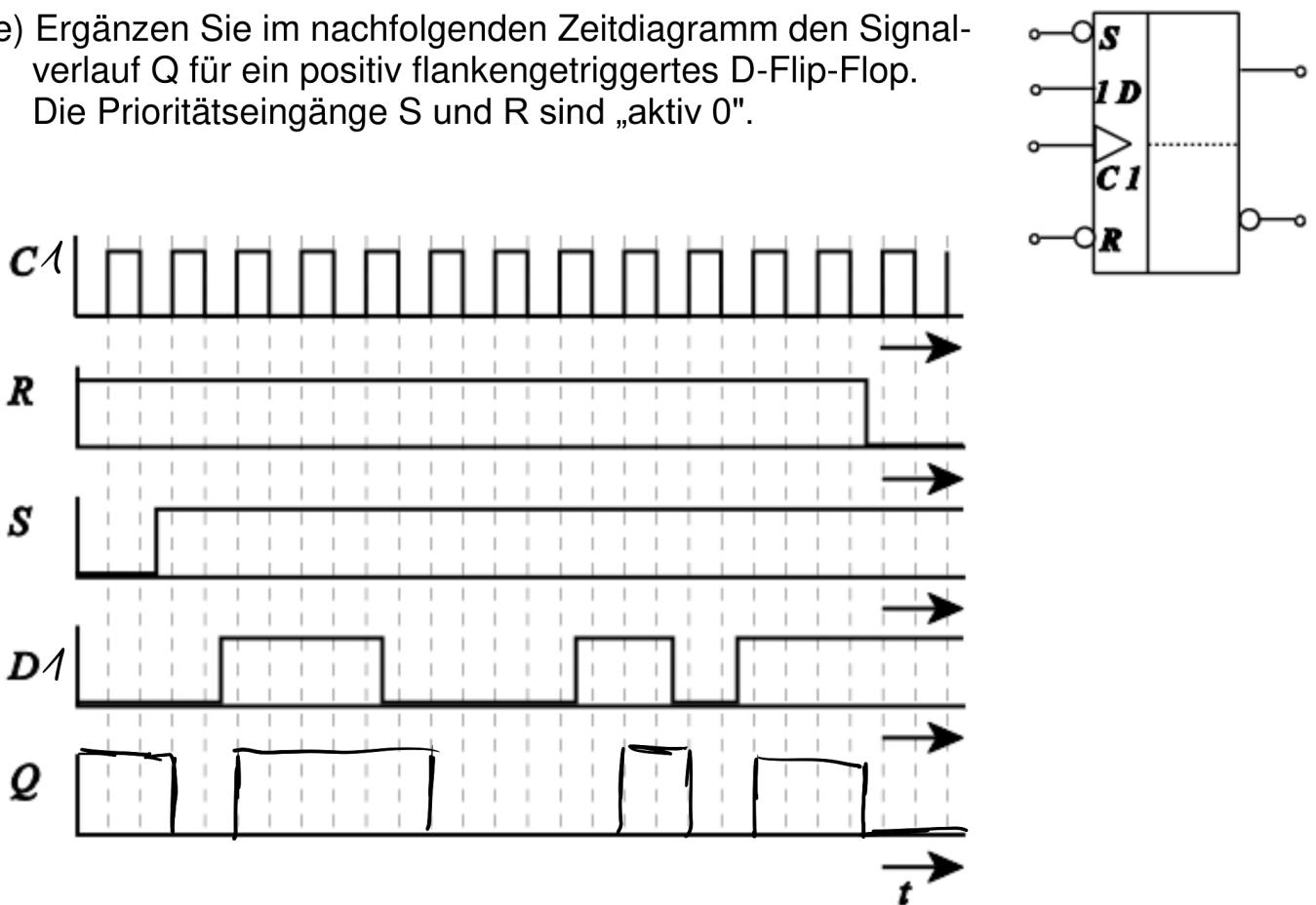


Dyn. D-Flipflop
https://www.youtube.com/watch?v=LKzbih_i_MI0

Unterschied zwischen statischer und dynamischer Ansteuerung :

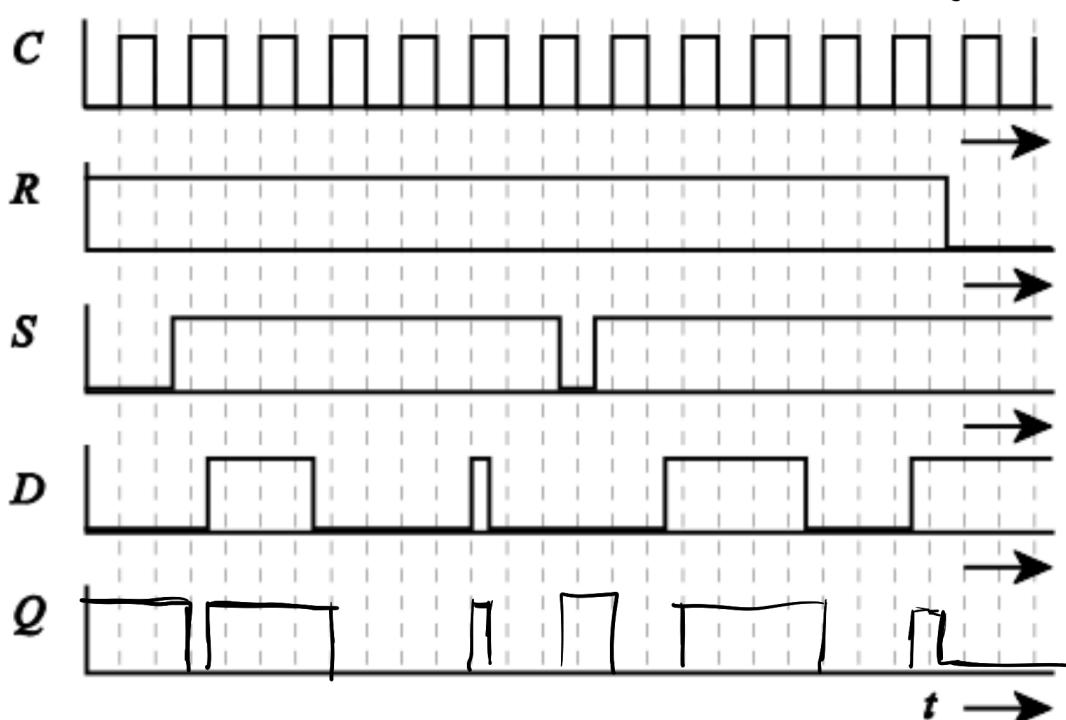


- e) Ergänzen Sie im nachfolgenden Zeitdiagramm den Signalverlauf Q für ein positiv flankengetriggertes D-Flip-Flop. Die Prioritätseingänge S und R sind „aktiv 0“.



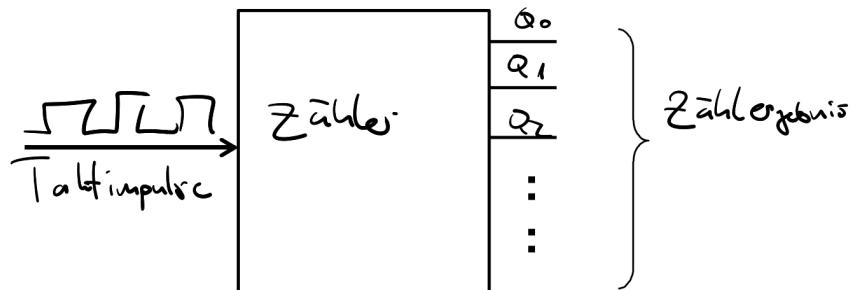
- f) Vervollständigen Sie das nachstehende Impulsdiagramm für ein positiv taktzustandsgesteuertes D-FF mit Prioritätseingängen und zeichnen Sie dessen Schaltsymbol!

Takt unabhängig



Input Zähler

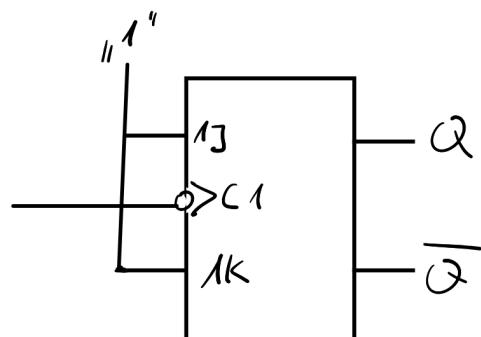
Prinzip: Taktimpulse werden aufaddiert und gespeichert; Ergebnis wird ausgegeben.



Grundbausteine: z.B. JK-FlipFlop

Schaltung:

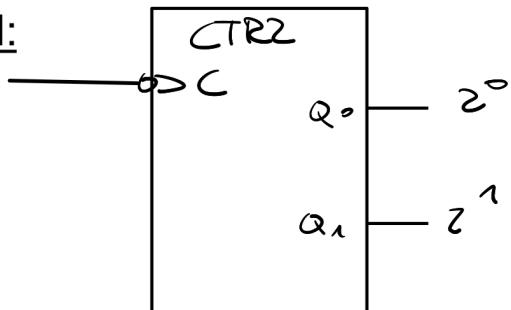
Zählfolge: $0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \dots \stackrel{\wedge}{=} \text{toggle mode} \rightarrow J = K = 1$



Weiter Stell(c(n)) → weiter(s) FF (Ausgang Q am Takt C)

⇒ asynchrones Dualzähler

Blockschaltbild:

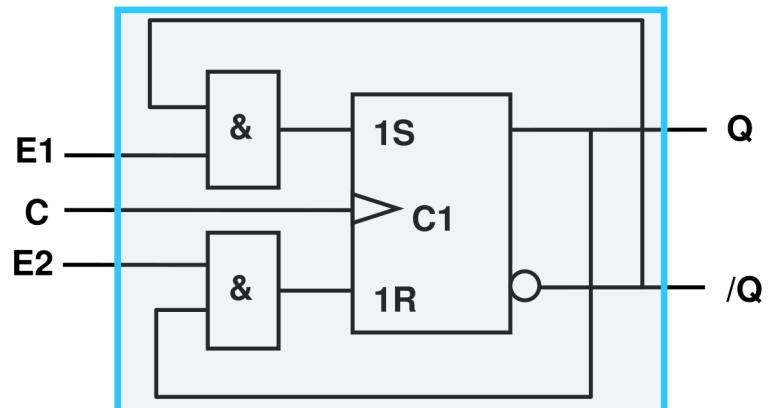


Aufgabe: SR-Flipflop mit Vorschaltung

- a) Beschreiben Sie die Ansteuerungsart am Eingang C des Flipflops.

Positiv flanken gesteuert

(austeigende Flanke)



- b) Beschreiben Sie die Bedeutung der „1“ vor den S bzw. R-Eingang und nach dem Takteingang C.

C1 ist für R1 und S1, C2 für R2 und S2, usw.

Ziffer nach C gibt die beeinflussten Eingänge an.

- c) Beschreiben Sie das Verhalten des Flipflops mit gegebener Wertetabelle:

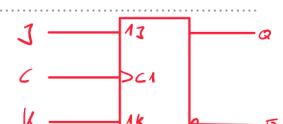
R		S				Bem.
E2	E1	C	Q _{n+1}	/Q _{n+1}		
K	J					
X	X	⊤	Q _n	¬Q _n	Speichern	
0	0	⊤	Q _n	¬Q _n	Speichern	
0	1	⊤	1	0	⊤: Q _{n+1} geht auf 1 = set	
1	0	⊤	0	0	⊤: Q _{n+1} geht auf 0 = Reset	
1	1	⊤	¬Q _n	Q _n	Q _{n+1} geht auf Q _n , d.h. wechselt mit ⊤ Zustand.	

Verhält sich wie ein RS-FF: $S \cong E1; R \cong E2$

Aber: Bei $E1=E2=1$ erfolgt abwechselnd Setzen und Rücksetzen.
(= toggle-mode)

JLK-Flipflop

Jump-Kill-Flipflop



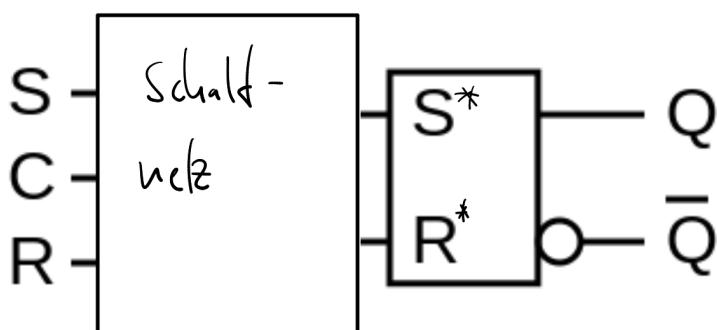
Problemstellung:

Manche Anwendungen verlangen, dass ein Flipflop nur zu einem bestimmten Zeitpunkt auf den Eingangszustand reagiert.

- Entwurf ein Bauteil, das sich eine Information nur zu einem bestimmten Zeitpunkt merkt.

Lösung:

Hinweis: An einem weiteren Eingang Takt (C) wird durch ein kurzzeitiges Taktsignal = 1 ($\sqcap\sqcup$) signalisiert, dass der aktuelle Wert von S bzw. R übernommen werden soll.



C	R	S	Q*	R*	Bemerkung
0	0	0	0	0	
0	0	1	0	0	
0	1	0	0	0	
0	1	1	0	0	
1	0	0	0	0	Speichern
1	0	1	1	0	Setzen
1	1	0	0	1	Rücksetzen
1	1	1	1	1	Nicht Speicherbar

$$\begin{aligned} S^* &= S \cdot C \\ R^* &= R \cdot C \end{aligned}$$

Verkürzte Wertetabelle:

C	R	S	Q	/Q	Bemerkung

Problemstellung:

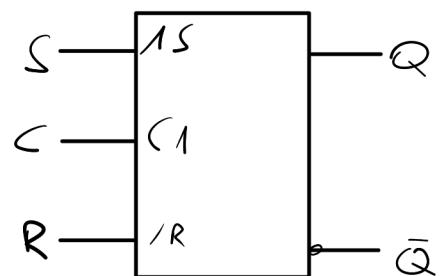
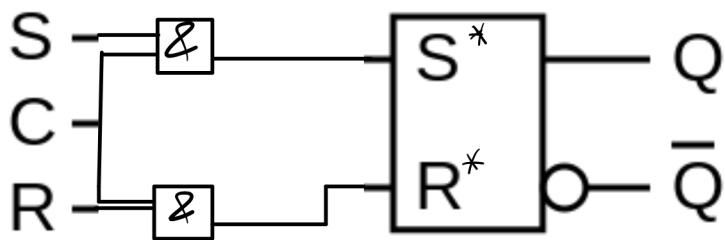
Manche Anwendungen verlangen, dass ein Flipflop nur zu einem bestimmten Zeitpunkt auf den Eingangszustand reagiert.

- Entwurf ein Bauteil, das sich eine Information nur zu einem bestimmten Zeitpunkt merkt.

Lösung:

Hinweis: An einem weiteren Eingang Takt (C) wird durch ein kurzzeitiges Taktsignal = 1 ($\uparrow L$) signalisiert, dass der aktuelle Wert von S bzw. R übernommen werden soll.

Taktgesteuertes Flipflop



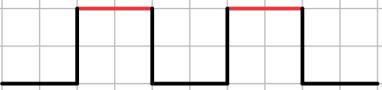
C	R	S	Q	/Q	Bemerkung
0	0	0	0	1	Speichern
0	0	1	1	0	Speichern
0	1	0	0	1	Speichern
0	1	1	1	0	Speichern
1	0	0	0	1	Speichern
1	0	1	1	0	Setzen
1	1	0	1	0	Rücksetzen
1	1	1	0	1	Nicht Speicherbar

} keine Änderung

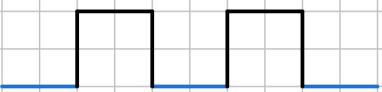
Verkürzte Wertetabelle:

C	R	S	Q	/Q	Bemerkung
0	X	X	0	1	Speichern
1	0	0	0	1	Speichern
1	0	1	1	0	Setzen
1	1	0	1	0	Rücksetzen
1	1	1	0	1	Nicht Speicherbar

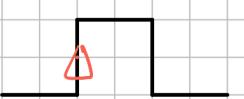
keine Änderung



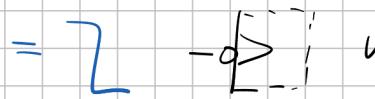
pos. Takt gest.



neg. Takt gest.



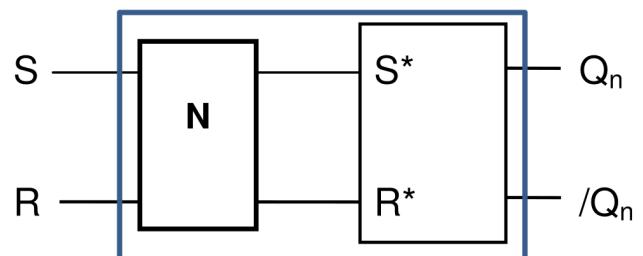
= pos. fahrtflanken gest.



= neg. fahrtflanken gest.

Aufgabe 1: Erweiterte RS - Speicherschaltung

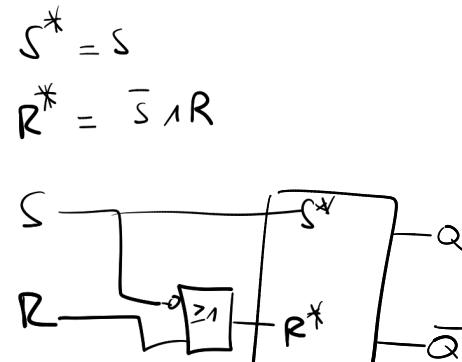
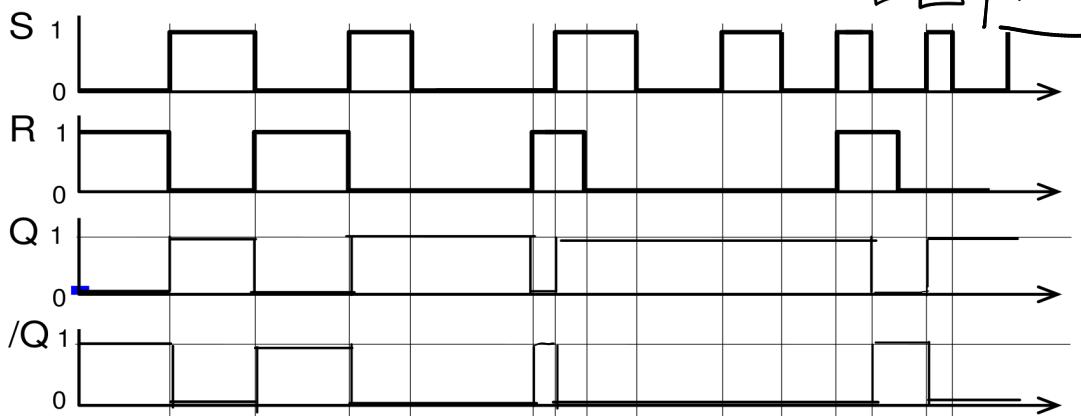
Durch eine kleine Zusatzschaltung N soll bei $R = S = 1$ der Ausgangszustand $Q = /Q$ verhindert werden.



- Entwickeln Sie eine Schaltung N, damit die Kippstufe bei $R = S = 1$ gesetzt (set) wird. Zeichnen Sie den Schaltplan der Gesamtschaltung.
- Erstellen Sie die kommentierte Wertetabelle der neuen Schaltung.

S	R	S*	R*	Bemerkungen
0	0	0	0	speichern
0	1	0	1	rücksetzen
1	0	1	0	setzen
1	1	1	0	setzen

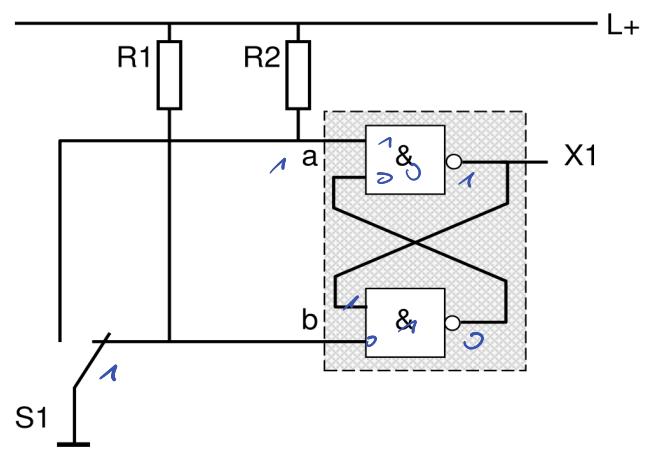
- Ergänzen Sie das Zeitablaufdiagramm.



Aufgabe 2: Prellfreier Schalter

Mechanische Taster und Schalter haben den Nachteil, dass sie durch ihren Aufbau prellen. Werden sie betätigt, wird durch das Kippen eine Kraft auf den Kontakt ausgelöst, die ihn im Submillimeterbereich schließen und öffnen lässt, das zu unerwünschten Signalunterbrechungen bzw. Mehrfachimpulsen führt. Da in der Digitaltechnik aber einwandfreie Zustände gefordert sind, darf ein Zurückfallen nicht entstehen.

Die abgebildete elektronische Schaltung dagegen gibt am Ausgang nur einen exakten Rechteckimpuls ab.



Erklären Sie die Arbeitsweise der Schaltung in wenigen Sätzen.

S_1 ist der Schalter, der an 2 NAND-Gatter geschaltet ist.
 x_1 ist der Anfang.

Wenn Schalter auf b: NAND-Gatter schaft (unlöslich) ab was ($b=1$)
($a=0$)

S1 in Ausgangslage

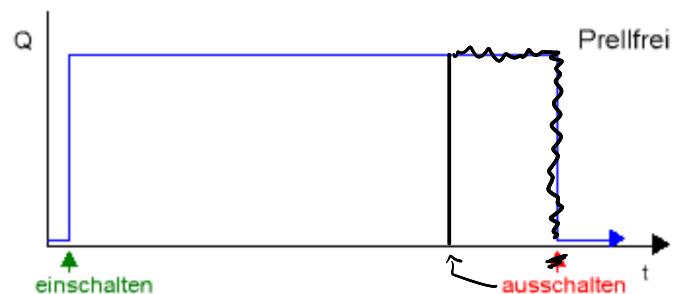
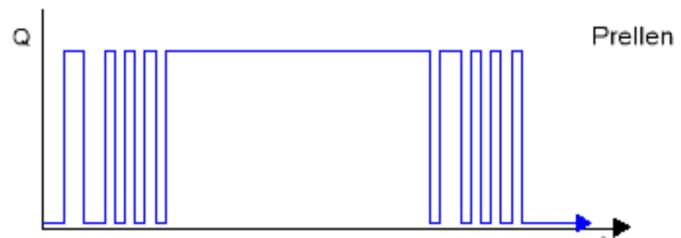
b liegt auf "0", a über R2 an "1"

→ Ausgang X1 auf "0" Prext

Umschalten von S1

Mit dem ersten Reihen an "0"
an a und "1" an b

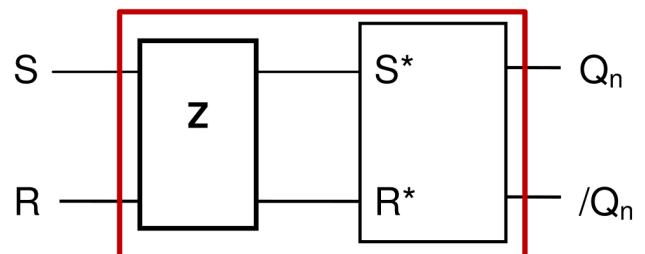
→ Ausgang X1 geht danach auf "1" set.



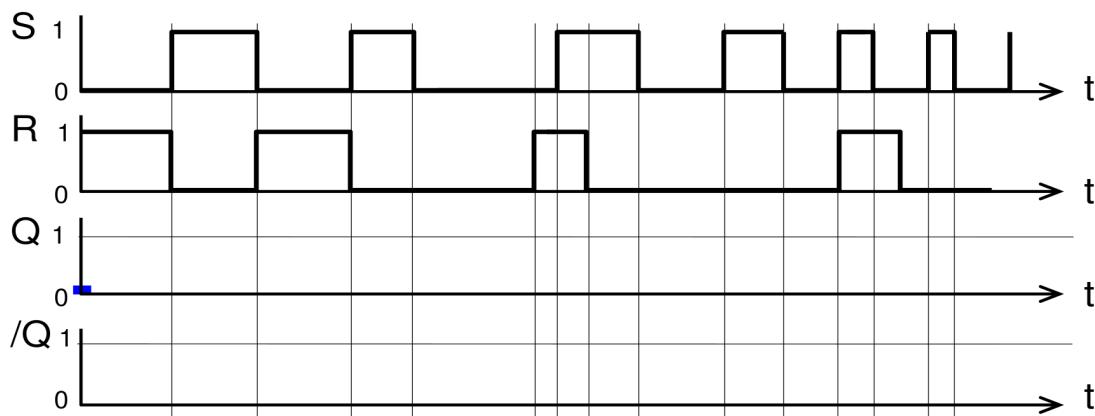
http://www.berel-am-ries.de/WVSS/Arbeitsvorlagen/Prellfreier_Schalter/Endprellen%20eines%20Umschalters.htm

Hausi-Aufgabe 3: Erweiterte RS - Speicherschaltung

Durch eine kleine Zusatzschaltung Z soll bei $R = S = 1$ der Ausgangszustand $Q = /Q$ verhindert werden.



- Entwickeln Sie eine Schaltung Z, damit das Basisflipflop bei $R = S = 1$ rückgesetzt (reset) wird.
Zeichnen Sie den Schaltplan der Gesamtschaltung im DigitalSimulator.
- Erstellen Sie die kommentierte Wertetabelle der neuen Schaltung.
- Ergänzen Sie das Zeitablaufdiagramm.

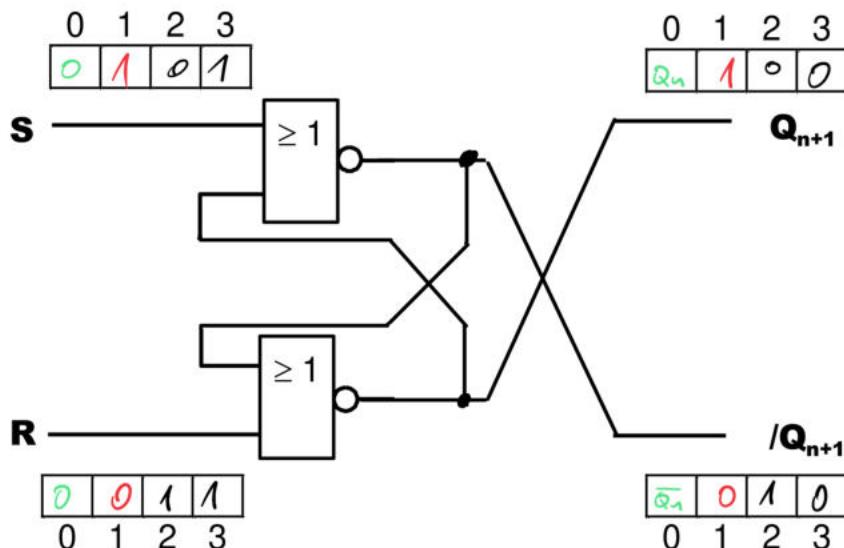


Z:

	R	S	R^*	S^*
0	0	0	0	0
0	1	0	1	Rücksetzen
1	0	1	0	
1	1	0	1	

RS - Flipflop aus NOR (NOR - Latch, Basis-Flipflop)

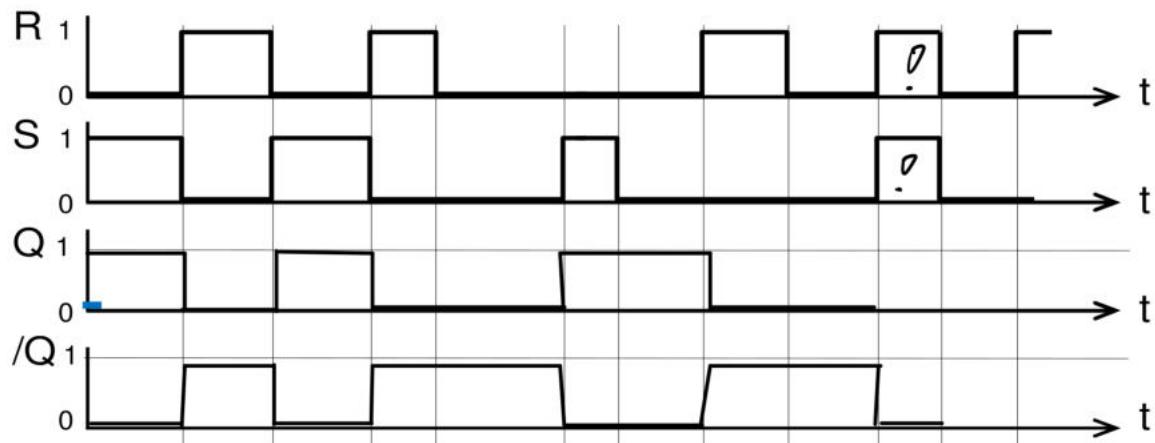
- Ergänzen Sie die Schaltung zu einem RS-Flipflop aus Nor-Gattern.
- Erarbeiten Sie die internen Abläufe des Flipflops und ergänzen Sie die Wertetabelle.
- Ergänzen Sie das Zeitablaufdiagramm.



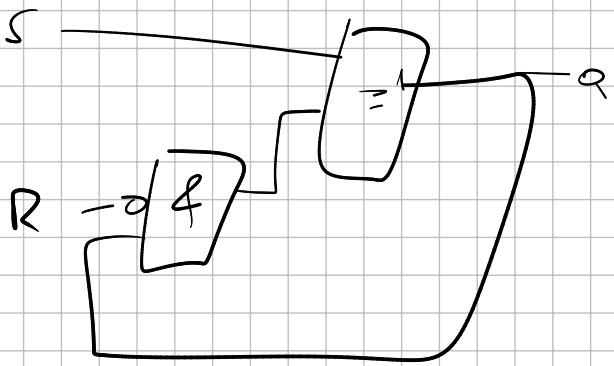
Zeile	R	S	Q _{n+1}	/Q _{n+1}	Bemerkung
0	0	0	Q _n	Q̄ _n	Speichern
1	0	1	1	0	Setzen (set)
2	1	0	0	1	Rücksetzen (reset)
3	1	1	0	0	Nicht speicherbar

Schaltungstechnisch verursacht (math. nicht definiert): $Q = \bar{Q}$

Zeitablaufdiagramm:

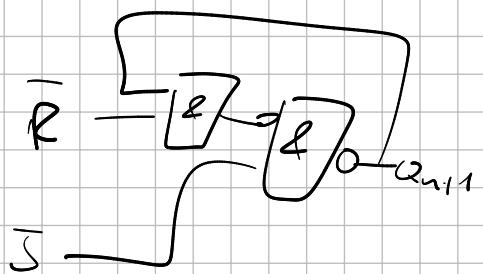


Zeile	R	S	Q_{n+1}	\bar{Q}_{n+1}
0	0	0		
1	0	1		
2	1	0		
3	1	1		



$$Q = S \vee (S \wedge \bar{R})$$

Full NAND



RS - Flipflop aus NAND - Gattern

a) Erarbeiten Sie das logische Verhalten des Flipflops und ergänzen Sie die Wertetabelle.

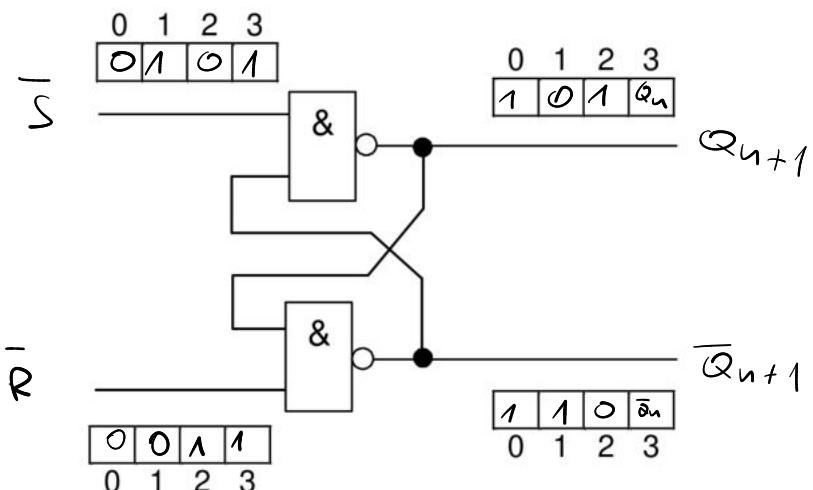
b) Vergleichen Sie die Wertetabellen des NOR- und NAND- FF's.

Das RS-Flipflop aus NAND hat lowaktive Eingänge und wird deshalb mit $S = \bar{0}/R = 0$ gesetzt/rückgesetzt.

c) Geben Sie das Schaltzeichen an und ergänzen Sie das Zeitablaufdiagramm.

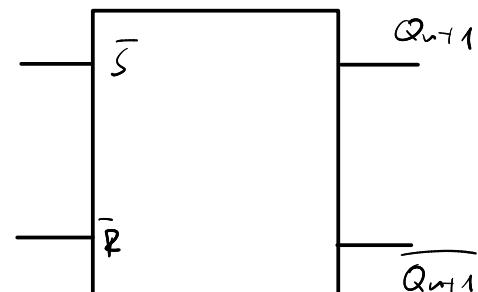
Zu a):

lowaktiv

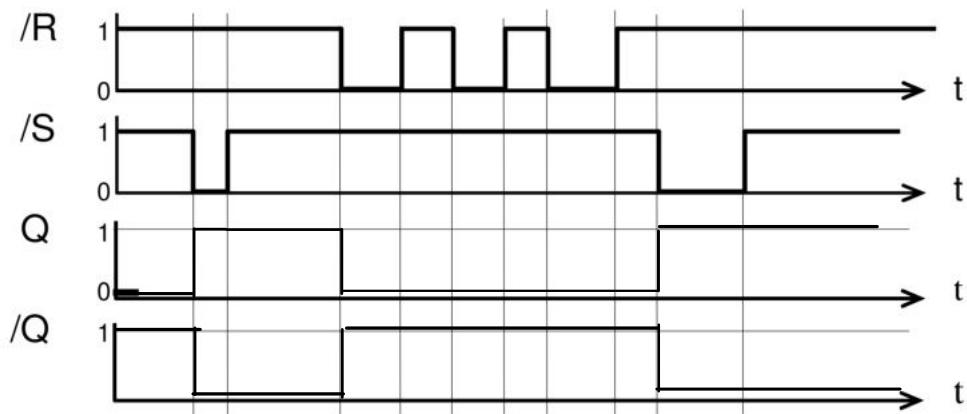


Zu a) Wertetabelle :

Nr.	\bar{R}	\bar{S}	Q_{n+1}	$/Q_{n+1}$	Bemerkungen
0	0	0	1	1	Nicht speicherbar
1	0	1	0	1	Rücksetzen
2	1	0	1	0	Setzen (set)
3	1	1	Q_n	\bar{Q}_n	speichern



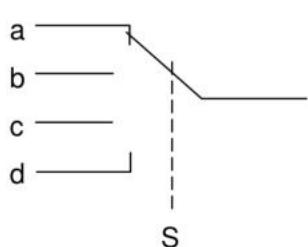
Zu c): Zeitablaufdiagramm :



Aufgabe: Mehrere parallel anliegende Daten über einen einzigen Übertragungskanal schicken.

⇒ Datenselektoren

Multiplexer – Prinzip



Demultiplexer-Prinzip

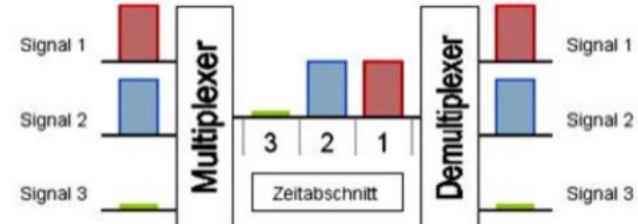
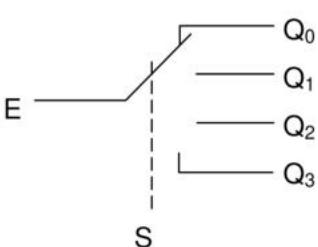


Bild 1 : Schematische Darstellung der Funktionsweise

In bestimmten Zeitabschnitten werden die Daten von verschiedenen Sendern auf einem Kanal übertragen.

Ein Multiplexer bzw. Datenselektor wählt aus verschiedenen anliegenden Daten die geforderte aus und leitet diese über den Ausgang weiter.

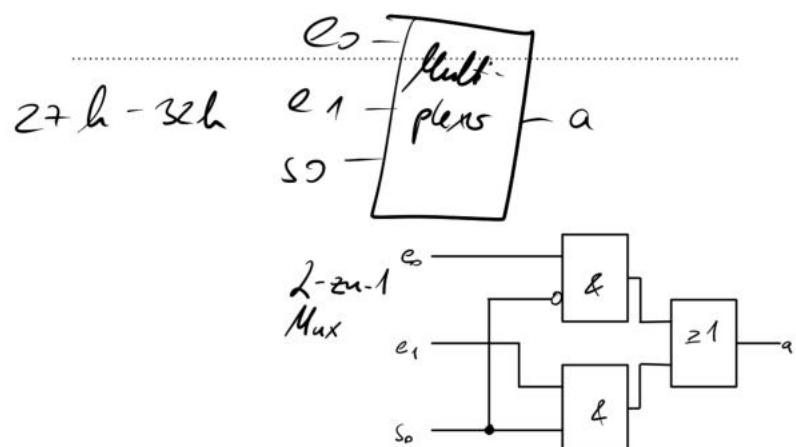
- Multiplexer sind somit Parallel / Scill-Umsetzer.

Einfaches Beispiel:

- zwei Signale e_0 und e_1 werden mit Hilfe eines Steuereinganges s_0 auf eine Leitung a umgesetzt.

Wertetabelle

s_0	e_1	e_0	a	Kommentar
0	0	0	0	Eingang
1	0	1	1	
2	0	1	0	e_0 ausgewählt
3	0	1	1	
4	1	0	0	Eingang
5	1	0	0	
6	1	1	0	e_1 ausgewählt
7	1	1	1	



Verkürzte Wertetabelle

s_0	a
0	e_0
1	e_1

\bar{e}_0	\bar{e}_1	\bar{s}_0	\bar{s}_1
1	1	1	1
1	0	1	0
0	1	0	1
0	0	0	0

$$a = (\bar{s}_0 \cdot \bar{e}_0) \vee (e_1 \cdot s_0)$$

Aufgabe: Erklären Sie die Funktionsweise des Multiplexers

4-zu-1-Datenselektor

Das Bild zeigt einen 1-aus-4- oder 4-auf-1-Multiplexer, zu dem wir auch 4-zu-1-Datenselektor sagen können.

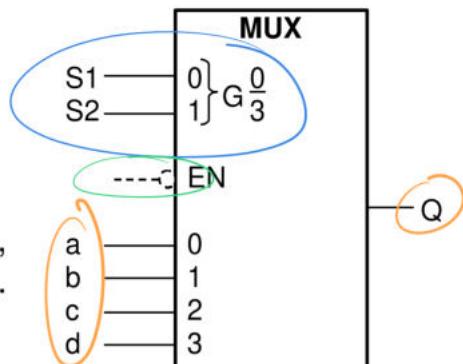


Bild: DIN-Symbol eines 1-aus-4-Multiplexers

Ein 4-zu-1 Multiplexer schaltet einen der vier Eingänge a-d zum Ausgang Q durch.

Diesen wählen die Steuereingänge S₁ und S₂ aus.

Bsp.: S₂=1, S₁=0

→ der mit 2 beschriftete Eingang c.

Der oftmals vorhandene passativer enable Eingang EN kann die Schaltung abschalten.

Demultiplexer

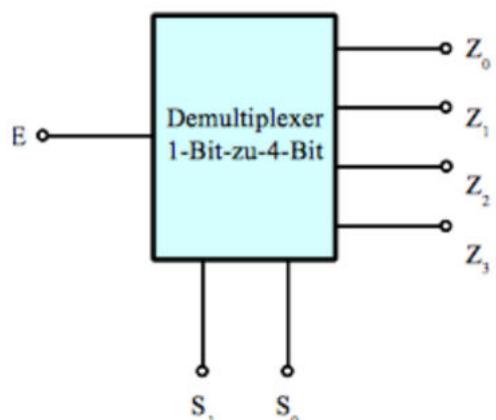
Ein Demultiplexer realisiert die Umgekehrte Funktion des Multiplexers.

Ein Demultiplexer verteilt die am Eingang erscheinenden Daten über Selektoreneingängen an den gewählten Ausgang

Demultiplexer erfüllen die Funktion eines Serial-/Parallel-Umsetzter.

Aufgabe:

Erklären Sie die Funktionsweise des abgebildeten Demultiplexers!



Problembeschreibung

1-Signal am Eingang 1 und 0-Signal an Eingang 2 setzt Ausgang Q auf 1

= Setzen (Wir nennen den Eingang dann Set, S)

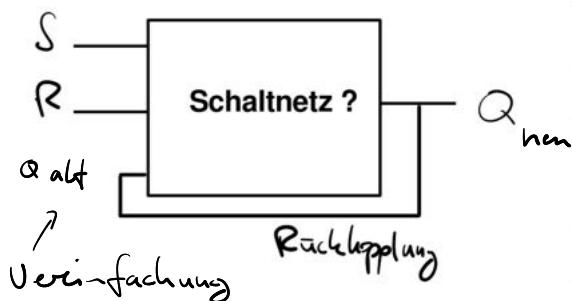
1-Signal am Eingang 2 und 0-Signal an Eingang 1 setzt Ausgang Q auf 0

= Rücksetzen (Reset, R)

Ausgangswert bleibt bei $S = R = 0$ erhalten = Speichern

Lösung

Wertetabelle



S	R	Q _{alt}	Q _{nem}	Bemerkungen
0	0	0	0	Speichern
0	0	1	1	Speichern
0	1	0	0	Rücksetzen
0	1	1	0	Rücksetzen
1	0	0	1	Setzen
1	0	1	1	Setzen
1	1	0	X	Nicht beschrieben
1	1	1	X	Nicht beschrieben

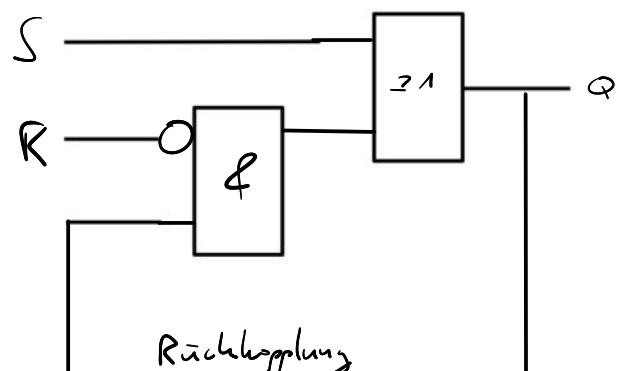
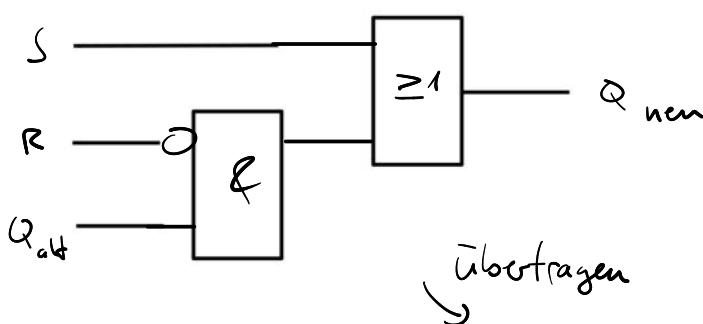
KV:

	Q_{alt}	1	\bar{Q}_{alt}	
R	X	X		
\bar{R}	1	1	0	
	1	1	0	

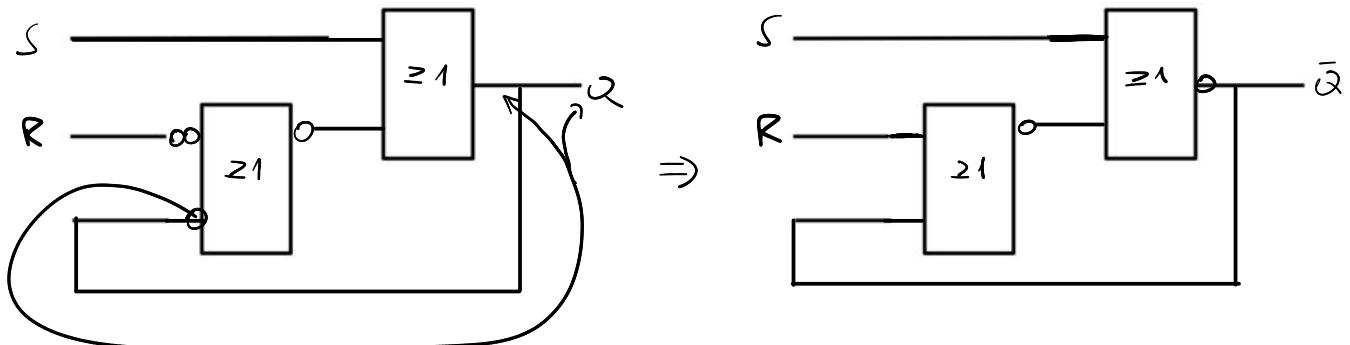
Funktionsgleichung:

$$Q_{nem} = S \vee (Q_{alt} \wedge \bar{R})$$

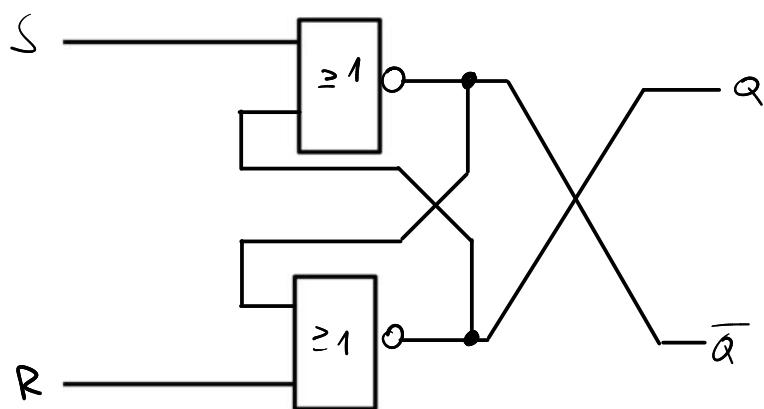
Schaltung:



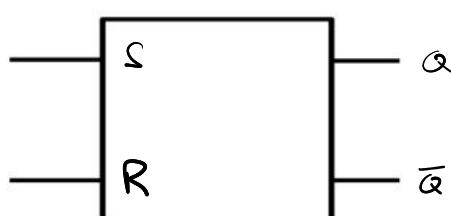
Umwandlung in NOR:



Übliche Darstellung:



Schaltbild:



\Downarrow
 S R - bzw. RS Flipflop
 Basis-Flipflop

Speicherschaltungen = Flipflop = Bistabile Kippstufe

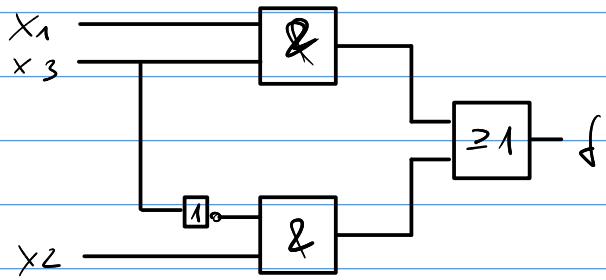


Kombinatorische Logik

- nur aus Gattern
- keine Rückkopplungen
- werden als Schaltnetze bezeichnet

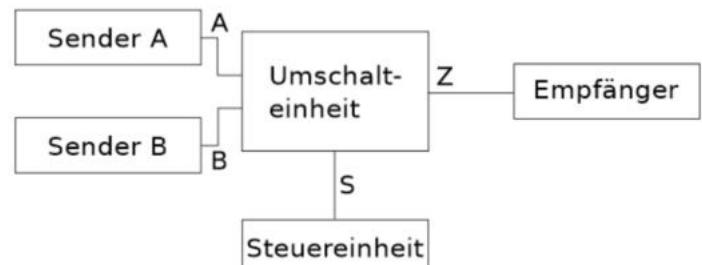
Sequentielle Logik

- aus Gattern und Flipflop
- Mit Rückkopplungen
- werden als Schaltworte bezeichnet



Aufgabe 1: Zugriff auf eine gemeinsame Datenleitung

Die Signale zweier verschiedener Geräte A und B sollen auf einer Datenleitung zum Empfänger übertragen werden. Um Konflikte zu vermeiden, übernimmt eine Steuereinheit S die Kontrolle und lässt jeweils nur einen Sender die Leitung benutzen. Ist das Steuersignal $S = 1$, sollen die Daten von A am Ausgang Z der Umschalteinheit erscheinen, bei $S = 0$ wird B mit dem Empfänger verbunden.

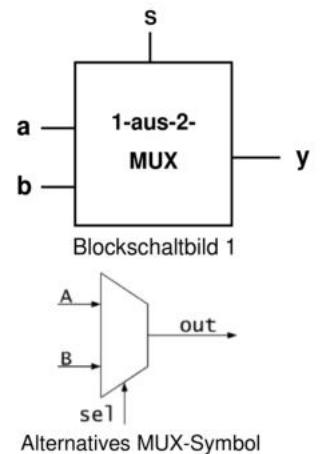


Entwerfen und zeichnen Sie die Schaltung aus Grundgattern für die obige Umschalteinheit. Wie nennt man diese Schaltungen noch?

Aufgabe 2: Multiplexer

Einen n-zu-1-Multiplexer kann man auch durch mehrfaches Kaskadieren von 2-zu-1-Multiplexern aufbauen. So lässt sich z.B. ein 4-zu-1-Multiplexer aus 2-zu-1-Multiplexern aufbauen.

- Setzen Sie solch einen Datenselektor aus Multiplexern nach Blockschaubild 1 zusammen.
- Wie viele 2-zu-1-MUX benötigt man für einen 8-zu-1-MUX?



Aufgabe 3: 1-bit-ALU

Multiplexer können auch in ALUs als Auswahl schalter eingesetzt werden (vgl. Bild). Über die Operationsauswahl (Mode) wird so festgelegt, welche arithmetische oder logische Operation ausgeführt werden soll.

Ziel dieser Aufgabe ist der Entwurf einer einfachen 1-Bit-ALU (vgl. Praktikum), deren Ergebnis-Ausgabe über eine Multiplexerschaltung erfolgen soll, die über die Signale S_0 und S_1 gesteuert wird und soll die Rechenoperationen der gegebenen Tabelle ausführen können.

- Ergänzen Sie das folgende Schaltbild durch eine entsprechende Multiplexerschaltung aus einzelnen Gattern, die die angegebenen Rechenoperationen ausgibt.
- Der MUX soll nun in Blockschaubild-Darstellung aus 2-zu-1-MUX gezeichnet werden.

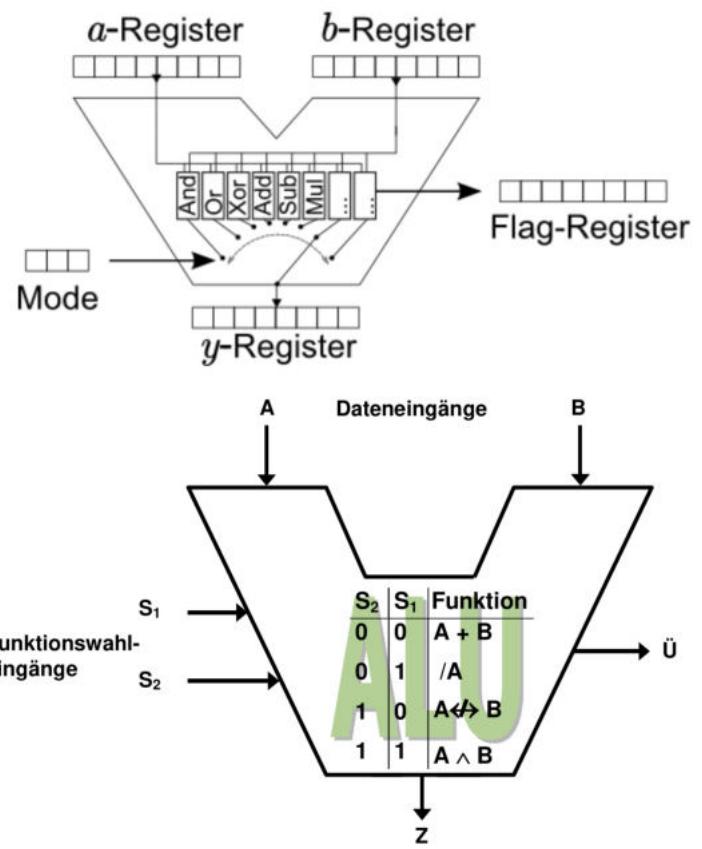
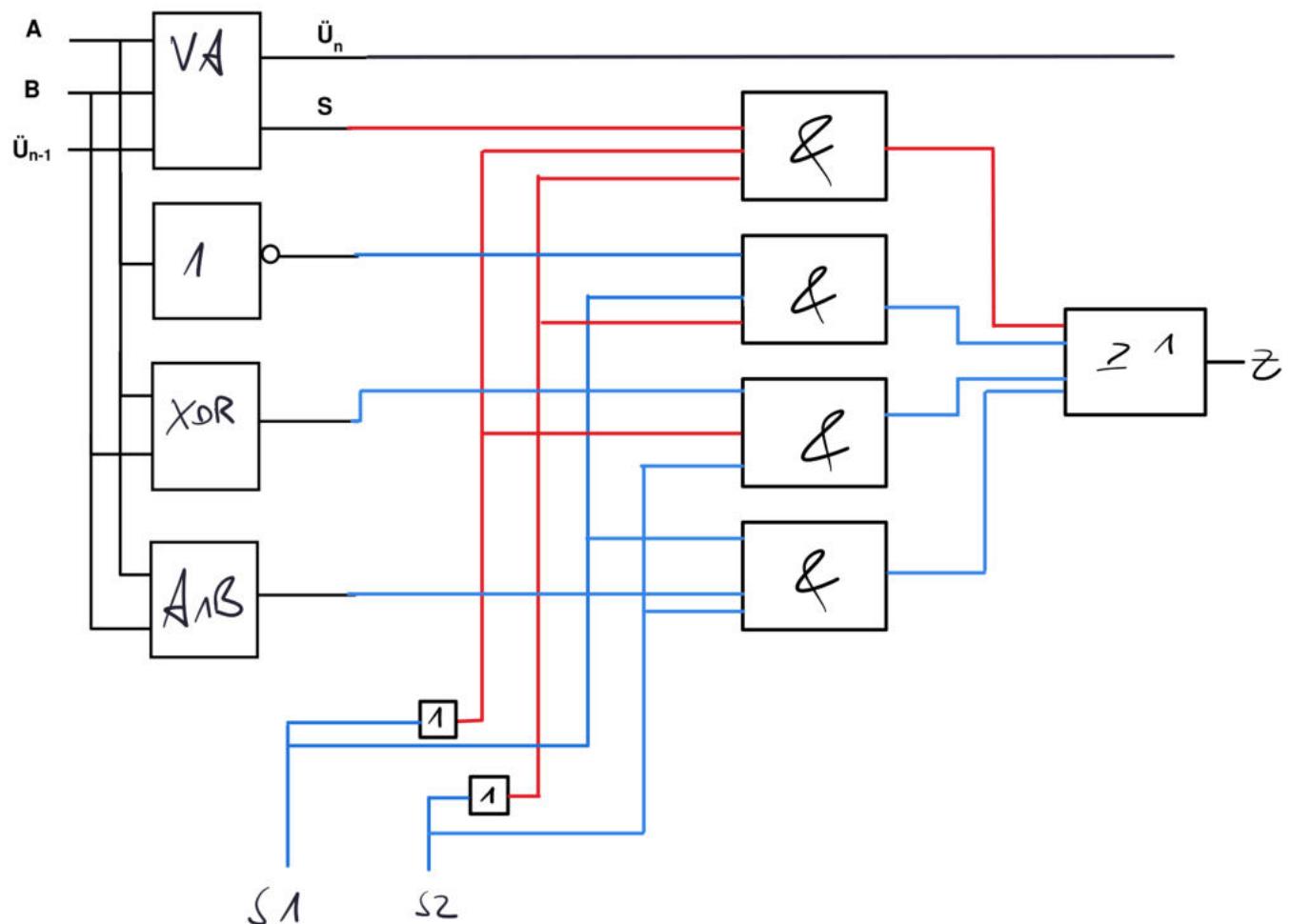
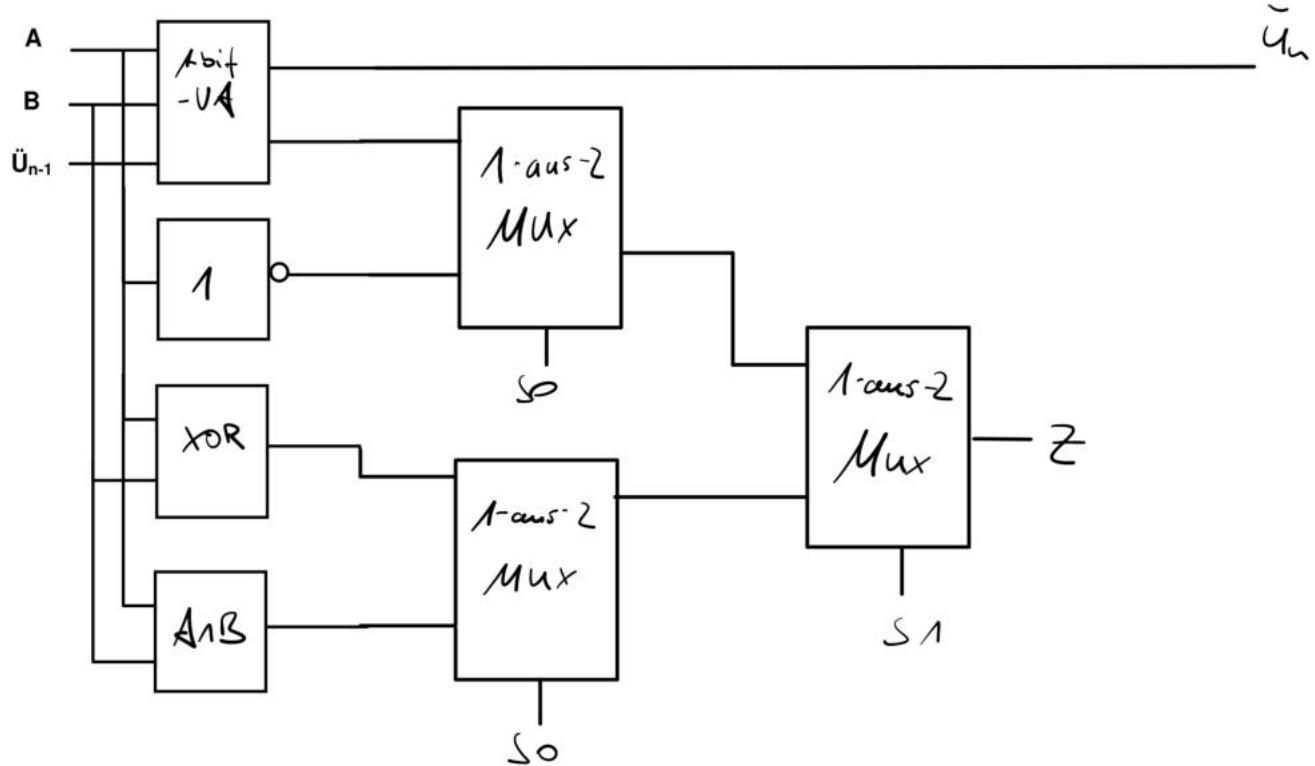


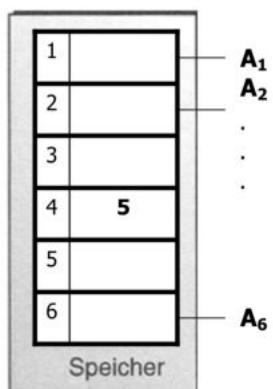
Bild: Blockschabild 1-bit-ALU



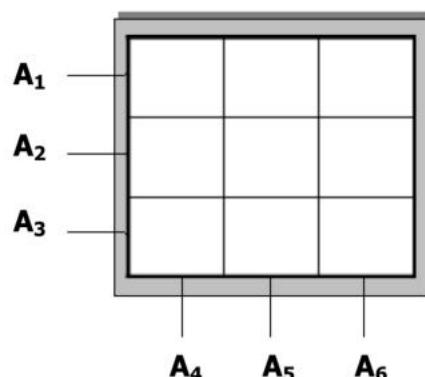


Wie werden Speicherzellen adressiert ?

1. Bsp.:



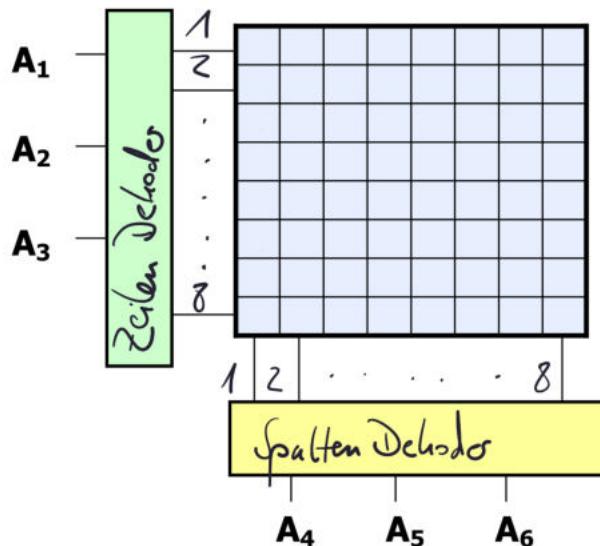
2. Bsp.:



- $z = 6$ Zellen
 $\rightarrow n = 6$ Adressleitungen nötig
- Allgemein: $z = n$
- **Aufwand an Adressleitungen bei z.B. nur 100 bit!!**

- $n = 6$ Adressleitungen
 $\rightarrow z = 9$ Speicherzellen adressierbar
- Allein durch die Anordnung lässt sich die Anzahl der Speicherzellen erhöhen
- Allgemein: $z = \left(\frac{n}{z}\right)^z$
- **Das wären bei 100 bit aber immer noch Adressleitungen!!**

3. Bsp.: Adressierung von Zeilen und Spalten über jeweils eigenen Decoder



- $n = 6$ Adressleitungen
 - **Adressdecoder**
 \rightarrow **Multiplexer** wählt einen aus z möglichen Ausgängen an
 - $z = 64$ Speicherzellen adressierbar
- Allgemein: $z = 2^n$

Aufgabe 1 : 1-aus-4-Multiplexer (Datenselektor)

Entwickeln Sie eine 4-zu-1- Multiplexer-Schaltung, die mit den Steuersignalen S1 und S2 (Selektionseingänge) vier (Daten-)Eingänge anwählen und dem Ausgang Q zuführen kann. Es sollen die Signale a, b, c und d auf den Ausgang Q durchgeschaltet werden.

Für die Abhängigkeit des Ausgangssignales von den Steuersignalen gilt folgendes:

$$\begin{array}{ll} S_1 = S_2 = 0 \Rightarrow Q = a \\ S_1 = S_2 = 1 \Rightarrow Q = d \end{array}$$

$$\begin{array}{ll} S_1 = 1; S_2 = 0 \Rightarrow Q = b \\ S_1 = 0; S_2 = 1 \Rightarrow Q = c \end{array}$$

- a) Ergänzen Sie die Eingangsspalten der verkürzten Wertetabelle nach folgender Vorgabe :

Selektions-eingänge		Dateneingänge				Ausgang
S ₂	S ₁	a	b	c	d	Q
0	0	a	X	X	X	a
0	1	X	b	X	X	b
1	0	X	X	c	X	c
1	1	x	X	X	d	d

- b) Erstellen Sie die Teilgleichungen für den Ausgang $Q = f(S_1, S_2, a, b, c, d)$ und zeichnen Sie den Schaltplan.
 c) Erweitern Sie die Schaltung um einen lowaktiven Freigabeeingang /EN (EN = enable). Erklären Sie dessen Funktion in der Schaltung.

Aufgabe 2 : 1-auf-4-Demultiplexer

Entwickeln Sie einen Demultiplexer, der das Eingangssignal E (= „H“) mit den Steuersignalen A und B auf die vier Ausgänge Z₀₋₃ verteilen kann. Der angewählte Ausgang Z geht dann auf „H“, die nichtangewählten bleiben auf L.

- a) Ergänzen Sie die nachfolgende Wertetabelle.
 b) Stellen Sie die DNF für die Ausgänge Z₀₋₃ auf und zeichnen Sie den Schaltplan.

Daten - eingang E	Steuer-signale		Ausgänge			
	A	B	Z ₀	Z ₁	Z ₂	Z ₃
E	0	0	H	L	L	L
E	0	1	L	H	L	L
E	1	0	L	L	H	L
E	1	1	L	L	L	H

$$E \cdot \bar{A} \cdot \bar{B}$$

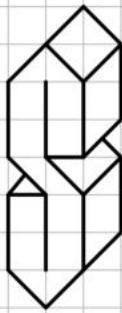
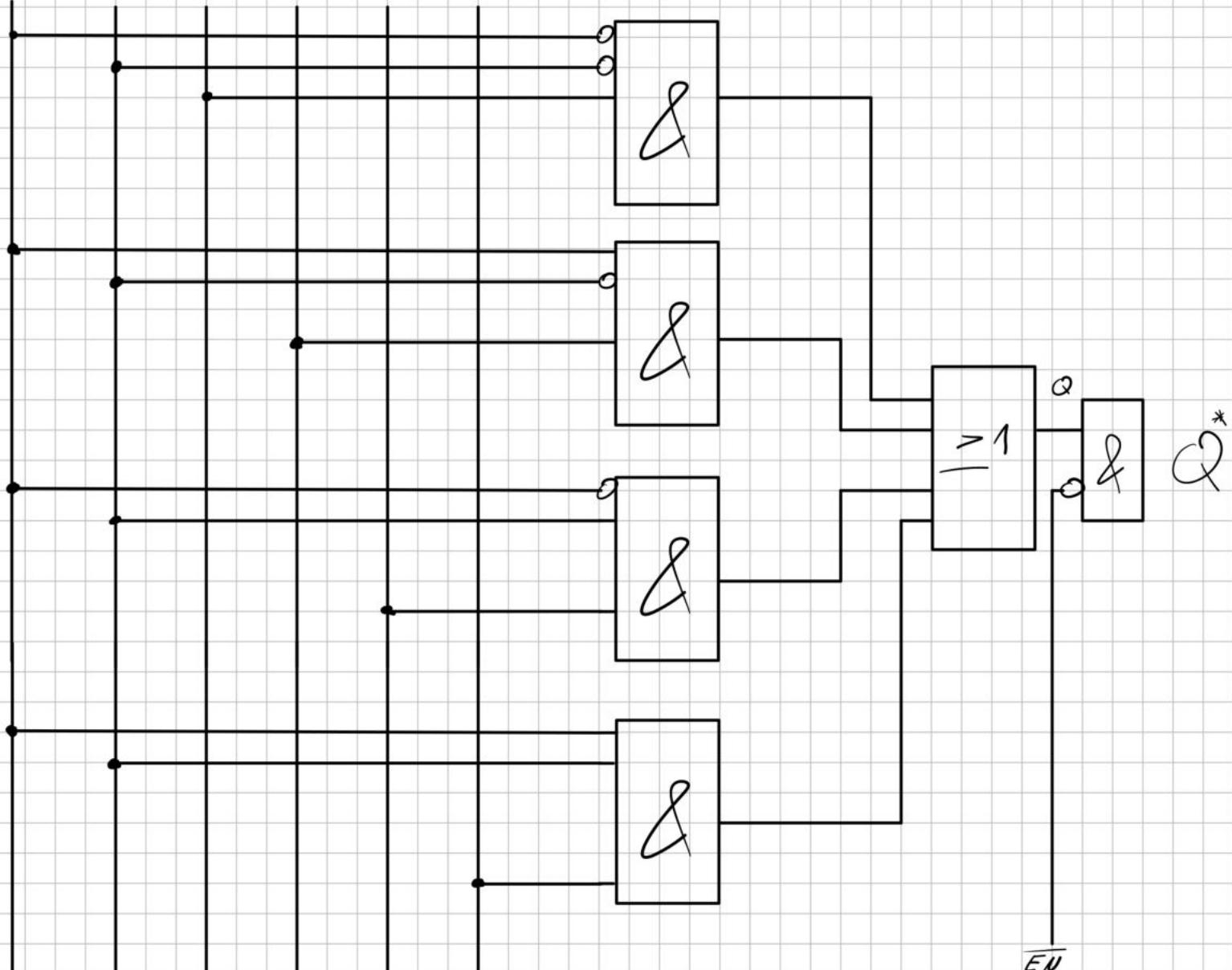
$$E \cdot \bar{A} \cdot B$$

$$E \cdot A \cdot \bar{B}$$

$$E \cdot A \cdot B$$

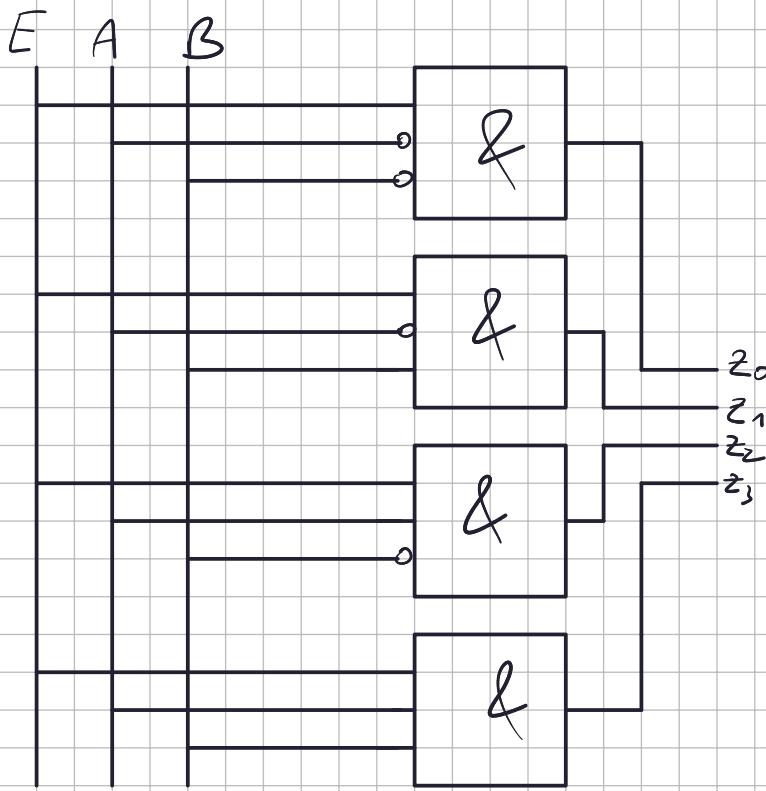
b) $Q = (\bar{s}_1 \wedge \bar{s}_2 \wedge a \wedge \bar{b} \wedge \bar{c} \wedge \bar{d}) \vee$
 $(\bar{s}_1 \wedge \bar{s}_2 \wedge \bar{a} \wedge b \wedge \bar{c} \wedge \bar{d}) \vee$
 $(\bar{s}_1 \wedge s_2 \wedge \bar{a} \wedge \bar{b} \wedge c \wedge \bar{d}) \vee$
 $(\bar{s}_1 \wedge s_2 \wedge \bar{a} \wedge \bar{b} \wedge \bar{c} \wedge d) \vee$

$s_1 \quad s_2 \quad a \quad b \quad c \quad d$



(2)

b) $Z_0 = E_1 \bar{A} \bar{B}$ $Z_1 = E_1 A \bar{B}$ $Z_2 = E_1 A B$ $Z_3 = E_1 \bar{A} B$



Aufgabe 1 : 1-aus-4-Multiplexer (Datenselektor)

Entwickeln Sie eine 4-zu-1- Multiplexer-Schaltung, die mit den Steuersignalen S1 und S2 (Selektionseingänge) vier (Daten-)Eingänge anwählen und dem Ausgang Q zuführen kann. Es sollen die Signale a, b, c und d auf den Ausgang Q durchgeschaltet werden.

Für die Abhängigkeit des Ausgangssignales von den Steuersignalen gilt folgendes:

$$\begin{array}{ll} S_1 = S_2 = 0 \Rightarrow Q = a \\ S_1 = S_2 = 1 \Rightarrow Q = d \end{array}$$

$$\begin{array}{ll} S_1 = 1; S_2 = 0 \Rightarrow Q = b \\ S_1 = 0; S_2 = 1 \Rightarrow Q = c \end{array}$$

- a) Ergänzen Sie die Eingangsspalten der verkürzten Wertetabelle nach folgender Vorgabe :

Selektions- eingänge		Dateneingänge				Ausgang
S ₂	S ₁	a	b	c	d	Q

- b) Erstellen Sie die Teilgleichungen für den Ausgang $Q = f(S_1, S_2, a, b, c, d)$ und zeichnen Sie den Schaltplan.
 c) Erweitern Sie die Schaltung um einen lowaktiven Freigabeeingang /EN (EN = enable). Erklären Sie dessen Funktion in der Schaltung.

Aufgabe 2 : 1-auf-4-Demultiplexer

Entwickeln Sie einen Demultiplexer, der das Eingangssignal E (= „H“) mit den Steuersignalen A und B auf die vier Ausgänge Z₀₋₃ verteilen kann. Der angewählte Ausgang Z geht dann auf „H“, die nichtgewählten bleiben auf L.

- a) Ergänzen Sie die nachfolgende Wertetabelle.
 b) Stellen Sie die DNF für die Ausgänge Z₀₋₃ auf und zeichnen Sie den Schaltplan.

Daten - eingang	Steuer- signale		Ausgänge				
	E	A	B	Z ₀	Z ₁	Z ₂	Z ₃

Multiplexer - Logische Bausteine & Schaltnetze 5 • SIMPLECLUB.DE



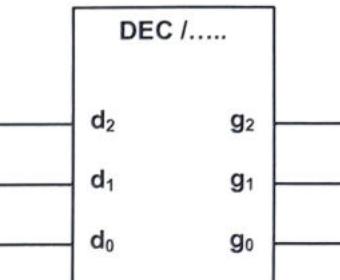
https://www.youtube.com/watch?v=JpnVqaSE5_w

Aufgabe: Codes, Codierung, Codewandler

Bei der binären Zahlendarstellung kommt es vor, dass sich zwei aufeinander folgende Werte in mehreren Bits gleichzeitig unterscheiden. Beispielsweise ändern sich beim Wechsel von dezimal 3 auf 4, also von binär 011 auf 100, alle drei Bits.

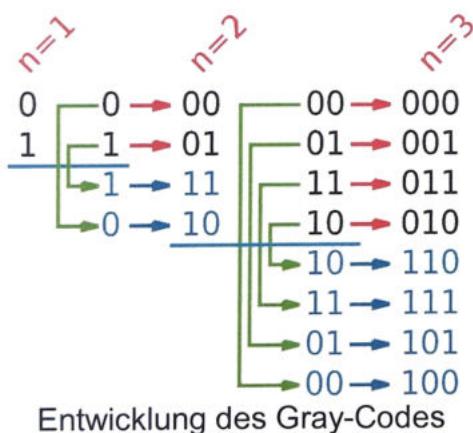
Ziel: Entwerfen / verwenden Sie einen 3-bit-Code, der diesen Nachteil beseitigt, indem sich beim Wechsel zwischen zwei aufeinanderfolgenden Werten jeweils nur ein Bit ändert.

- Wie nennt man allgemein einen Code, bei dem sich jeweils nur ein Bit ändert?
- Welchen Vorteil hat der Gray Code gegenüber dem Binär Code?
- Warum ist der Binärcode ein bewertbarer Code?
- Geben Sie eine Wertetabelle an, die eine 3-Bit-Dezimalzahl $d_2..d_0$ im 4-2-1-Code auf die geforderte 3-Bit-Darstellung $g_2..g_0$ abbildet.
- Minimieren Sie die Funktionen aus d) weitestgehend.

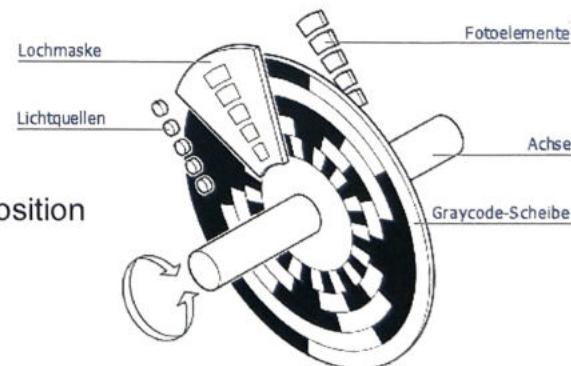


Geschichte

Noch bevor die Bezeichnung Gray-Code eingeführt wurde, gab es bereits mathematische Knobelspiele, in denen das Prinzip angewendet wurde. Erst später fand der Code die Beachtung von Ingenieuren. Der Franzose Jean-Maurice-Émile Baudot verwendete Gray-Codes im Jahr 1887 für die elektrische Telegrafie. Namensgebend war allerdings Frank Gray, Forscher in den Bell Laboratories, der den Code erst 1946 für seine Zwecke wiederentdeckte.



Gray-Code-Scheibe



Anwendungsbeispiel: Bestimmung der absoluten Position zur Winkel- oder Drehgeschwindigkeitsmessung

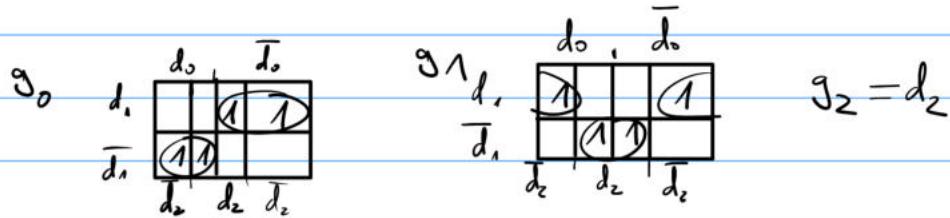
a) Einschrittiger Code

b) ϕ

c) Jede Stelle stellt einen Wert dar.

d)

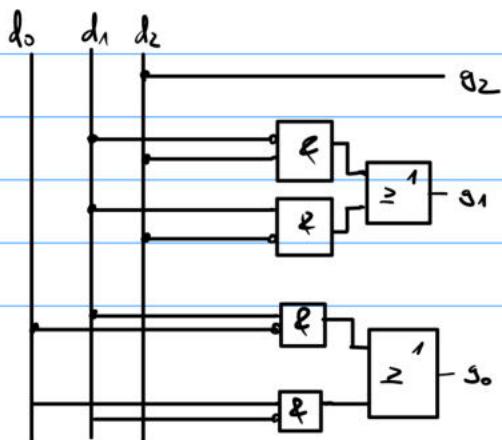
d_2	d_1	d_0	g_2	g_1	g_0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0



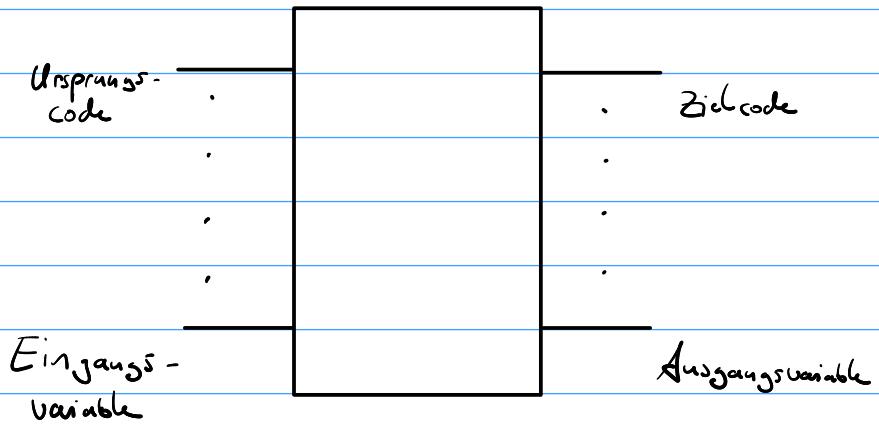
$$g_2 = d_2$$

$$g_1: (\bar{d}_1 \wedge d_2) \vee (d_1 \wedge \bar{d}_2)$$

$$g_0: (d_1 \wedge \bar{d}_0) \vee (\bar{d}_0 \wedge \bar{d}_1)$$



Codewandler

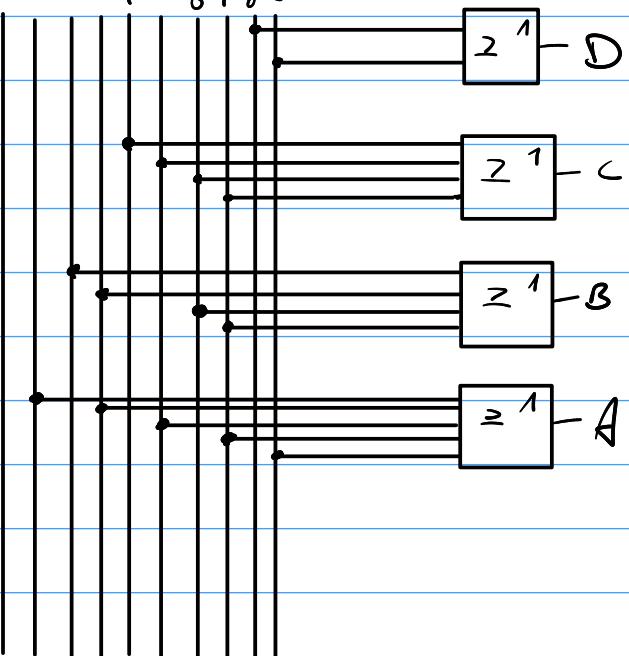


Decimal zu BCD-Wandler

Ursprungscode Dezimalc	Zielcode				
	D _{z³}	C _{z²}	B _{z¹}	A _{z⁰}	
0	0	0	0	0	D = 8 v 9
1	0	0	0	1	C = 4 v 5 v 6 v 7
2	0	0	1	0	B = 2 v 3 v 6 v 7
3	0	0	1	1	A = 1 v 3 v 5 v 7 v 9
4	0	1	0	0	
5	0	1	0	1	
6	0	1	1	0	
7	0	1	1	1	
8	1	0	0	0	
9	1	0	0	1	



0 1 2 3 4 5 6 7 8 9



Von der Ziffer zur Anzeige - Codewandler



Beispiel:

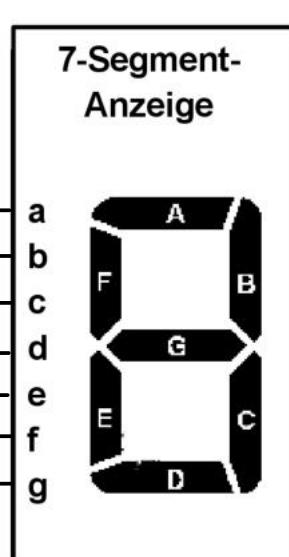
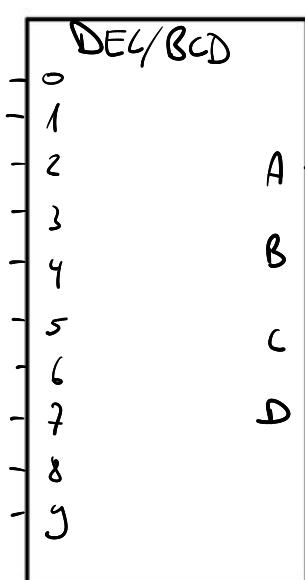
Ziffer 0 zur Anzeige übertragen



Ziffer 0 mit wenigen Leitungen übertragen



Umwandeln / Codieren in andere Darstellung



Dezimal zu BCD-
Wandler

BCD-zu-7 Segment-
Wandler

Siebensegment-
Anzeige

Tetradische Codes

- Mit 4 Bit pro Ziffer lassen sich $2^4 = 16$ Tetraden $= 16$ Ziffern bilden;
- davon müssen 10 Tetraden für die Dezimalziffern ausgewählt werden.
- Nicht benutzte heißen (die 6) Pseudotetraden
- zum Vereinfachen: nicht verw (x)

Tetraden Dualcode					Dezimalcodes		
Nr	d	c	b	a	8-4-2-1-Code (BCD-Code)	Excess-3-Code (Stibitz-Code)	Gray-Code
0	0	0	0	0	0	Pseudo - tetraden	0
1	0	0	0	1	1		1
2	0	0	1	0	2		3
3	0	0	1	1	3	0	2
4	0	1	0	0	4	1	7
5	0	1	0	1	5	2	6
6	0	1	1	0	6	3	4
7	0	1	1	1	7	4	5
8	1	0	0	0	8	5	15
9	1	0	0	1	9	6	14
10	1	0	1	0	Pseudo - tetraden	7	12
11	1	0	1	1		8	13
12	1	1	0	0		9	8
13	1	1	0	1		Pseudo - tetraden	9
14	1	1	1	0			11
15	1	1	1	1			10

Aufgabe : Geben Sie die Dezimalzahl 2579 im jeweiligen Code an.

Dualcode	→	2579 ₁₀ = 101000010011		
Tetradennr. Code	4	3	2	1
Dezimalziffer	2	5	7	9
8-4-2-1-Code	0010	0101	0111	1001
Exzeß-3-Code	0101	1000	1010	1100

Aufgabe : Ordnen Sie jeweils die Dezimalziffern des Codes den Feldern der KV-Tafel zu.

BCD - Code	
a	/a
0 ₃	1 ₇
X ₄	X ₅
1 ₅	X ₆
0 ₁	1 ₅

b /d
d
/b
/c

Exzeß-3 – Code	
a	/a
0 ₃	X ₇
0 ₄	X ₅
6 ₅	X ₆
X ₁	1 ₅

b /d
d
/b
/c

Was ist ein ... ?

Binärcode

Binärcode ist ein Code, bei dem jedes Codewort aus Binärzeichen (0,1 oder auch H,L) besteht.

BCD-Code

Binary Coded Decimal, 4-Bit - dual kodierte Dezimalziffer. Der BCD-Code wird auch als 8-4-2-1-Code bezeichnet.

Ziffernkode

getrennte Codierung jeder Ziffer einer Zahl
Beispiel: BCD-Code

Dualcode

Bildet Zahlen als ganzes Wort im Dualsystem ab.

Wortcode

Codierung einer Zahl als ganzes Wort, nicht ziffernweise
Beispiel: dualcode

Bewertbarer Code

Jede Stelle besitzt eine Wertigkeit (Stellenwert)
Beispiel: 8-4-2-1-Code

Nicht bewertbarer Code

Die Tetradenstellen haben keine Bewertung.
= Anordnungscode - die Position der einzelnen Bits ist von Bedeutung.
Beispiel: Gray-Code

Vollständiger Code

Ein vollständiger Code hat keine Pseudotetraden
Gegen-Beispiel: 8-4-2-1-Code
mögliche Kombinationen,
Codeworte verwendet
=> nicht vollständiger Code!!

Warum benutzt man verschiedenartige Codes?

↪ es gibt doch den Binärkode

$$\begin{aligned}
 4_{10} &\stackrel{!}{=} 100_2 \\
 12_{10} &\stackrel{!}{=} 1100_2 \\
 27_{10} &\stackrel{!}{=} 11011_2 \\
 2023_{10} &\stackrel{!}{=} ???
 \end{aligned}$$

↓ Rechenleistung steigt

Einfacher?

⇒ BCD - Code

größte Dez. Ziffer: $9_{10} \stackrel{!}{=} 1001_2$
4 Stellen

Binary Coded Decimal

2	0	2	3	$_{10}$
0010	0000	0010	0011	

↪ Jede Dezimalzahl wird für sich mit einer 4 stelligen Dualzahl dargestellt.

↪ schnell umgeschaut.

27
0010 0111 BCD

Wie groß ist ein Gigabyte?

Technikärgernis Maßeinheiten → Warum Festplatten plötzlich schrumpfen

Jede Woche wieder entdeckt irgendwo in Deutschland ein Computernutzer, dass ihm Unbekannte eine Menge Gigabytes gestohlen haben. Da kauft sich jemand eine "Festplatte mit 1000 GB" und stellt fest: "Nun an Windows angestöpselt - siehe da: 910 GB!"

Woher stammen diese abweichenden Größenangaben?

Die großen Abweichungen bei den Angaben zum Speicherplatz röhren daher, dass Festplattenhersteller und Betriebssysteme mit zwei unterschiedlichen Standards rechnen.

Zweier- oder Zehnerpotenzen?

Alle Einheiten des in der Informatik genutzten Binärsystems sind Potenzen von Zwei.

Der Rest der Welt rechnet aber etwas anders. Ein Kilometer hat ja nicht 1024, sondern 1000 Meter. Entfernungen usw. geben zumindest Europäer mit Maßeinheiten an, die sich in Zehnerpotenzen steigern.

Wenn Informatiker also von einem Kilobyte sprechen und damit 1024 Bytes meinen, weichen sie von einem internationalen Standard ab - um 2,4 Prozent. Dieses Kuddelmuddel wollte das Normungsgremium für Elektrotechnik IEC im Jahr 2000 beenden. Wer das alte Binärsystem benutzt, soll doch bitte fortan von z.B. Gibibytes sprechen und als Abkürzung auch GiB statt GB verwenden, damit die Verwirrung aufhört.

Einheiten Bit	
kBit	1000 Bit
MBit	1000 kBit
GBit	1000 MBit
TBit	1000 GBit
PBit	1000 TBit

Einheiten Byte (dezimal)	
Kilobyte (kB)	1000 Byte
Megabyte (MB)	1.000.000 Byte
Gigabyte (GB)	1.000.000.000 Byte
Terabyte (TB)	1.000.000.000.000 Byte
Petabyte (PB)	1.000.000.000.000.000 Byte

Einheiten Byte (binär)	
Kibibyte (KiB)	1024 Byte
Mebibyte (MiB)	1.048.576 Byte
Gibibyte (GiB)	1.073.741.824 Byte
Tebibyte (TiB)	1.099.511.627.776 Byte
Pebibyte (PiB)	1.125.899.906.842.624 Byte

Nur leider ignorieren die beiden Anwenderbetriebssysteme Windows und MacOS diesen Standard komplett und beziffern Datenmengen im guten alten Binärformat, ohne das kenntlich zu machen. Wenn Windows und Mac behaupten, "GB" zu messen, meinen sie eigentlich "GiB". Die Hersteller von Festplatten, Speicherkarten und USB-Sticks hingegen nutzen ausschließlich den Zehnerstandard.

Es gilt also z.B.:

$$256 \text{ kiByte} = 2^{8*2^{10}*2^3} \text{ Bit} = 2^{21} \text{ Bit} = 2.097.152 \text{ Bit}$$

$$256 \text{ kByte} = 2^{8*10^3*2^3} \text{ Bit} = 1000 * 2^{11} \text{ Bit} = 2.048.000 \text{ Bit}$$

SI- und IT-Präfixe im Vergleich

Präfix (SI)	Symbol	Bedeutung (SI)	Bedeutung (IT)	Differenz (Prozent)	Differenz (absolut)
Kilo	k	10^3	2^{10}	2,4	24
mega	M	10^6	2^{20}	4,9	48.567
giga	G	10^9	2^{30}	7,4	73.741.824
tera	T	10^{12}	2^{40}	10,0	~100 Milliarden
peta	P	10^{15}	2^{50}	12,6	~126 Billionen
exa	E	10^{18}	2^{60}	15,3	~153 Billiarden



<https://www.itwissen.info/Praefix-prefix.html>

<https://de.wikipedia.org/wiki/Bin%C3%A4rpr%C3%A4fix>



Aufgabe: Kapazitätsberechnung

Binärpräfixe sind Vorsätze für Maßeinheiten (Einheitsvorsätze), die Vielfache bestimmter Zweierpotenzen bezeichnen.

SI-Präfixe sind **Dezimalpräfixe**, die Vielfache bestimmter Zehnerpotenzen bezeichnen.

Als Einheitsvorsatz wird sowohl der Name als auch das zugehörige Symbol bezeichnet.

Aus: <http://de.wikipedia.org/wiki/Bin%C3%A4rpr%C3%A4fix>

- Berechnen Sie die Speicherkapazität der Festplatte in Bit.
- Welche Kapazität in GiB zeigt das Betriebssystem an?
- Um wieviel % ist die aufgedruckte Angabe auf der SSD-Festplatte zu groß?



Auf einer Festplatte wird die Kapazität immer in GB angegeben.

Das Betriebssystem rechnet jedoch mit GiB.

$$a) 128 \text{ GB} = 128 \cdot 1000 \cdot 1000 \cdot 1000 \cdot \underbrace{8 \text{ bit}}_{2^3 \cdot 2^3} = 1024 \cdot 10^9 \text{ bit}$$

$$1,024 \cdot 10^9 \text{ bit}$$

$$b) \text{GiB}$$

$$1024 \cdot 10^9 \cdot 8 \cdot 2^{30} \approx 115,21 \text{ GiB}$$

$$128 \text{ GB} : 1.073.741.824 \text{ Byte} \approx 115,21 \text{ GiB}$$

$$c) \text{Zu groß, d.h. } 115,21 \text{ GiB} = 100\%$$

$$(128 : 115,21) \cdot 100 = 107,37\%$$

$\approx 7,4\%$ zu große Angabe.

Hierbei kommt es nicht auf die Größe an, solange diese in GByte angegeben ist, denn für diese Einheit gilt generell:

$$10^9 : 2^{30} - 1 == 7,4 \%$$

Bei xTB müssten wir rechnen: $10^{12} : 2^{40} - 1 == 9,95\%$
Wiederum ist dies bei allen Festplatten im Terabyte Bereich der Fall.

Aufgabe 1: Rechnen in Zahlensysteme

Addieren Sie direkt, d.h. ohne Umformen in ein anderes Zahlensystem im

Dualsystem : $111001_2 + 11110_2 = ?$

Hexadezimalsystem : $94_{16} + 85_{16} = ?$
 $32F_{16} + DAB_{16} = ?$

....
41ABh
....
....
00FFh
....

Aufgabe 2: Adressierung

Bei einem Speicherbereich sind Anfangsadresse 00FFh und Endadresse 41ABh der belegten Speicherblöcke gegeben.
 Ermitteln Sie die Anzahl der Speicherblöcke (Hex. und Dez.).

Aufgabe 3: IP-Adressen nach Version 4 (IPv4)

aus: <http://www.tinohempel.de/info/info/netze/ip.htm>

Die Datenkommunikation im Internet funktioniert auf Grundlage eindeutiger IP-Adressen.



Aufbau der IP-Adresse

Die IP-Adresse ist eine 32 bit große Binärzahl. Zur besseren Lesbarkeit wird die Zahl in vier gleich große Blöcke zu je 8 Bit zerlegt. Die acht Bits werden zur Dezimalzahl zusammengefasst. Damit ergibt sich daraus nach Einfügen des Punkt-Trennzeichens die Dezimal-Punkt-Notation.

Beispiel: 00000001 00000000 00000010 00000010 → 1.0.2.2

a) Wie viele Adressen sind mit einer 32bit großen IP-Adresse darstellbar?

b) Rechnen Sie die Binäradressierung 01010101 00001101 10000100 10011010 in die übliche Dezimal-Punkt-Notation um.

Wird die Subnetzmaske mit der vorhandenen IP-Adresse logisch UND-verknüpft, so erhält man die Netzadresse.

Beispiel: IP-Adresse 195.13.132.163, Subnetzmaske 255.255.255.224.

c) Berechnen Sie die Netzadresse:

IP-Adresse	195.013.132.163	11000011. 00001101. 1000100. 1010011
<u>UND</u> Netzmakske	255.255.255.224	1111111. 1111111. 1111111. 1110000
= Netzadresse	195.013.132.160	11000011. 00001101. 1000100. 1010000

(1)

$$\begin{array}{r} 111001_2 \\ 101110_2 \\ \hline 1010111_2 \end{array}$$

$$\begin{array}{r} 94_{16} \\ 85_{16} \\ \hline 119_{16} \end{array}$$

$$\begin{array}{r} 32F_{16} \\ DAB_{16} \\ \hline 10DA \end{array}$$

(2)

$$\begin{array}{r} 60FF \\ - 41AB \\ \hline 18F54 \end{array}$$

(3)

a) $2^{32} = 4.294.967.296$ Adressen

b) 85. 13. 132. 154 \Rightarrow jedes Block einzeln

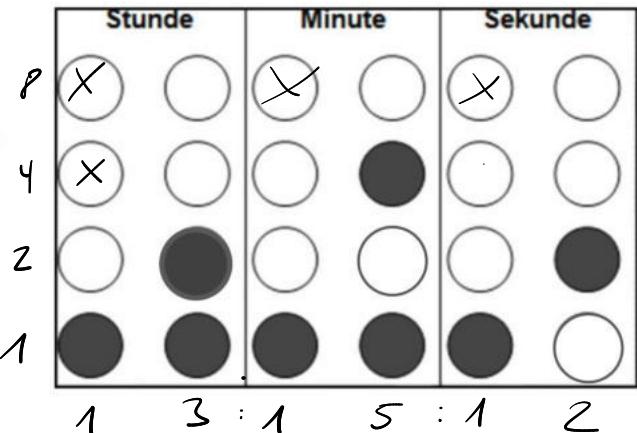
c)

Aufgabe 4: Binäre Uhren

Binäre Uhren gibt es in vielfältigen Ausführungen. Hier werden zwei Darstellungsarten vorgestellt, eine binäre Aufstelluhr und eine binäre Armbanduhr. Die Anzeige erfolgt jeweils mit LEDs.

Bild 1: Prinzip einer Aufstelluhr, bei der Stunden, Minuten und Sekunden in Einer- und Zehnerstellen getrennt dargestellt werden.

- Wie spät ist es auf der Uhr? $13:15:12$
- Für die Darstellung der Uhrzeit im 24h-Mode nach Bild 1 sind nicht alle eingezeichneten LEDs notwendig.



Kennzeichnen Sie die LEDs, die nie leuchten. Begründen Sie Ihre Wahl.

Bild 2: Armbanduhr. Aus Platzmangel werden oft nur Stunden und Minuten in zwei Zeilen angezeigt.

- Erklären Sie die Zeichen auf dem äußeren Ring der Armbanduhr. Stunden - Ziffer
- Erklären Sie die zweizeilige Angabe im Zentrum der Uhr. Zeitangabe in Binär-Code oben Stunden unten Minuten
- Wie spät ist es auf der Armbanduhr? $12:38\text{Uhr}$



Aufgabe 5:

Subtraktionen werden mittels des Zweierkomplements in Additionen zurückgeführt.

- Welcher Grundbaustein bildet das (Einer-)Komplement?
- Berechnen Sie mit Hilfe des Zweierkomplements in einem 8-bit-System:

$$33_{\text{dez}} - 18_{\text{dez}}$$

$$7_{\text{dez}} - 18_{\text{dez}}$$

in einem zehnstelligen System:

$$-38_{10} - 42_{10}$$

5) a) Invert, Negation, Nicht

b)

$$\begin{array}{r} 100001 \\ + 001111 \\ \hline 1101111 = 15 \end{array}$$

$$\begin{array}{r} 000111 \\ + 001111 \\ \hline 010100 \end{array}$$

positive

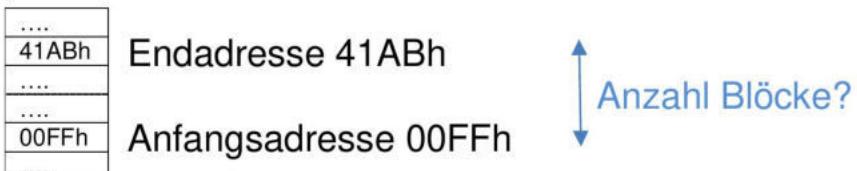
10110000

Aufgabe 1: Rechnen in Zahlensysteme

Dualsystem : $111001_2 + 11110_2 = \textcolor{red}{1010111}$

Hexadezimalsystem : $94_{16} + 85_{16} = \textcolor{red}{118}$
 $32F_{16} + DAB_{16} = \textcolor{red}{10DA}$

Aufgabe 2: Adressierung



Lösungshinweis:

4
3
2
+ 1

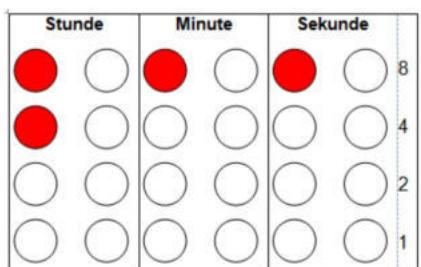
Endadresse 4
- Anfangsadresse 2
= Speicherblöcke 2
+ 1 3

$$\begin{array}{r}
 41AB \\
 - 00FF \\
 \hline
 11 \\
 = 40AC_{\text{hex}} + 1 = 40AD_{\text{hex}}
 \end{array}$$

Speicherblöcke
 $= 4 \cdot 16^3 + 0 \cdot 16^2 + 10 \cdot 16^1 + 12 \cdot 16^0 + 1 = 16557_{\text{dez}}$

Aufgabe 4: Binäre Uhren

13:15:12



Stunden-Zehner wird nur 0,1 und 2 (2 bit) benötigt.

Minute und Sekunde geht nur bis je 5_{dez} , also nur 3 bit in Zehnerstelle.

Symbolischer Stundenring 1-12 in BCD

Zeilen entsprechen Stunden und Minuten 12:38

Aufgabe 5:

a) Inverter

b) Berechnen Sie mit Hilfe des Zweierkomplements in einem 8-bit-System:

$33_{\text{dez}} - 18_{\text{dez}}$

$7_{\text{dez}} - 18_{\text{dez}}$

in einem zehnstelligen System: $-38_{10} - 42_{10}$

Code =

Vorschrift, wie die ursprünglichen Zeichen einem Zeichensystem zugeordnet werden.

Objektmenge

eindeutige Zuordnung
(= Codierung)

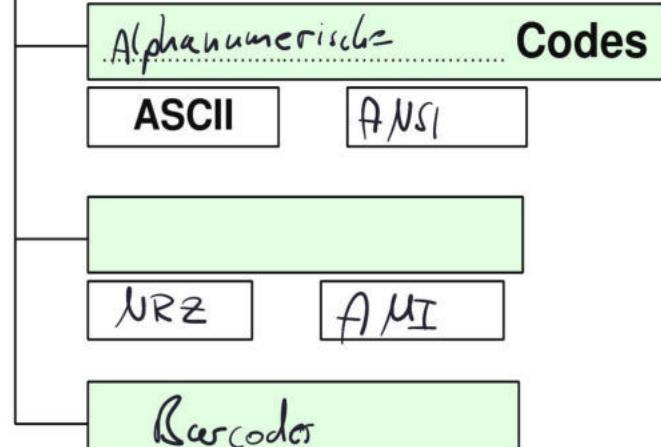
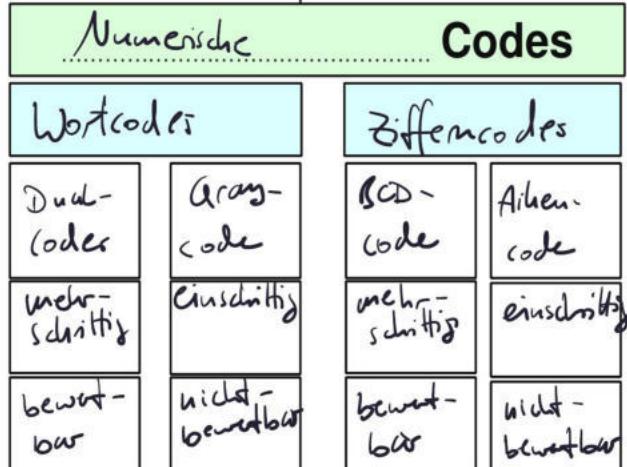
Bildmenge
Codewort

⇒ Die Codierung ändert nur die Darstellung einer Nachricht, nicht ihre Bedeutung!

Die Codierung dient u.a.:

- dem ...zweckmäßigen... und ...einfachen... Übertragen und Verarbeiten... von Informationen,
- zur Darstellung der Informationen mit einer möglichst ...geringen... Anzahl von Zeichen
- dem ...Sichern... von Informationen ...gegen Übertragungsfehler...
- der ...Geheimhaltung... von Informationen.

Code - Arten



9 783540 345329



Aufgabe: 4-bit-Addierer

- Mit Volladdierern lassen sich n-stellige Dualzahlen addieren.
- Je Stelle wird ein Volladdierer benötigt.
- Da an der niedrigwertigsten Stelle eingesangsseitig kein Übertrag aufkommen kann, könnte man für diese Stelle einen Halbbaddierer verwenden. Dies wird in der Praxis jedoch nicht gemacht. Man beschaltet statt dessen den Übertragungseingang hier fest mit der logischen 0.

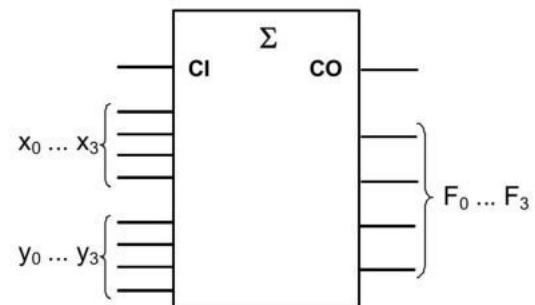


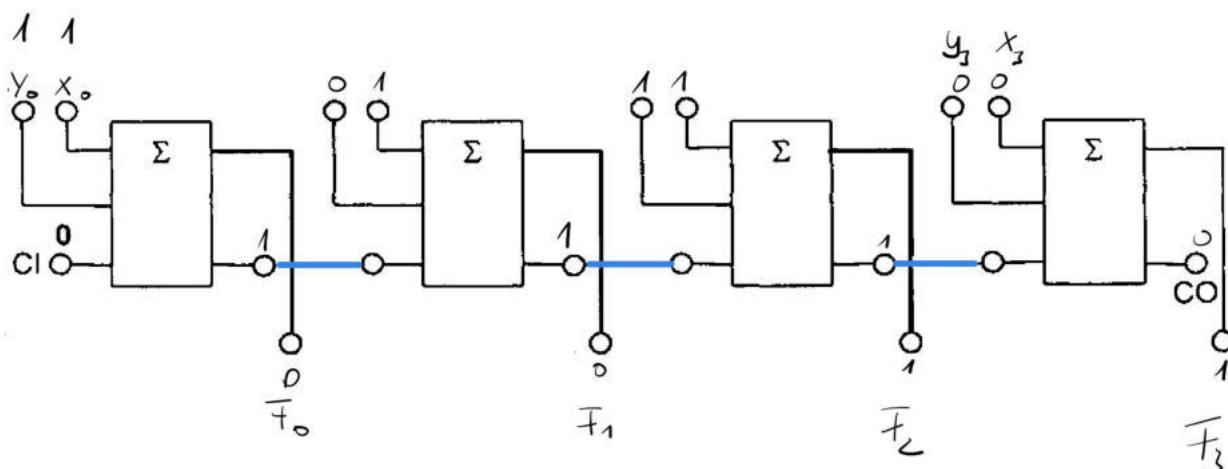
Abb.1: Blockschaltbild eines 4-bit-Addierers

Die Zahlen 7_{Dez} und 5_{Dez} sollen nach unten gegebener Schaltung addiert werden. Hierzu wird am Eingang CI des 4stelligen Addierers fest eine logische 0 angelegt.

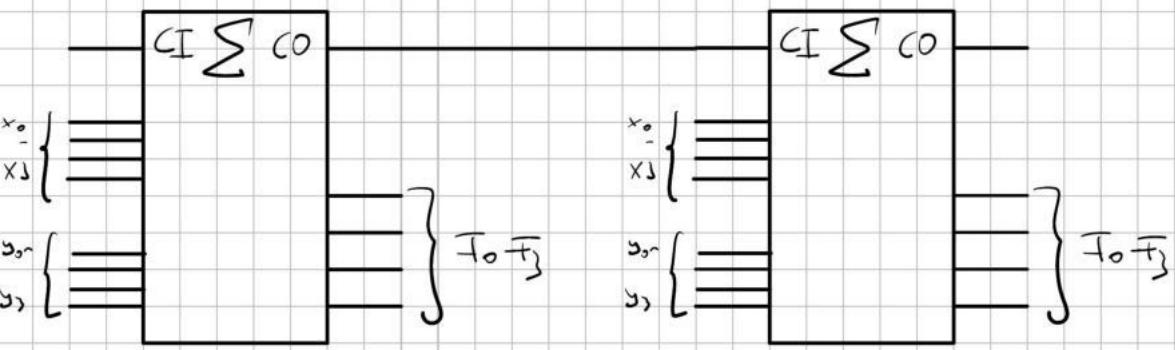
- a) Vervollständigen Sie die nachfolgende Schaltung zur Addition zweier 4-Bit-Zahlen und führen Sie die Kontrollrechnung aus.

Kontrollrechnung:

$$\begin{array}{r}
 \text{Summand 1 :} & \underline{0 \ 1 \ 1 \ 1} & (7_{10}) \\
 \text{Summand 2 : +} & \underline{0 \ 1 \ 0 \ 1} & (5_{10}) \\
 \text{Übertrag :} & \underline{1 \ 1 \ 1} & \leftarrow \text{Ovflaw} \\
 \hline
 \text{Summe :} & \underline{1 \ 1 \ 0 \ 0_2} & = 12_{10} \checkmark
 \end{array}$$



- b) Was bedeutet nach einer Addition zweier Zahlen eine „1“ im Übertragsregister? $\rightarrow \text{Ovflaw}$
- c) Mit 4-bit-Addierern nach Abb.1 sollen zwei 8-bit-Zahlen addiert werden. Zeichnen Sie das Blockschaltbild. \rightarrow wie 2
- d) Welche Werte stehen an den Ausgängen F_0 und F_4 , wenn die Zahlen AB_{hex} und 48_{hex} addiert werden sollen?



$A \quad B$
 $\downarrow \quad \downarrow$
 $1010 \quad 1011$

$$\begin{array}{r}
 1010 \\
 + 0100 \\
 \hline
 1110
 \end{array}
 \qquad
 \begin{array}{r}
 10101011 \\
 + 01000000 \\
 \hline
 10101011
 \end{array}$$

$$\begin{array}{l}
 F_0 = 1 \\
 F_4 = 1
 \end{array}$$

Entwicklung einer Schaltung zur Addition zweier einstelliger Dualzahlen.

- Orientierung an schriftlicher Addition

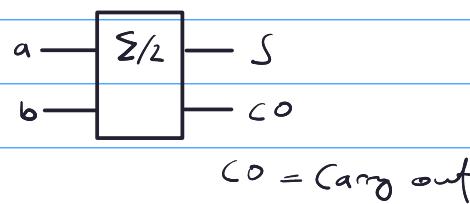
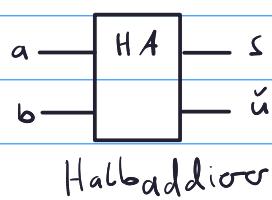
b	a	s	\bar{u}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$\begin{array}{r}
 4 \\
 + 3 \\
 \hline
 7
 \end{array}
 \quad
 \begin{array}{r}
 4 \\
 + 8 \\
 \hline
 12
 \end{array}$$

Übertrag

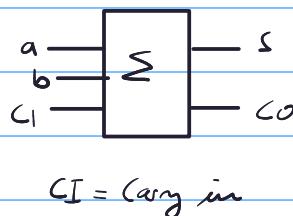
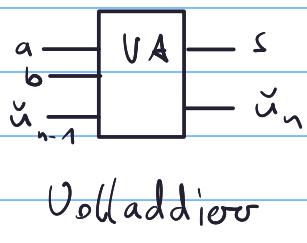
Übertrag \rightarrow muss beim nächsten Bit berücksichtigt werden!

Schaltbild - Addierer:



- Mit einem Halbaddierer lassen sich nur 2 bit addieren, ein Übertrag kann nicht verarbeitet werden.

- Mehrstellige Dualzahlen erfordern 3 Eingänge.



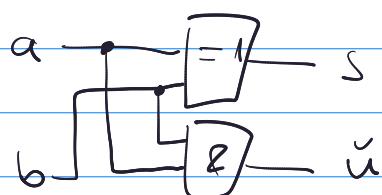
Für eine vollständige Additionschaltung mehrstelliger Dualzahlen braucht man für jede Stelle einen Volladdierer.

Aufgabe

- a) Zeichnen Sie die Schaltung eines Halbaddiers aus einzelnen Gattern!
- b) Wie viele Binärziffern muss ein Volladdierer addieren können?
- c) Erstellen Sie den Schaltplan eines Volladdierers!

b	a	s	\bar{u}
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

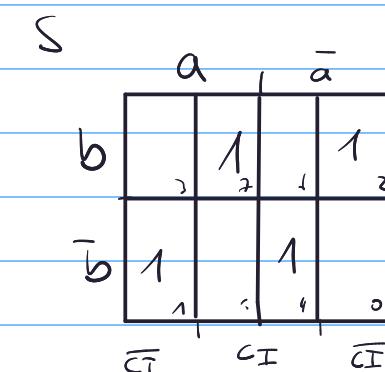
\uparrow \uparrow
XOR UND



$$\begin{array}{r}
 & 0 & 1 \\
 + & 0 & 1 \\
 \hline
 & 1 & 0
 \end{array}$$

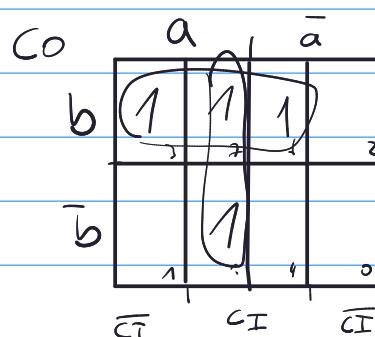
b)

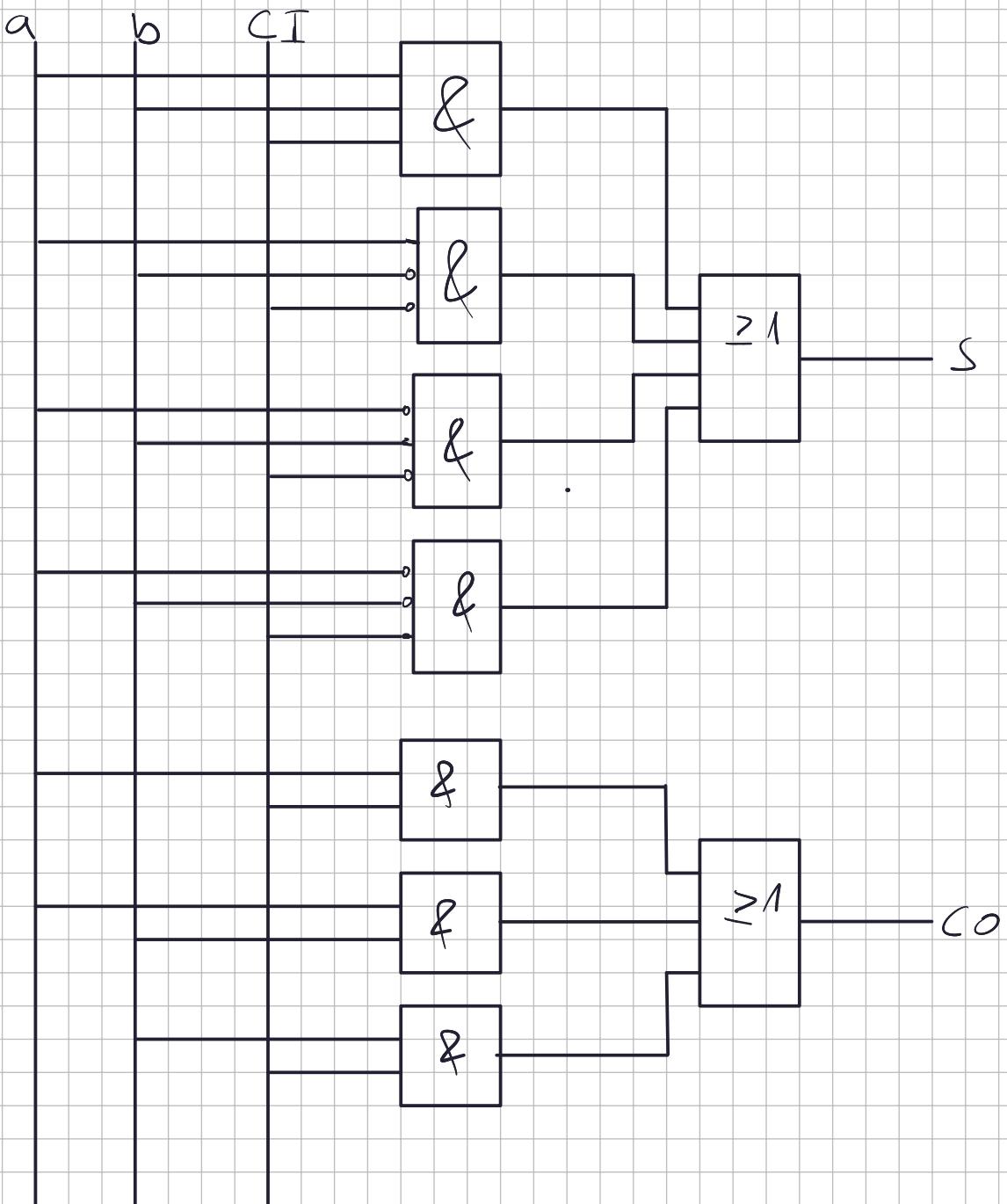
	C_I	b	a	s	C_O
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

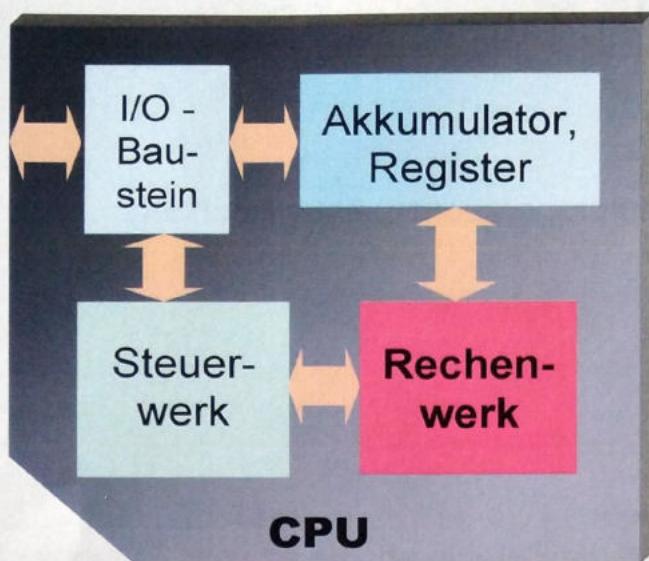


$$\begin{aligned}
 s &= (a \wedge b \wedge C_I) \vee (a \wedge \bar{b} \wedge \bar{C}_I) \vee (\bar{a} \wedge b \wedge \bar{C}_I) \vee \\
 &\quad (\bar{a} \wedge \bar{b} \wedge C_I)
 \end{aligned}$$

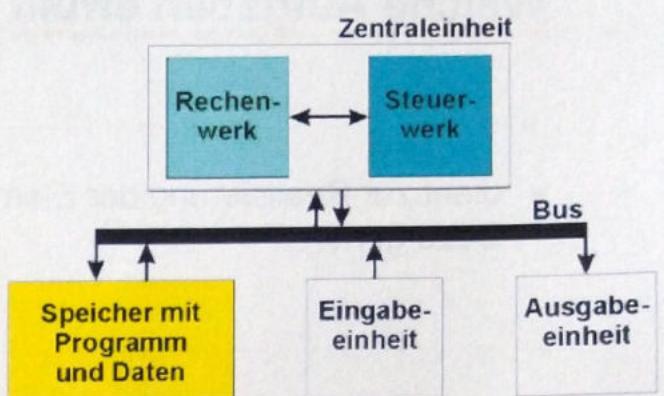
$$C_O = (a \wedge C_I) \vee (a \wedge b) \vee (b \wedge C_I)$$



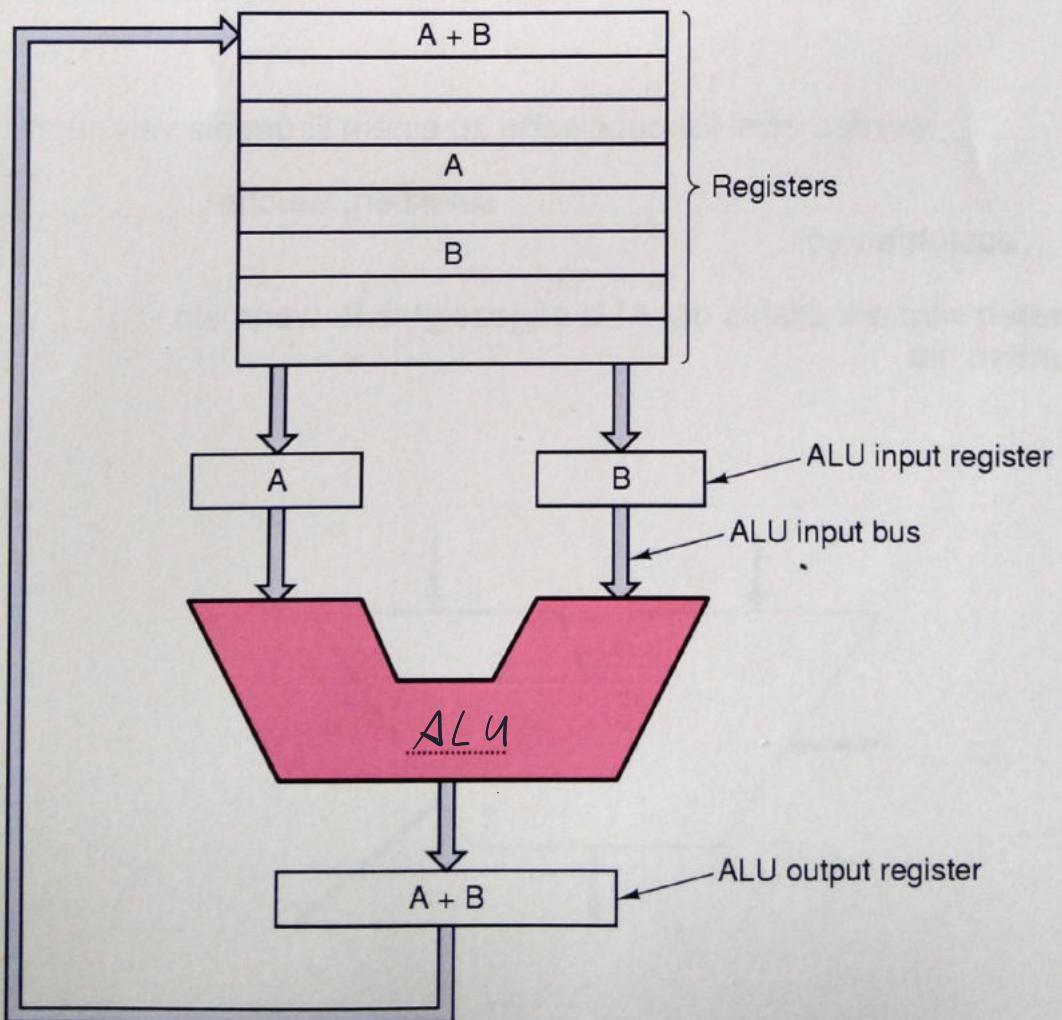




Von – Neumann – Architektur :



Dem Rechenwerk stehen verschiedene Register (z.B. Akkumulator) zur Verfügung, mit denen vor Ausführung einer Operation die Operanden gespeichert werden.



Kern des
Rechenwerkes
→ ALU

Ein Rechenwerk kann höchstens ALU enthalten.
Die ALU ist ein reines Schaltwerk

Welche Aufgaben erfüllt der Kern einer CPU ?

ALU - Arithmetic Logic Unit / Arithmetisch Logische Einheit

- dient zur Realisierung der Elementaroperationen eines Rechners.
- Dazu gehören

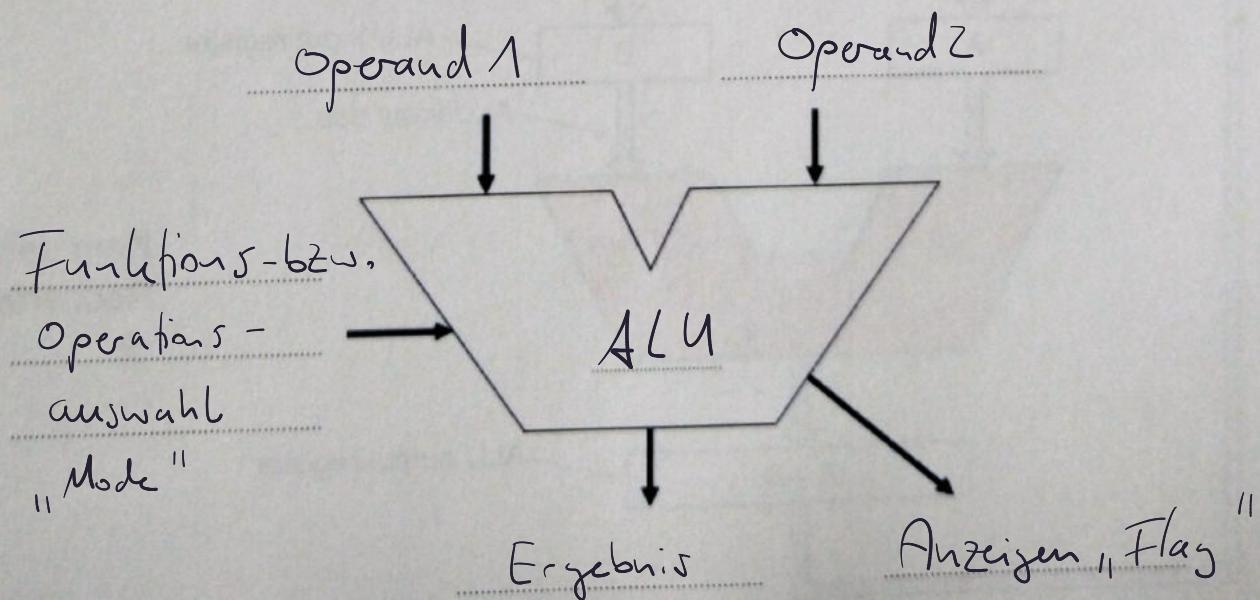
Arithmetische (Rechen-) Operation : +, -, ×, ÷

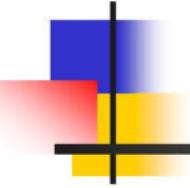
Logische Operatoren : \wedge , \vee , \neg

Darüber hinaus kann das Rechenwerk Vergleiche und Schiebebefehle ausführen. Das Rechenwerk arbeitet die Befehle ab, die es vom Steuerwerk erhält.

→ Alle Rechenoperationen lassen sich auf Addition zurückführen

- In einer ALU werden zwei Eingabewerte zu einem Ergebnis verknüpft.
- Die "Funktions- bzw. Operationswahl" selektiert, welche Funktion die ALU ausführen soll.
- Des weiteren wird der Status der ALU angezeigt, z.B. wenn ein Überlauf stattgefunden hat





Zahlensysteme

Rechnen mit dem Zweierkomplement



Rechnen mit dem Zweierkomplement

Aufgabe: 29 - 17

$$29 = | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$17 = | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

6-stelliges System

$$\text{abzuziehende Zahl :} | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

$$\begin{array}{r} \text{invertierte Zahl} \\ + \\ \hline \end{array} | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} |$$

$$\text{2erKomplement :} | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{1} |$$

$$29 = | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$\text{2erKomplem.:} + | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{1} |$$

$$\text{Übertrag} | \underline{1} | \underline{1} | \underline{1} | \underline{1} | \underline{1} | \underline{1} |$$

$$\text{Ergebnis:} | \underline{1} | \underline{0} | \underline{0} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$= \underline{\underline{1}} \underline{\underline{2}} \text{ Dez}$$

Rechnen mit dem Zweierkomplement

Aufgabe: 29 - 17

$$29 = | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$17 = | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

6-stelliges System

$$\begin{array}{l} \text{abzuziehende Zahl} : | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \end{array}$$

$$\begin{array}{l} \text{invertierte Zahl} : | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \\ + \quad \quad \quad \quad \quad \quad \quad \quad 1 \end{array}$$

$$\text{2erKomplement} : | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{1} |$$

$$29 = | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$\text{2erKomplem.:} + | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{1} |$$

$$\text{Übertrag} : | \underline{1} | \underline{1} | \underline{1} | \underline{1} | \underline{1} | \dots |$$

$$\text{Ergebnis:} | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{0} | \underline{0} |$$

$$= \underline{\underline{12}} \text{ Dez}$$

Rechnen mit dem Zweierkomplement

Aufgabe: 29 - 17

$$29 = | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$17 = | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

6-stelliges System

$$\begin{array}{l} \text{abzuziehende Zahl} : | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \end{array}$$

$$\begin{array}{l} \text{invertierte Zahl} : | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \\ + \quad \quad \quad \quad \quad \quad \quad \quad 1 \end{array}$$

$$\text{2erKomplement} : | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{1} |$$

$$29 = | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$\text{2erKomplem.:} + | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{1} |$$

$$\text{Übertrag} : | \underline{1} | \underline{1} | \underline{1} | \underline{1} | \underline{1} | \dots |$$

$$\text{Ergebnis:} | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{0} | \underline{0} |$$

$$= \underline{\underline{12}} \text{ Dez}$$

Rechen mit dem Zweierkomplement

Rechnen mit dem Zweierkomplement

Aufgabe: 29 - 17

8-stelliges System

$$29 = | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

abzuziehende Zahl : | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

$$17 = | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

invertierte Zahl : | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

+ 1

2erKomplement : | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

29 = | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

2erKomplem.: + | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

Übertrag | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Ergebnis: | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |

= 12 Dez

Rechnen mit dem Zweierkomplement

Aufgabe: 29 - 17

8-stelliges System

$$29 = | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

abzuziehende Zahl : | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

$$17 = | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

invertierte Zahl : | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

+ 1

2erKomplement : | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

29 = | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |

2erKomplem.: + | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

Übertrag | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

Ergebnis: | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |

= 12 Dez

Rechnen mit dem Zweierkomplement

Aufgabe: 17 - 29

$$29 = | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$17 = | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

$$\text{Ergebnis: } = -12_{\text{Dez}}$$

6-stelliges System

Subtrahend:

$$| \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$\begin{array}{r} \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ + \end{array} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{0} |$$

$$1$$

2erKomplement:

$$| \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{1} |$$

$$17 =$$

$$| \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

$$2\text{erKomplem.: } +$$

$$| \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{1} |$$

Übertrag

$$| \dots | \dots | \dots | \underline{1} | \underline{1} | \dots |$$

Ergebnis:

$$| \underline{1} | \underline{1} | \underline{0} | \underline{1} | \underline{0} | \underline{0} |$$

Kein Overflow

Rechnen mit dem Zweierkomplement

Aufgabe: 17 - 29

$$29 = | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$17 = | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

$$\text{Ergebnis: } = -12_{\text{Dez}}$$

6-stelliges System

Subtrahend:

$$| \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$\begin{array}{r} \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ \downarrow \\ + \end{array} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{0} |$$

$$1$$

2erKomplement:

$$| \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{1} |$$

$$17 =$$

$$| \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

$$2\text{erKomplem.: } +$$

$$| \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{1} |$$

Übertrag

$$| \dots | \dots | \dots | \underline{1} | \underline{1} | \dots |$$

Ergebnis:

$$| \underline{1} | \underline{1} | \underline{0} | \underline{1} | \underline{0} | \underline{0} |$$

$\neq -12_{\text{Dez}}$??

Rechnen mit dem Zweierkomplement

Aufgabe: 17 - 29

$$29 = | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

$$17 = | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

Ergebnis: $= -12_{\text{Dez}}$

6-stelliges System

Subtrahend:

$$| \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} |$$

↓ ↓ ↓ ↓ ↓ ↓

invertierte Zahl:

$$| \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{0} |$$

+ 1

2erKomplement:

$$| \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{1} |$$

$$17 =$$

$$| \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} |$$

2erKomplem.:

$$+ | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{1} |$$

Übertrag

$$| \dots | \dots | \dots | \underline{1} | \underline{1} | \dots |$$

Ergebnis:

$$| \underline{1} | \underline{1} | \underline{0} | \underline{1} | \underline{0} | \underline{0} |$$

? $\neq -12_{\text{Dez}}$??

1 → Negative Zahl → löst Rückkomplementieren aus

Rechnen mit dem Zweierkomplement

Aufgabe: 17 - 29

6-stelliges System

Ergebnis:

$$| \underline{1} | \underline{1} | \underline{0} | \underline{1} | \underline{0} | \underline{0} |$$

Rückkomplementierung:

$$| \underline{0} | \underline{0} | \underline{1} | \underline{0} | \underline{1} | \underline{1} |$$

$$+$$

$$1$$

Ergebnis:

$$| \underline{0} | \underline{0} | \underline{1} | \underline{1} | \underline{0} | \underline{0} |$$

$$= -12_{\text{Dez}} \quad \checkmark$$

Rechnen mit dem Zweierkomplement

Aufgabe: -29 - 17

8-stelliges System

$$29 = | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} | \quad \text{2erKomplement: } | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{1} |$$

$$17 = | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{0} | \underline{1} | \quad \text{2erKomplement: } | \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{1} |$$

Ergebnis: $| \underline{1} | \underline{1} | \underline{1} | \underline{0} | \underline{1} | \underline{0} | \underline{0} | \underline{1} | \underline{0} |$

Rückkomplementierung: $| \underline{0} | \underline{0} | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{0} | \underline{1} | + 1$

Ergebnis: $| \underline{0} | \underline{0} | \underline{1} | \underline{0} | \underline{1} | \underline{1} | \underline{1} | \underline{0} | = -46_{\text{DEZ}}$



FIN

Subtraktion durch Addition des Komplements

Dezimalsystem:

Bsp.:

$$\begin{array}{r}
 1000 \quad \text{Minuend} \\
 - 046 \quad \text{Subtrahend} \\
 \hline
 954 \quad \text{Differenz}
 \end{array}
 \quad
 \begin{array}{r}
 1000 \\
 + 954 \\
 \hline
 1954
 \end{array}$$

954 → 954 ist das Zehnerkomplement zur Zahl 46 (des Subtrahenden -46).

Im Dezimalsystem ergänzen sich Komplement und abzuziehende Zahl bei n-stelliger Darstellung zu 10^n .

Das gilt ebenso im **Dualsystem**:

Bsp.: n = 3 Stellen

$$\begin{array}{r}
 000 \\
 - 101 \\
 \hline
 011
 \end{array}
 \quad
 \xrightarrow{\text{Zwischkomplement}}
 \quad
 \begin{array}{r}
 1000 \\
 + 011 \\
 \hline
 1011
 \end{array}$$

011 → Zwischkomplement zu 101 (hier $101 + 011 = 1000$)

Darstellung negativer Dualzahlen im Zwischkomplement

→ Wie bildet man das Zwischkomplement ?

$$\begin{array}{r}
 0 \\
 - 3 \\
 \hline
 - 3
 \end{array}
 \quad
 \begin{array}{r}
 000 \\
 - 011 \\
 \hline
 011
 \end{array}$$

101 ist eigentlich 5

$$\begin{array}{r}
 + 3 \quad 011 \\
 - 3 \quad 101
 \end{array}
 \quad
 = \text{Invertierte Subtrahend} + 1 = \text{Zwischkomplement}$$

Zwischkomplementbildung:

1. Abzuziehende Zahl auf volle Stellenzahl durch Vorgesetzten von Nullen ergänzen.

2. Abzuziehende Zahl invertieren

3. Zur invertierten Zahl 1 addieren

Vorzeichenbit :

Negative Dualzahlen sind durch eine 1 an erste Stelle (MSB) gekennzeichnet.

Ausgangssituation

Ihr Freund Sami bittet Sie als BKI-Schüler und angehenden PC-Experten wegen eines neuen PC-Komplettsystems um Hilfe bei der Auswahl der Komponenten.

Über einige Komponenten hat er sich schon vorab in einer PC-Zeitschrift unter der Rubrik „Die besten ...“ informiert. Deshalb will er unbedingt einen sog. „Intel“-Rechner mit dem Mainboard GIGABYTE Z590 GAMING X, was auch seinem begrenzten Budget (1000€) entspricht.

Vor der endgültigen Auswahl und Auftragerteilung wünscht er von Ihnen Kaufberatung und nähere Erklärungen zu Fachbegriffen und Daten.

Aufträge

1. Informieren Sie sich mit dem Video/der angegebenen Seite über die Komponenten eines PCs.

Links:



<http://slideplayer.org/slide/3136156/>



http://www.pcbasiswissen.de/pcbasiswissen/pc_hardware/aufbau.html

2. Bearbeiten Sie anschließend das Arbeitsblatt „Komponenten eines PC“ folgendermaßen:

- Tragen Sie in der Tabelle unter der Nummer die gesuchte Komponente ein.
- Tragen Sie in den Kästchen im Bild die zugehörige Nummer der Komponente ein.

3. Mit dem ALTERNATE-PC-Konfigurator stellt ihr euch nach den Vorgaben von Sami und Arbeitsblatt 1 euren Wahl-PC zusammen, d.h. Basis-Komponenten ohne Betriebssystem und Grafikkarte.



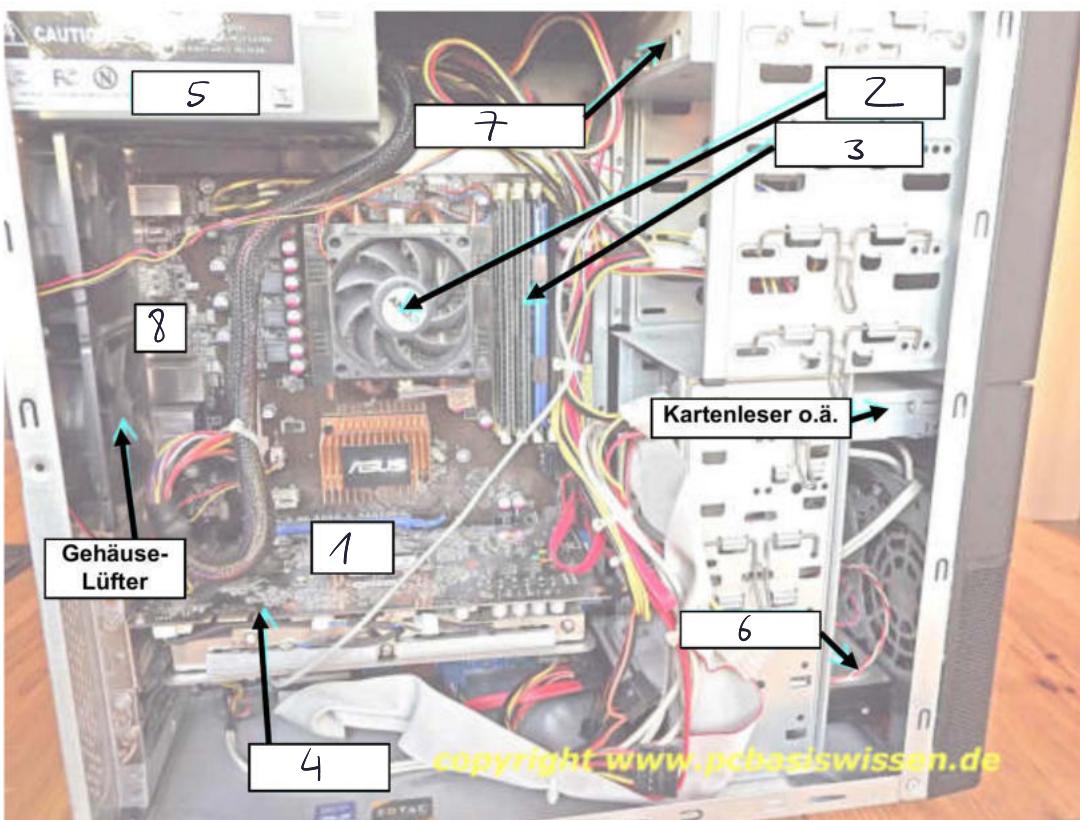
Link: <https://www.alternate.de/PC-Konfigurator#/components>

Der eine oder andere von euch hat nun festgestellt, dass die Komponenten nicht beliebig zusammengewürfelt werden können, sondern „zusammenpassen“ sollten.

4. Erklären Sie „zusammenpassen“ in diesem Bezug mit 2 Beispielen.

Videos zu vielen Themen: unter „The simple club“, „brainfaqk“, „IFDO“ auf YouTube.
<https://www.gigabyte.com/Motherboard/Z590-GAMING-X-rev-10#kf>

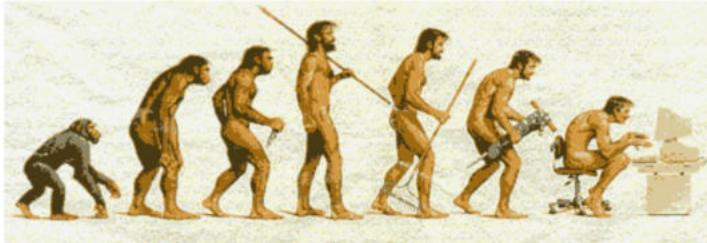
Arbeitsblatt: Komponenten eines PC



1 <u>Mainboard</u>	Zentral in der Mitte ist das Mainboard oder Motherboard, die Hauptplatine des PC. Auf ihr laufen alle Anschlüsse zusammen, der Prozessor, der Arbeitsspeicher, die Grafikkarte und verschiedene Erweiterungskarten befinden sich dort.
2 <u>(Lüfter) + Prozessor</u>	Die CPU (Central Prozessor Unit), der Rechner des PC ist eines der wichtigsten Teile, er führt alle Berechnungen durch und seine Leistungsfähigkeit bestimmt die Geschwindigkeit des Computers.
3 <u>Arbeitsspeicher</u>	In den Arbeitsspeichern werden Informationen zwischengespeichert und bei Bedarf abgerufen. Die Größe bestimmt die Menge der Daten die gespeichert werden und damit auch die Geschwindigkeit.
4 <u>Grafikkarte</u>	Die Grafik ist für die Darstellung auf dem Bildschirm nötig und es gibt Grafikprozessoren auf dem Mainboard, Prozessoren mit Grafik und extra Grafikkarten. Die Grafikleistung ist für schnelle und gute Bildwiedergabe entscheidend.
5 <u>Netzteil</u>	Das Netzteil ist die Energieversorgung für alle Systeme. Seine Leistung muss zum Energieverbrauch der Bauteile passen.
6 <u>Festplatte</u>	Die Festplatte ist der Speicher auf dem sich alle Daten und Programme befinden. Ohne Sie läuft nichts.
7 <u>Laufwerke</u>	Die Laufwerke ermöglichen es von einem Medium (CD, DVD, Blu-Ray) Daten einzulesen oder auf dieses Medium zu speichern.
8 <u>Sockelpanel</u>	Die Anschlüsse verbinden die inneren Bauteile eines PC mit den Äußerem. Verschiedene Geräte benötigen unterschiedliche Anschlüsse.

Die Geschichte der Rechenmaschinen

Von der Ziffer zum Allround-Computer



	Systeme	Hardware	Software
1980	Informations-systeme	VLSI – Prozessoren	Programmiersprachen
1950	EDV - Systeme	Transistoren	Gespeicherte Programme
1930	Rechenautomaten	Elektronenröhre	Lochstreifen
1800	Rechenautomaten	Mechanik	Lochkarten
1700	Webstuhl	Mechanik	Lochkarten
1600	Automaten	Mechanik	Dualsystem (Leibnitz)
1000 n.Chr. v.Chr.	Zähleräder	Mechanik	Dezimalsystem
	Abacus	Mechanik	Erste Zahlensysteme

Zahlensysteme und frühe Rechenhilfen

Geeignete Zahlensysteme → Grundlage des Rechnens
(Dezimalsystem; „10 Finger“)

- Abakus (1100 v.Chr., indo-chinesischer Kulturraum)
- Adam Riese (1522)
- John Napier (1550-1617)
Entwicklung von Logarithmentafeln, Rechenstäbchen (Napier's bones)
- William Oughtred (1632) Gilt als Erfinder des Rechenschiebers

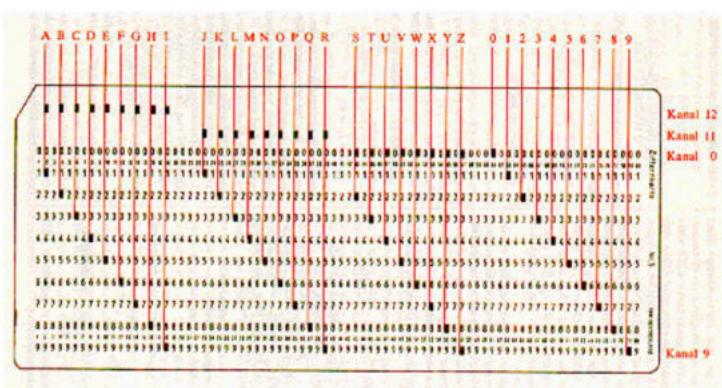
Mechanische Rechenmaschinen

- Wilhelm Schickardt (1592 – 1635)
Erste mechanische Rechenmaschine („Rechenuhr“, 1623) Addition und Subtraktion bis zu 6 Stellen
- Blaise Pascal (1623 - 1662) Pascaline
(sechsstellige Additionen und Subtraktionen, 1642-45)
- Gottfried Wilhelm von Leibnitz (1646 – 1716)
Entdeckung des Dualsystems, Gesetze der binären Arithmetik (1679)
- Charles Babbage :
Mathematiker, gilt als der "Vater der Computer".
Pionier des heutigen Computerkonzeptes
Analytical Engine, Universeller Rechner mit vielen Funktionen (1864)



Automatische Prozessteuerung

- Joseph Marie Jacquard (1801)
Lochkartenbänder aus Holz
- Herman Hollerith (1860 – 1929)
1889 Zähl- und Sortierapparatur
zur Informationsverarbeitung mit Lochkarten



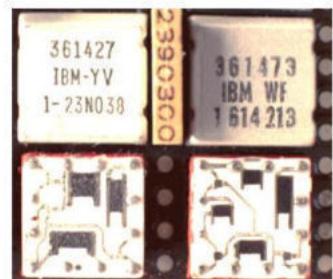
Digitale Rechenmonster: Relaisrechner

- Konrad Zuse „Vater des modernen Computers“
Z3 vorgestellt im Mai 1941: programmierbarer mit binärer Gleitkommarechnung arbeitender Relaisrechner
- Howard H. Aiken (1900 - 1973)
1934 -1944 Mark I; 1959
demontiert, war er eigentlich schon bei seiner Entwicklung veraltet.
- John von Neumann
Begründer des bis heute gültigen Arbeitskonzepts eines Computers



Computergenerationen

- 1. Generation
Elektronenröhren-Computer
(Ende 40er – Mitte 50er Jahre)
- 2. Generation
Mitte 50er - Anfang 60er Jahre
→ diskrete Halbleitertechnik
- 3. Generation
Anfang 60er - 70er Jahre
→ integrierte Halbleitertechnik
- 4. Generation
Anfang 70er Jahre
→ hochintegrierte Halbleitertechnik
- Supercomputer
1980 Cray-Computer (= 5. Generation??)



Linksammlung:

www.bernd-leitenberger.de

http://www.mathematik.uni-marburg.de/~thormae/lectures/ti1/ti_1_2_ger_web.html#1

Wikipedia mit entsprechenden Stichworten

<http://www.rechenschieber.org/englische.pdf>

Bilder z.T. über Suchmaschine „Google“

http://www.schule-bw.de/unterricht/faecher/physik/online_material/e_lehre_1/elektronik/mosfetani.gif

Video zur von-Neumann-Architektur: <http://www.youtube.com/watch?v=UomUEphRAwM>

Die Zuse-Story - Wie ein Deutscher den Computer erfand <http://www.youtube.com/watch?v=TUPj-Aep9PI>

Die kurze Geschichte des PC's

http://www.youtube.com/watch?v=yKk_gfvBvlg&index=2&list=PLZLLD7UZE9OPe669H4hwE8piX8OOhYnNO

UV-Diagramme ✓
Schaltfunktionen lernen

Schaltungen in „Nand“ oder „Nor“ umwandeln

Gleichungen in „Nand“ oder „Nor“ umwandeln

	a	b	c	\bar{J}	
0	0	0	0	1	
1	0	0	1	1	
2	0	1	0	1	
3	0	1	1	1	
4	1	0	0	0	
5	1	0	1	0	
6	1	1	0	0	
7	1	1	1	1	

$$J = (B \wedge \bar{A} \wedge C) \vee (\bar{A} \wedge B)$$

UO-Diagramme

2 Eingänge

A	\bar{A}
B	3 2
\bar{B}	1 0

3 Eingänge

A	I	\bar{A}
B	3 7 6 2	
\bar{B}	1 5 4 0	

4 Eingänge

A	\bar{A}	D
B	3 7 6 2 11 15 14 10	
\bar{B}	1 5 4 0	

Einfügen

A	B	X
0 0	0 0	0
1 0	1 1	1
2 1	0 1	1
3 1	1 0	0

Schaltalgebraisch kürzen

$$Z = (\bar{A} \wedge B) \vee (\bar{A} \wedge \bar{B})$$

$$Z = A \cdot (B \vee \bar{B})$$

$$Z = A$$

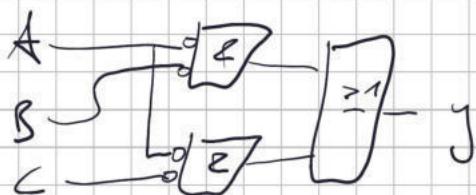
$$X = \bar{D}E\bar{F} \vee D\bar{E}\bar{F} \vee D\bar{E}\bar{F}$$

$$X = \bar{D}\bar{E}\bar{F} \vee (D\bar{E} \wedge (\bar{F} \vee \bar{F}))$$

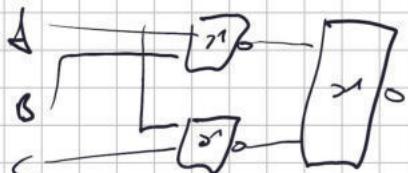
$$X = \bar{D}EF \vee D\bar{E}$$

$$y = \bar{C}BA \vee C\bar{B}A \vee \bar{C}\bar{B}A \vee C\bar{B}\bar{A}$$

$$y = (\bar{A} \wedge C) \vee (\bar{B} \wedge \bar{C})$$



Ugr:



A	\bar{A}
B	3 7 6 1
\bar{B}	1 1 1 0

Z	A	B	C	y
0	0 0	0 0	0	1
1	0 0	0 1	0	0
2	0 0	1 0	0	1
3	0 0	1 1	0	0
4	1 0	0 0	0	1
5	1 0	0 1	1	1
6	1 1	1 0	0	0
7	1 1	1 1	1	0

Full Nor / NAND

Nor:

$$Q = \overline{\overline{(A \wedge B)} \wedge \overline{(\bar{A} \wedge B)}}$$

change the sign, swap the loc.

$$Q = \overline{\overline{(A \wedge B)}} \vee \overline{\overline{(\bar{A} \wedge B)}}$$

$$Q = \overline{\overline{\bar{A} \vee B}} \vee \overline{\overline{\bar{\bar{A}} \vee \bar{B}}}$$

$$Q = \overline{\overline{\bar{A} \vee B}} \vee \overline{\overline{A \vee \bar{B}}}$$

$$Q = \overline{\overline{\bar{A} \vee \bar{B}}} \vee \overline{\overline{A \vee \bar{B}}}$$

Zeile	C	B	A	T	S	W	R
0	0	0	0	1	0	0	0
1	0	0	1	0	1	0	0
2	0	1	0	1	0	0	1
3	0	1	1	1	1	0	1
4	1	0	0	1	0	1	0
5	1	0	1	1	0	1	0
6	1	1	0	1	0	1	0
7	1	1	1	1	1	1	0

T:

	A	\bar{A}	
B	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀
\bar{B}	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀

$$T = (\bar{A} \cdot 1_0) \vee B$$

W:

	A	\bar{A}	
B	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀
\bar{B}	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀

$$w = c$$

D:

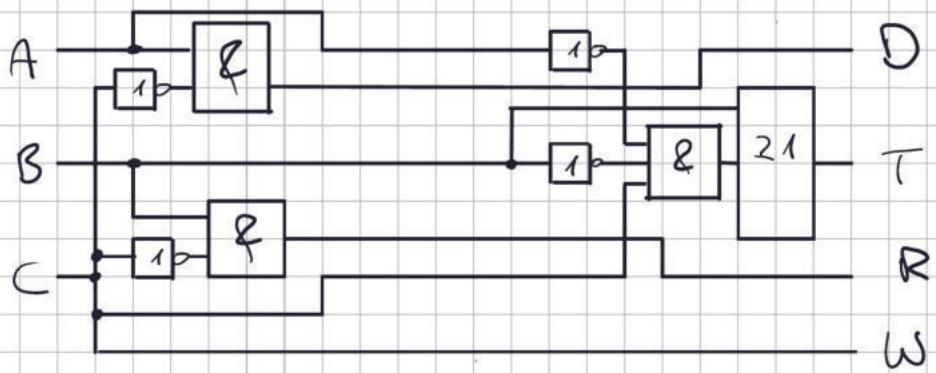
	A	\bar{A}	
B	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀
\bar{B}	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀

$$D = (A \cdot 1_2)$$

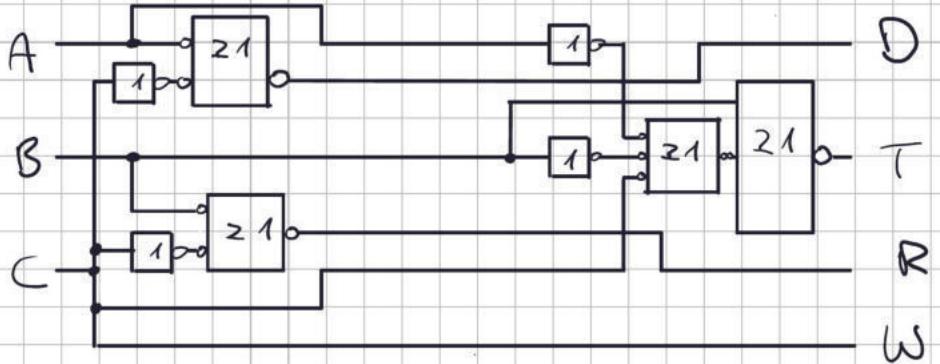
R:

	A	\bar{A}	
B	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀
\bar{B}	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀	1 ₃ 1 ₂ 1 ₁ 1 ₀

$$R = (B \cdot 1_2)$$



Fall Nor einsäuge Tastchen



zute	c	b	a	L
0	0	0	0	1
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

$L:$

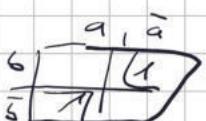
A	1	1	1
1 ₃	2	6	2
1 ₁	5	1 ₄	1 ₀
C	C	C	C

$$L: (\bar{A} \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B})$$

$$L: \overline{(\bar{A} \wedge \bar{C}) \vee (\bar{A} \wedge \bar{B})}$$

$$L = \overline{(\bar{A} \wedge \bar{C})} \wedge \overline{(\bar{A} \wedge \bar{B})}$$

$$L: \overline{(\bar{A} \wedge \bar{C})} \wedge \overline{(\bar{A} \wedge \bar{B})}$$



7

Aufbau - Wie werden Zahlen dargestellt?

Grundvoraussetzung: *sinuslicher System*

Römisches Zahlensystem: 2er- und 5er-Bündelung
Macht eine Rechnung sehr umständlich!



- ⇒ Wichtig sind nur die „Stellenwertsysteme“:
- Alle Zahlensysteme sind nach den gleichen Gesetzmäßigkeiten aufgebaut

Das Dezimalsystem

10 Ziffern: 0, 1, ..., 9

Menge > 9: Stellenwertigkeit eingeführt

$$10^0 = 1 \rightarrow \text{Potenzschreibweise}$$

$$\begin{array}{c|c} 10^1 & 10^0 \\ \hline & : \\ 1 & 0 \end{array}$$

Jede Dezimalzahl kann als Potenz der Basis 10 dargestellt werden.

Bsp.: Aufbau des Dezimalsystems

Tausender	Hunderter	Zehner	Einer	Zehntel
10^3	10^2	10^1	10^0	10^{-1}
1	2	3	4	5

stellenwert
stellenwertfaktor

$$1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0 + 5 \cdot 10^{-1} = 1234,5_{10}$$

Allgemein:

$$z = \sum_{i=-m}^n a_i b_i$$

stellenwert
stellenwertfaktor

Das Dualsystem

Basis: 2

Ziffern: 0, 1

Jede Dualzahl kann als Potenz der Basis 2 dargestellt werden.

Bsp.: Aufbau des Dualsystems

2^3	2^2	2^1	2^0	2^{-1}
1	0	1	0	1

stellenwert
stellenwertfaktor

$$1 \quad 0 \quad 1 \quad 1 \quad 0,5 = 1010,1_2$$

$$= 10,5_{10}$$

Dezimal-, Dual- und Hexadezimalzahlen

Stellenwertigkeit	Dual								Dezimal		Hexadezimal	
						2^1	2^0	10^1	10^0	16^1	16^0	
0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	1	0	1	0	1	
0	0	0	0	0	0	1	0	0	2	0	2	
0	0	0	0	0	1	0	0	0	3	0	3	
0	0	0	0	1	0	1	1	0	4	0	4	
0	0	0	0	0	1	0	0	0	5	0	5	
0	0	0	0	0	1	0	1	0	6	0	6	
0	0	0	0	0	1	1	0	0	7	0	7	
0	0	0	0	0	1	1	1	0	8	0	8	
0	0	0	0	1	0	0	0	0	9	0	9	
0	0	0	0	0	1	0	1	0	10	0	A	
0	0	0	0	0	1	0	1	0	11	0	B	
0	0	0	0	1	0	1	1	1	12	0	C	
0	0	0	0	0	1	1	0	1	13	0	D	
0	0	0	0	0	1	1	0	1	14	0	E	
0	0	0	0	1	0	1	0	1	15	0	F	
0	0	0	0	0	1	1	1	1	16	1	0	
0	0	0	0	0	1	0	0	0	17	1	1	
0	0	0	0	1	0	0	1	0	18	1	2	
0	0	0	0	0	1	0	1	1	19	1	3	
...	
0	0	0	1	1	0	0	1	2	5	1	9	
0	0	0	0	1	1	0	0	2	6	1	A	
0	0	0	0	0	1	1	0	1	7	1	B	
...	-1	
0	0	0	1	1	1	1	1	3	1	1	F	
0	0	0	1	0	0	0	0	3	2	2	0	
0	0	0	1	0	0	0	0	1	3	2	1	
...	
...	
2^{10}								1024	400			
2^{16}								65536	10000			
2^{20}								1048576	100.000			

Potenzen von 2^n

n =	5	4	3	2	1	0	-1	-2	-3	-4	-5
$2^n =$	32	16	8	4	2	1	$1/2 = 0,5$	$1/4 = 0,25$	$0,125$	$0,0625$	$0,03125$

④

a) 0000 0000 bis 0DFF FFFF

$$\begin{array}{r} 16 \\ \hline 16 \end{array}^6$$

b)

Ausgangsadresse 00h

Endadresse 1Ah

$$1A - 00 = 1A + 1 = \underline{\underline{1B}} \text{ h}$$

$$\hookrightarrow 1B = 1 \cdot 16^0 + 11 = \underline{\underline{27}} \text{ dez.}$$

Wie rechnet der Computer?

mit 0, 1

Subtraktion:

$$\hookrightarrow 4 - 3$$

Addition mit negativen Vorzeichen

$$(4) + (-3)$$

↑ Redenzichen

$$\begin{array}{r}
 100 \\
 -011 \\
 \hline
 100
 \end{array}$$

$\begin{array}{r}
 100 \\
 100 \\
 \hline
 1001
 \end{array}$
) Einshkomplement
 $\begin{array}{r}
 100 \\
 100 \\
 \hline
 001
 \end{array}$
) zwischkomplement.
 $\underline{\underline{001}}_b = 1_d$

 MES Max-Eyth-Schule <small>Gewerbliche Schule Kirchheim unter Teck</small>	Minimieren von Logikschaltungen durch KV - Diagramme	Klasse: Fach :
---	---	-------------------------------

Aufgabe: Füllstandssteuerung

Der Inhalt eines Silos kann über die Pumpen P1 und P2 entleert werden (s. Bild 1.1). Die Auswahl der Pumpe, d.h. welche der beiden Pumpen bei der Entleerung des Silos eingeschaltet wird, ist abhängig vom Silofüllstand.

- Befindet sich der Füllstand unterhalb von Sensor S2, ist Pumpe P1 einzuschalten.
- Liegt der Füllstand zwischen Sensor S2 und Sensor S3, wird die Pumpe P2 eingeschaltet.
- Bei Füllstand oberhalb von Sensor S3 laufen beide Pumpen P1 und P2.
- Die Meldeleuchte H1 leuchtet auf, sobald S3 anspricht.
- Die Entleerung des Silos wird mit dem Schalter S4 ein- und ausgeschaltet.
- Beim Auftreten einer Fehlstelle (z.B. Sensor S3 meldet und Sensor S2 meldet nicht) wird eine Meldelampe H1 eingeschaltet ($S_4 = 1$). Die Meldung erfolgt auch, wenn keine Siloentleerung stattfindet ($S_4 = 0$).

Hinweis: Alle Schalter liefern beim Betätigen '1'. Die Pumpen und die Störungsanzeige sind mit '1' aktiv.

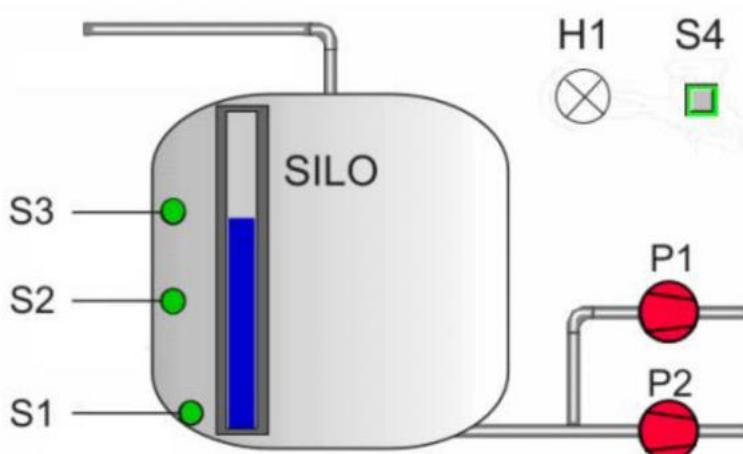


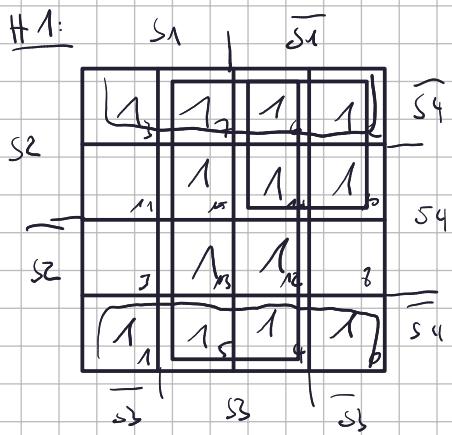
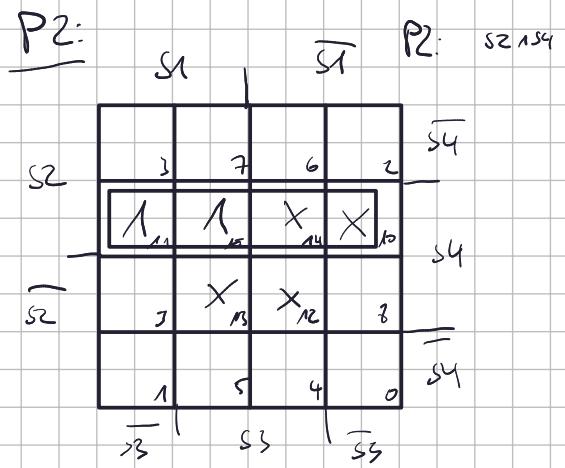
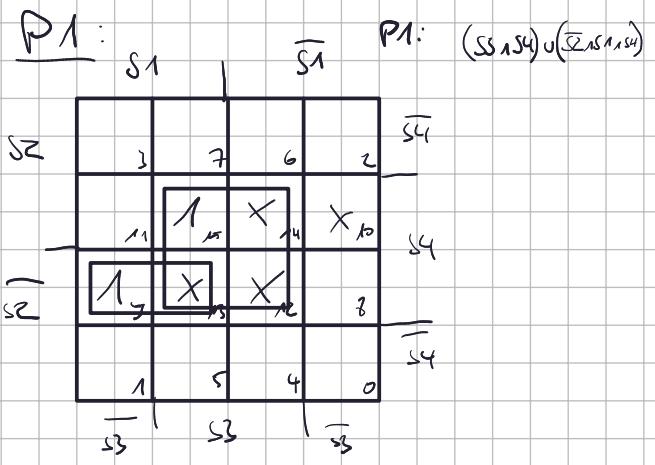
Bild 1.1: Siloentleerung

Aufgaben:

- Ermitteln Sie die minimierten Funktionsgleichungen der Pumpen P1 und P2.
- Zeichnen Sie die Schaltungen in Full - NOR - Technik.
- Ermitteln Sie die minimierte Funktionsgleichung der Meldeleuchte H1.
- Zeichnen Sie deren Schaltung in Full - NAND - Technik.

d)

	s_4	s_3	s_2	s_1	P_1	P_2	H_1
0	0	0	0	0	0	0	1
1	0	0	0	1	0	0	1
2	0	0	1	0	0	0	1
3	0	0	1	1	0	0	1
4	0	1	0	0	0	0	1
5	0	1	0	1	0	0	1
6	0	1	1	0	0	0	1
7	0	1	1	1	0	0	1
8	1	0	0	0	0	0	0
9	1	0	0	1	1	0	0
10	1	0	1	0	X	X	1
11	1	0	1	1	0	1	0
12	1	1	0	0	X	X	1
13	1	1	0	1	X	X	1
14	1	1	1	0	X	X	1
15	1	1	1	1	1	1	1



$$H_1: s_3 \vee \bar{s}_4 \vee (\bar{s}_1 \wedge s_2)$$

Don't-Care-Zustände sind Ergebnisse, die „egal“ sind.

Beispielsweise handelt es sich um Zeilen in der Wahrheitstabelle, für die keine Eingaben vorgesehen sind, physikalisch oder technisch unmöglich sind und folglich auch keine Ausgaben berücksichtigt werden wollen.

Sie werden als „X“ in das KV-Diagramm eingetragen und dürfen für den Zweck der Gruppenbildung als „1“ oder „0“ betrachtet werden.

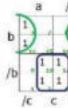
Die Wahl erfolgt so, dass sich besonders günstig Gruppen bilden lassen (wenig Gruppen, große Gruppen).

	A	\bar{A}	\bar{A}	\bar{A}
B			1	1
B			1	1
\bar{B}		X	X	1
\bar{B}	1	1	1	1
C	\bar{C}	\bar{C}	\bar{C}	C

D Bild 7-2a zeigt eine optimale Gruppenbildung mit diesen Don't-Care-Zuständen. Werden beide „X“ in den roten Block einbezogen und zum 4er-Block erweitert, ist die Blockbildung optimal.

	A	\bar{A}	\bar{A}	\bar{A}
B			1	1
B			1	1
\bar{B}				1
\bar{B}	1	1		1
C	\bar{C}	\bar{C}	\bar{C}	C

D Ohne diese Don't-Care-Zustände wäre die Gruppenbildung ungünstiger (Bild 7-3).
[https://de.wikibooks.org/wiki/Karnaugh-Veitch-Diagramm:_Beispiele_\(Teil_3\)](https://de.wikibooks.org/wiki/Karnaugh-Veitch-Diagramm:_Beispiele_(Teil_3))



Aufgabe: Elektronischer Würfel

Würfeln Sie doch mal elektronisch! Dieses Gerät soll bei Würfelspielen den almodischen Würfel ersetzen. Auf der Platine sind 7 Leuchtdioden entsprechend den Augen auf einem Würfel angeordnet. Nach dem Drücken und wieder Loslassen der Starttaste leuchten zufallsbedingt eine oder mehrere der LED's auf. Die Anzeige entspricht den Augen 1...6 eines Würfels. Die letzte Anzeige bleibt solange stehen, bis die Starttaste erneut gedrückt wird.

Die gewürfelte Zahl wird ebenfalls über eine Siebensegmentanzeige als Dezimalzahl angezeigt.

Für die Aufgabenstellung beachten und entwickeln wir an dieser Stelle nur die Ansteuerschaltung der 7 LEDs.

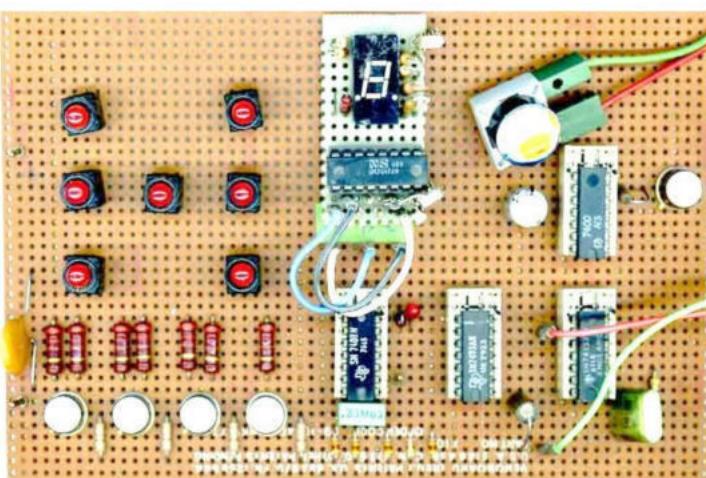
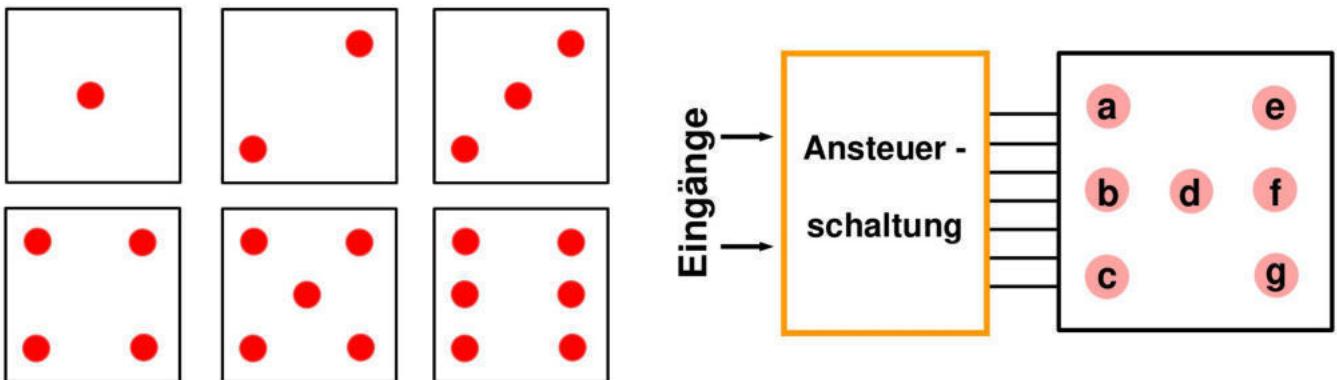


Bild: http://mezdata.de/ti-basis/510_elektr-wuerfel/index.php



- Vorüberlegung: Wie viele „Eingänge“ (\rightarrow „Zustände“) hat der Würfel?
- Erstellen Sie die komplette Wertetabelle.
- Vereinfachen Sie die Funktionsgleichungen mit KV-Diagrammen.

Zeile	? Zustände / Eingänge	a	...	g
0	...			

zile	x_2	x_1	x_0	a	b	c	d	e	f	g
0	0	0	0	X	X	X	X	X	X	X
1	0	0	1	0	0	0	1	0	0	0
2	0	1	0	0	0	1	0	1	0	0
3	0	1	1	0	0	1	1	1	0	0
4	1	0	0	1	0	1	0	1	0	1
5	1	0	1	1	0	1	1	1	0	1
6	1	1	0	1	1	1	0	1	1	1
7	1	1	1	X	X	X	X	X	X	X

$$d = x_0$$

$$ce = \bar{x}_2 x_1$$

$$cde = x$$

a)

	x_0	\bar{x}_0	
x_1	1, x_2	1, x_2	2
\bar{x}_1	1, x_2	1, x_2	0
	x_2	\bar{x}_2	

$$a = x_2$$

b)

	x_0	\bar{x}_0	
x_1	1, x_2	1, x_2	2
\bar{x}_1	1, x_2	1, x_2	0
	x_2	\bar{x}_2	

$$b = x_1 \wedge x_2$$

c)

	x_0	\bar{x}_0	
x_1	1, x_2	1, x_2	2
\bar{x}_1	1, x_2	1, x_2	0
	x_2	\bar{x}_2	

$$c = x_1 \vee x_2$$

d)

	x_0	\bar{x}_0	
x_1	1, x_2	1, x_2	2
\bar{x}_1	1, x_2	1, x_2	0
	x_2	\bar{x}_2	

$$d = x_0$$

e)

	x_0	\bar{x}_0	
x_1	1, x_2	1, x_2	2
\bar{x}_1	1, x_2	1, x_2	0
	x_2	\bar{x}_2	

$$e = x_1 \vee x_2$$

f)

	x_0	\bar{x}_0	
x_1	1, x_2	1, x_2	2
\bar{x}_1	1, x_2	1, x_2	0
	x_2	\bar{x}_2	

$$f = x_1 \wedge x_2$$

g)

	x_0	\bar{x}_0	
x_1	1, x_2	1, x_2	2
\bar{x}_1	1, x_2	1, x_2	0
	x_2	\bar{x}_2	

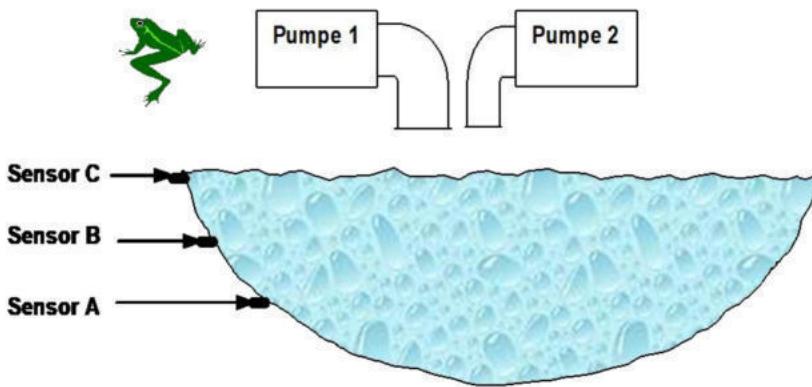
$$g = x_2$$

! Anstatt die 7 KU-Diagramme, kann man auch 4 benutzen, da sich manche doppeln und man die x' 'e weglassen kann.

 MES Max-Eyth-Schule <small>Gewerbliche Schule Kirchheim unter Teck</small>	Minimieren von Logikschaltungen durch KV - Diagramme	Klasse: Fach :
---	---	-------------------------------

Teichbefüllung

Als Amphibienfreunde wollen wir das Überleben der Frösche im Gartenteich sichern und entwickeln eine Steuerung zur Befüllung eines Froschteiches. Dieser kann durch zwei Pumpen P1 und P2 befüllt werden, wobei P1 die größere Leistung hat. Durch die drei Sensoren A, B, C kann die Füllstandshöhe gemessen werden.



Die Sensoren liefern 1 Signal wenn das Wasser sie erreicht bzw. 0 Signal, wenn kein Wasser am Sensor ist.

- Befindet sich der Wasserstand unterhalb des Sensors A, dann laufen beide Pumpen.
- Befindet sich der Wasserstand zwischen den Sensoren A und B darf nur Pumpe 1 laufen.
- Befindet sich der Wasserstand zwischen den Sensoren B und C darf nur Pumpe 2 laufen.
- Erreicht der Wasserstand den Sensor C oder höher darf keine der Pumpen in Betrieb sein.

Lösungshinweis:

Falls trotz entsprechender Füllhöhe einzelne darunter liegende Fühler nicht ansprechen, steuert der jeweils oberste Sensor die Pumpen.

Aufgabe :

Ermitteln Sie die minimierten Schaltfunktionen für P1 und P2 mittels KV-Diagrammen.

	C	B	A	P1	P2
0	0	0	0	1	1
1	0	0	1	1	0
2!	0	1	0	x	x
3	0	1	1	0	1
4!	1	0	0	x	x
5!	1	0	1	x	x
6!	1	1	0	x	x
7	1	1	1	0	0

	X	1	\bar{A}
B			x
\bar{B}	(1)	x	x

$$P1 = \bar{B} \quad \text{Weiter vereinfachen}$$

	X	1	\bar{A}
B	(1)	x	x
\bar{B}	x	x	(1)

Bestimmte Zustände können nicht auftreten, d.h. das Ausgangssignal kann sowohl '1' als auch '0' sein:

\Rightarrow don't care

In der Wertetabelle markieren wir entsprechende Stellen mit 'x'.

1.8 121P
§ 4.1

Inf- I K42 Korrektur

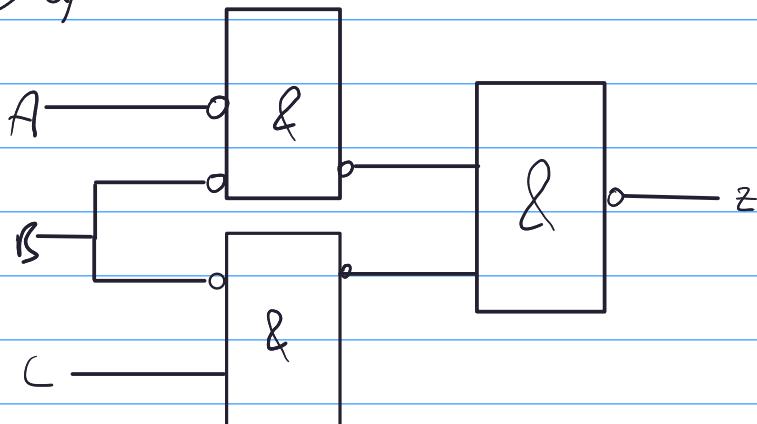
①

$$y = (\bar{A} \wedge \bar{B} \wedge \bar{C}) \vee (\bar{A} \wedge B \wedge C) \vee (A \wedge \bar{B} \wedge C) \vee (A \wedge B \wedge \bar{C})$$

$$y = (\bar{B} \wedge \bar{C}) \wedge (\bar{A} \wedge \bar{A}) \vee (A \wedge B) \wedge (\bar{C} \vee C)$$

$$y = (\bar{B} \wedge \bar{C}) \vee (A \wedge B)$$

③ a)



④ b)

$$x = (c \wedge d \wedge \bar{a}) \vee (c \wedge \bar{b} \wedge a) \vee (a \wedge c \wedge \bar{d})$$

c)

	E_1	\bar{E}_1		
E_2				E_4
\bar{E}_2				\bar{E}_4
	1	1	1	
	1	1	1	\bar{E}_4
				E_4
	E_3	\bar{E}_3	E_5	

$$Q = (\bar{E}_1 \wedge \bar{E}_2) \vee (\bar{E}_2 \wedge E_3) \vee (\bar{E}_2 \wedge \bar{E}_4)$$