1 Digital vs. Analog

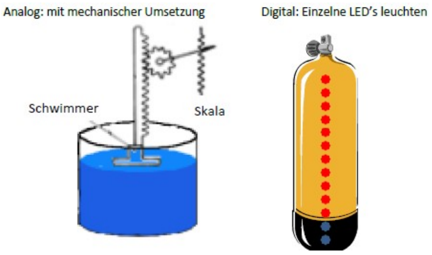
1.1 Analog

- Die reale Welt ist analog (z.B. Sinnesorgane)
- Die Analoge Verarbeitung stellt das Ergebnis einer Berechnung praktisch sofort zur Verfügung.
- Analoge Systeme sind anfällig auf externe Störsignale.
- Analoge Systeme sind extrem aufwändig und erfordern viel Fachwissen.

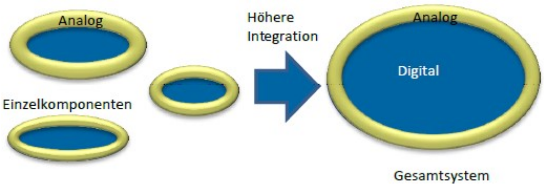
1.2 Digital

- Einfacher automatisierbarer Entwurf und Test mit Hilfe von CAD möglich.
- Digitale Schaltungen werden als integrierte Schaltungen mit Transistoren hergestellt uns sind somit reproduzierbar.
- Digitale Systeme sind bis zu einem gewissen Grad weitgehend immun gegen Störsignale.

1.3 Unterschied Analog zu Digital

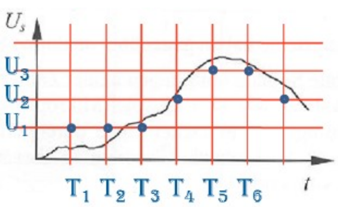


1.3.1 Integration



Trend zu höherer Integration. Digitaler Anteil wächst und die Analoge „Schale“ bleibt bestehen.

1.4 Von Analog zu Digital



Abtasttheorem:	Nyquist-Shannon besagt, dass ein Signal mit f_{max} mit mindestens einer Frequenz von $2 * f_{max}$ abgetastet werden muss um das Ursprungssignal wieder herzustellen.
Amplitudenauflösung:	Die Anzahl abzählbarer Amplitudenwerte (Quantisierungsstufen) bestimmt die Auflösung eines AD-Wandlers. Diese wird in der Regel in Bits angegeben. Je kleiner der Bereich aufgeteilt ist, desto kleiner ist der Abstand Δ zwischen zwei benachbarten Amplitudenwerten.
Quantisierungsfehler:	Bei linearer Quantisierung ist der der Quantisierungsfehler (Quantisierungsrauschen) maximal $\frac{\Delta}{2}$

2 Zahlendarstellungen und Codes

2.1 Zahlensysteme

2.1.1 Zahlensysteme ohne festen Stellenwert

- Römisches Zahlensystem (keine Null, kein fester Stellenwert, schlechte Unterscheidung)

2.1.2 Zahlensysteme mit festem Stellenwert

- Babylon (Basis B = 60)
- Maya (Basis B = 20)
- Dezimal (Basis B = 10)
- Binär (Basis B = 2)
- Octal (Basis B = 8)
- Hexadezimal (Basis B = 16)

Die Wertigkeit des Symbols hängt von seiner Position innerhalb der Symbolkette ab:

$$z = \sum_{k=0}^{n-1} a_k * B^k$$

z: Wert der Zahl (im Dezimalsystem)
a: Nennwert der Ziffer
B: Basis des Zahlensystems
n: Stellenanzahl

Bsp: $4156.78 = 4 * 10^3 + 1 * 10^2 + 5 * 10^1 + 6 * 10^0 + 7 * 10^{-1} + 8 * 10^{-2}$

Auch gebrochene Zahlen können nach dem gleichen Muster binär dargestellt werden. Wichtig ist, dass das Komma immer an einer festen Stelle steht (Festkommadarstellung). Definiert ist, dass eine Binärzahl 8 Ziffern (n) vor dem Komma und 4 Ziffern (m) nach dem Komma besitzt. Eine gebrochene Binärzahl sieht dann so aus:

$$z_2 = a_{n-1}a_{n-2} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m+1} a_{-m}$$

Der gesuchte Zahlenwert im Dezimalsystem wird dann folgendermassen berechnet:

$$z_{10} = a_{n-1} * B^{n-1} + a_{n-2} * B^{n-2} + \dots + a_0 * B^0 + a_{-1} * B^{-1} + a_{-2} * B^{-2} + \dots + a_{-m+1} * B^{-m+1} + a_{-m} * B^{-m}$$

2.2 Gebräuchliche polyadische Zahlensysteme

System	Basis	Stellenwerte	Ziffern	Beispiel
Binär	2	$\dots 2^2\ 2^1\ 2^0 \dots$	0, 1	$110_{(2)} = 6_{(10)}$
Oktal	8	$\dots 8^2\ 8^1\ 8^0 \dots$	0, 1, 2, 3, 4, 5, 6, 7	$273_{(8)} = 187_{(10)}$
Dezimal	10	$\dots 10^2\ 10^1\ 10^0 \dots$	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	$192_{(10)} = 192_{(10)}$
Hexadezimal	16	$\dots 16^2\ 16^1\ 16^0 \dots$	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	$2AFF_{(16)} = 11007_{(10)}$

2.3 Begriffe im Zusammenhang mit dem binären Zahlensystem

- Bit (b): Binary Digit: Kleinsmögliche Speichereinheit in der Digitaltechnik. Kann zwei mögliche Zustände annehmen: 0 und 1
- Byte (B): Einheit von 8 Bits. Auch genannt Oktett: 1 Oktett = 1 Byte = 8 Bit. Byte ist die Standartbezeichnung von Speicherkapazitäten und Datenmengen.
- Nibble: Binärzahlen in Gruppen von 4 Bits
- MSB: Most Significant Bit. Bit mit höchster Wertigkeit, steht ganz links im binären Wort
- LSB: Least Significant Bit. Bit mit tiefster Wertigkeit, steht ganz rechts im binären Wort

2.4 Umwandlung von Dezimalzahlen

Beispiel für die Umwandlung der Zahl $109.78125_{(10)}$:

Dezimal zu Binär						
109	:	2	=	54	Rest:	1
54	:	2	=	27	Rest:	0
27	:	2	=	13	Rest:	1
13	:	2	=	6	Rest:	1
6	:	2	=	3	Rest:	0
3	:	2	=	1	Rest:	1
1	:	2	=	0	Rest:	1
0.78125	*	2	=	0.5625	+	1
0.5625	*	2	=	0.125	+	1
0.125	*	2	=	0.25	+	0
0.25	*	2	=	0.5	+	0
0.5	*	2	=	0	+	1
$109_{(10)} = 1101101.11001_{(2)}$						

Dezimal zu Oktal						
109	:	8	=	13	Rest:	5
13	:	8	=	1	Rest:	5
1	:	8	=	0	Rest:	1
0.78125	*	8	=	0.25	+	6
0.25	*	8	=	0	+	2
$109_{(10)} = 155.62_{(8)}$						

Dezimal zu Hexadeximal						
109	:	16	=	6	Rest:	D
6	:	16	=	0	Rest:	6
0.78125	*	16	=	0.5	+	C
0.5	*	16	=	0	+	8
$109_{(10)} = 6D.C8_{(16)}$						

2.5 Addition

2.5.1 Ganzzahlen

	0	1	1	1	1	1	1	0
+	0	0	1	1	0	1	0	1
Carry	1	1	1	1	1	0	0	
=	1	0	1	1	0	0	1	1

2.5.2 Festkommazahlen

	0	1	1	0	0,	0	1	0
+	0	0	1	1	0,	1	1	1
Carry	1	1	0	0	1	1	0	
=	1	0	0	1	1,	0	0	1

2.5.3 Vorzeichen-Darstellung

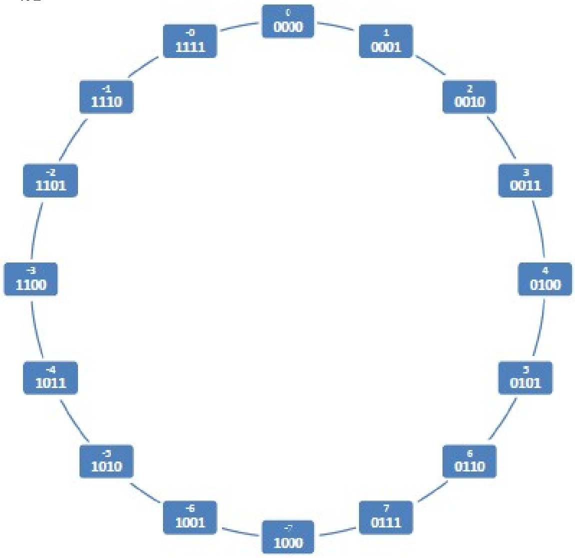
Es gilt:
- 0→positiv
- 1→negativ

Beispiel:

+	12310	=	0 111 1011
−	12310	=	1 111 1011

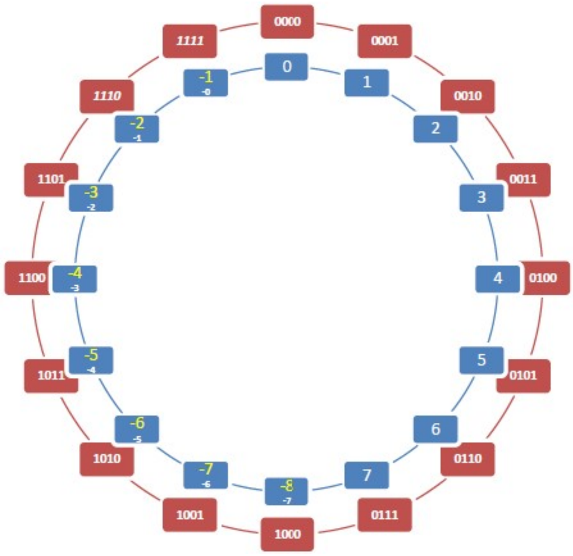
2.6 Einerkomplement

Das Einerkomplement wird durch Invertieren aller Bits der positiven Zahl gebildet.
$A_{k1} = \overline{A}$



2.7 Zweierkomplement

Das Zweierkomplement wird durch Invertieren aller Bits der positiven Zahl und der Addition von 1 gebildet.
$A_{k2} = \overline{A} + 1 = (2^n - A)$



2.8 Subtraktion

Für die Subtraktion A−B=C sind folgende Schritte notwendig:

- 1) Der Subtrahend B muss ins Zweierkomplement gebracht werden.
- 2) Die arithmetische Operation muss durchgeführt werden.
- 3) Das Resultat muss korrekt interpretiert werden.

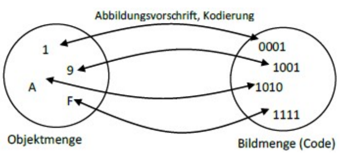
In der Praxis bedeutet die Subtraktion von 2^n, dass das Bit, welches an höchster Stelle $(n + 1)$ steht, ignoriert wird.

2.9 Bereichsüberschreitung

In Hardware sind die Wortbreiten immer begrenzt, so kann es passieren, dass diese Wortbreite bei einer arithmetischen Operation überschritten wird.
Mit diesem Verfahren kann geprüft werden, ob eine Ergebnis korrekt ist:

Operation	Richtiges Ergebnis	Überlauf
A+B	$c_n = 0,\ c_{n-1} = 0$	$c_n = 0,\ c_{n-1} = 1$
A−B	$c_n = c_{n-1}$	Nicht möglich
−A−B	$c_n = 1,\ c_{n-1} = 1$	$c_n = 1,\ c_{n-1} = 0$

2.10 Codes



Folgende Codes sind die am häufigst gebrauchten:

- Binärcode
- Reiner Dualcode: Zuordnung von Dezimalzahlen zu den Bitfolgen des Dualsystems
- BCD Code:

Dezimal	1	2	3	...	9	>9
BCD	0001	0010	0011	...	1001	ungültig

3 Schaltalgebra

3.1 Symbolik

Folgende Symbole werden verwendet:

$\neg = \overline{\ } =$NOT $\quad \vee = + =$OR $\quad \wedge = * =$AND $\quad \oplus = \$ =$EXOR

3.2 Rechenregeln

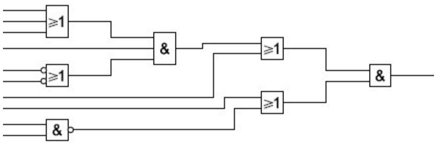
Verknüpfung mit 0	$a \vee 0 = a$	$a \wedge 0 = 0$	$a \oplus 0 = a$
Verknüpfung mit 1	$a \vee 1 = 1$	$a \wedge 1 = a$	$a \oplus 1 = \overline{a}$
Verkn. mit sich selbst	$a \vee a = a$	$a \wedge a = a$	$a \oplus a = 0$
Verkn. mit Inversem	$a \vee \overline{a} = 1$	$a \wedge \overline{a} = 0$	$a \oplus \overline{a} = 1$
Kommutativgesetz	$a \vee b = b \vee a$	$a \wedge b = b \wedge a$	$a \oplus b = b \oplus a$
Assioziativgesetz	$(a \vee b) \vee c = a \vee (b \vee c)$	$(a \wedge b) \wedge c = a \wedge (b \wedge c)$	$(a \oplus b) \oplus c = a \oplus (b \oplus c)$
Distributivgesetz	$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$	$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$	$a \wedge (b \oplus c) = (a \wedge b) \oplus (a \wedge c)$

3.3 Vereinfachungen

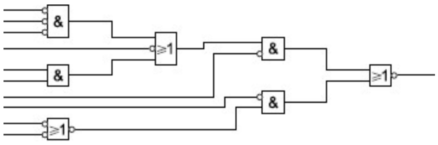
$a \vee (a \wedge b) = a$	$(a \wedge \overline{b}) \vee b = a \vee b$	$(a \wedge \overline{b}) \oplus b = a \vee b$	$(a \wedge b) \vee (a \wedge \overline{b}) = a$
$a \wedge (a \vee b) = a$	$(a \vee \overline{b}) \wedge b = a \wedge b$	$(a \oplus \overline{b}) \wedge b = a \wedge b$	$(a \vee b) \wedge (a \vee \overline{b}) = a$

3.4 Shannon und DeMorgan

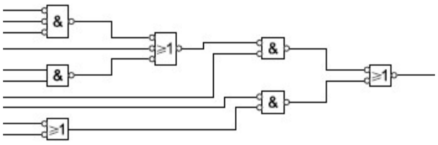
Ursprungsschaltung:



Shannon



DeMorgan



3.5 Normalformen

Ausgangslage:

Dezimal	x2	x1	x0	y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	d
6	1	1	0	1
7	1	1	1	1

3.5.1 Kanonisch disjunktive Normalform

$\overline{x2} \wedge \overline{x1} \wedge x0$
$\overline{x2} \wedge x1 \wedge x0$
$[x2 \wedge \overline{x1} \wedge x0]$
$x2 \wedge x1 \wedge \overline{x0}$
$x2 \wedge x1 \wedge x0$

$y = (\overline{x2} \wedge \overline{x1} \wedge x0) \vee (\overline{x2} \wedge x1 \wedge x0) \vee [x2 \wedge \overline{x1} \wedge x0] \vee (x2 \wedge x1 \wedge \overline{x0}) \vee (x2 \wedge x1 \wedge x0)$

Es werden nur diejenigen Zeilen der Wahrheitstabelle aufgeführt, deren Funktionswert 1 oder d ist.

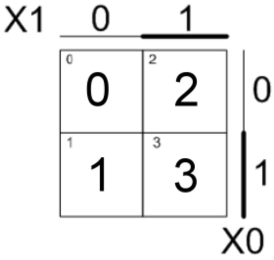
3.5.2 Kanonisch konjunktive Normalform

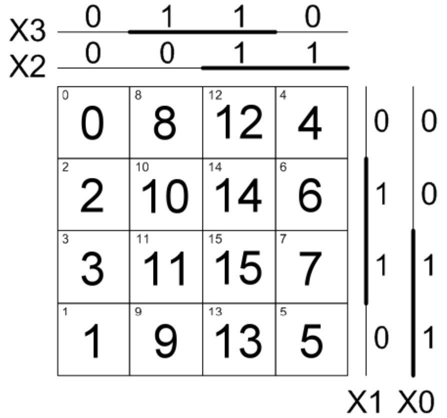
$x2 \vee x1 \vee x0$
$x2 \vee \overline{x1} \vee x0$
$\overline{x2} \vee x1 \vee x0$
$[\overline{x2} \vee x1 \vee \overline{x0}]$

$y = (x2 \vee x1 \vee x0) \wedge (x2 \vee \overline{x1} \vee x0) \wedge (\overline{x2} \vee x1 \vee x0) \wedge [\overline{x2} \vee x1 \vee \overline{x0}]$

Es werden nur diejenigen Zeilen der Wahrheitstabelle aufgeführt, deren Funktionswert 0 oder d ist.

3.6 Karnaugh-Diagramm





3.6.1 Arbeiten mit dem KV-Diagramm

- 1) Aufstellen der Wahrheitstabelle.
- 2) Übertragen der Werte der Wahrheitstabelle in KV Diagramm.
- 3) Möglichst grosse Gruppen a 2^n Felder bilden.
- 4)

Kanonisch disjunktive Normalform:	Kanonisch konjunktive Normalform:
Gruppen von Feldern mit Wert 1 oder d	Gruppen von Feldern mit Wert 0 oder d
Primimplikanten: AND Verknüpfung	Primimplikanten: OR Verknüpfung
OR Verknüpfung aller Primimplikanten	AND Verknüpfung aller Primimplikanten

4 Logische Gatter

4.0.2 Ralisierungsaufwand

Anzahl Transistoren ist ein mögliches Mass für Komplexität (Marketing). In der Technik spricht man von Gatteräquivalent: 1 Gatteräquivalent entspricht 4 Transistoren.

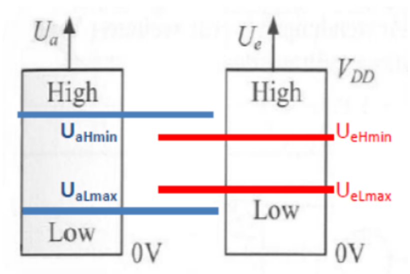
4.1 Anzahl Schaltfunktionen

Permutation: $F = 2^{2^n}$ F: Anzahl möglicher Schaltfunktionen, n: Anzahl Eingänge

4.2 Verhalten logischer Gatter

4.2.1 Störabstand

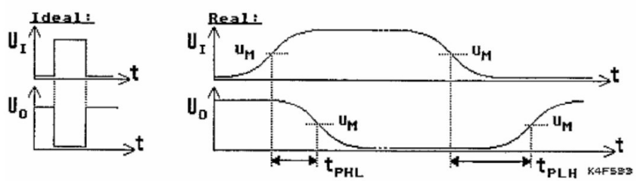
High-Pegel: $U_{nH} = U_{aHmin} - U_{eHmin}$
Low-Pegel: $U_{nL} = U_{eLmax} - U_{aLmax}$



4.2.2 Propagation delay Time Anstiegs-/Abstiegszeit

Zeit zwischen 50% von V_{max} am Eingang und 50% von V_{max} am Ausgang.

$t_{pd} = \frac{t_{pLH} + tpHL}{2}$

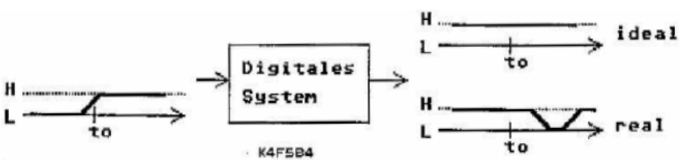


4.2.3 Transition delay Time (Verzögerungszeit)

Zeit zwischen 10% und 90% von V_{max}.
t_{tLH}: Transition time low to high.
t_{tHL}: Transition time high to low.

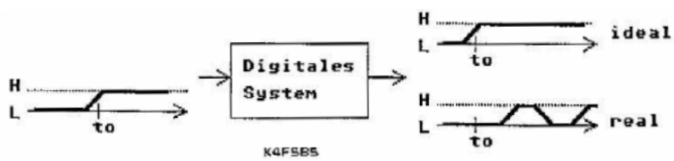
4.3 Hazards

4.3.1 Statischer Hazard



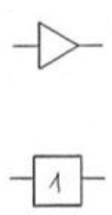
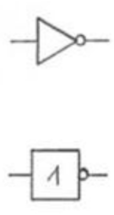
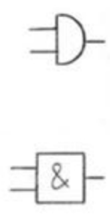
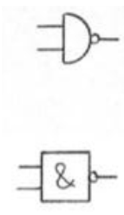
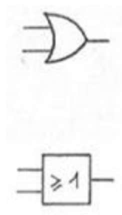
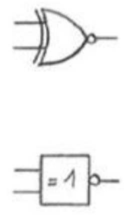
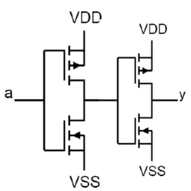
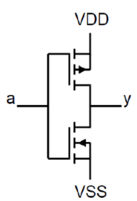
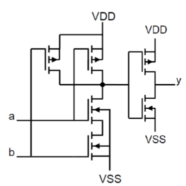
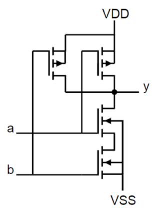
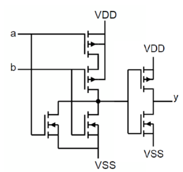
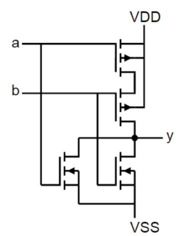
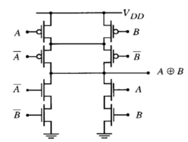
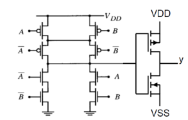
Der Funktionswert eines kombinatorischen Systems ändert kurzzeitig seinen Pegel, wenn eine Eingangsvariable den Pegel ändert, obschon von der logischen Funktion keine Änderung gegeben ist.

4.3.2 Dynamischer Hazard



Der Funktionswert eines kombinatorischen Systems ändert mehrmals kurzzeitig seinen Pegel, wenn eine Eingangsvariable den Pegel ändert, obschon von der logischen Funktion her nur eine einzige Änderung erfolgen sollte.

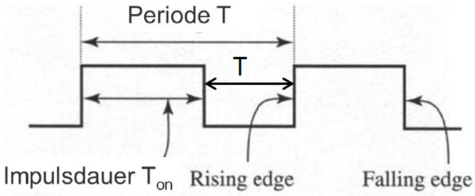
4.4 Aufbau logischer Gatter

Funktion	Buffer	NOT Nicht Inverter	AND Und Konjunktion	NAND Nicht Und	OR Oder Disjunktion	NOR Nicht Oder	EXOR Exklusiv Oder Antivalenz	XNOR Nicht Ex. Oder Äquivalenz
Formel	a a	$\overline{a}$ $\overline{a}$	$a * b$ $a \wedge b$	$\overline{a * b}$ $\overline{a \wedge b}$	$a + b$ $a \vee b$	$\overline{a + b}$ $\overline{a \vee b}$	$a \oplus b$ $a\$b$	$\overline{a \oplus b}$ $\overline{a\$b}$
Symbol								
WHT (0,0) (0,1) (1,0) (1,1)	 0 1 	 1 0 	 0 0 0 1	 1 1 1 0	 0 1 1 1	 1 0 0 0	 0 1 1 0	 1 0 0 1
								
Trans	4	2	6	4	6	4	8	10

5 Sequentielle Systeme

5.1 Taktsignal

Weil sequentielle Schaltungen in der Regel synchron arbeiten, muss eine Referenz zur Einhaltung der Synchronität definiert werden. Das Taktsignal ist ein binäre Signal, das in regelmässiger Abfolge zwischen zwei Zuständen hin und her pendelt.

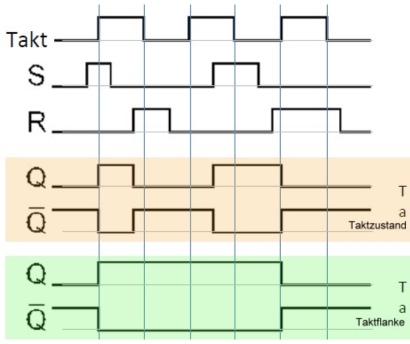


$$f = \frac{1}{T}\ [\text{Hz}]$$

5.2 Flipflops und Latches

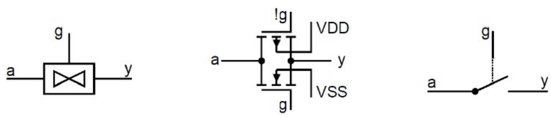
5.2.1 Unterschied Flipflop und Latch

Taktzustandsgesteuerte Systeme haben den Nachteil, dass in ihrer transparenten Phase auch asynchrone Schaltvorgänge stattfinden können. Echt synchrone Systeme ändern ihren Zustand nur bei der aktiven Flanke des Taktsignals. Genau in diesem Moment und sonst nie wird das Eingangssignal bei einem Speicherelement in den Speicher übertragen. Nur beim taktflankengesteuerten System wechseln die Ausgänge immer genau zum Zeitpunkt der aktiven Taktflanke. Beim taktzustandsgesteuerten System sind während der transparenten Phase auch Zustandsänderungen zwischen zwei Taktflanken möglich.
Taktflankengesteuerte Speicherelemente werden Flip-Flops genannt. Taktzustandsgesteuerte Speicherelemente werden Latches genannt

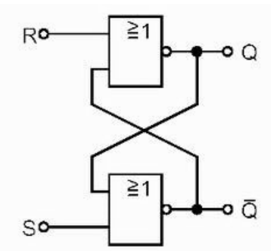


5.2.2 Transmission Gate

Das Transmission Gate ist von seiner Funktion her ein einfacher Schalter, der Signale sowohl auf positivem, als auch auf negativem Pegel schalten kann.

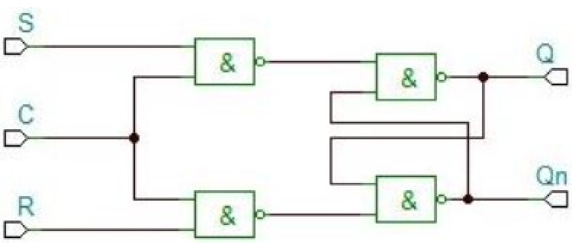


5.2.3 RS-Latch

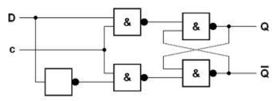


S	R	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	1	0
1	1	ungültig	

5.2.4 RS-Latch mit Clock

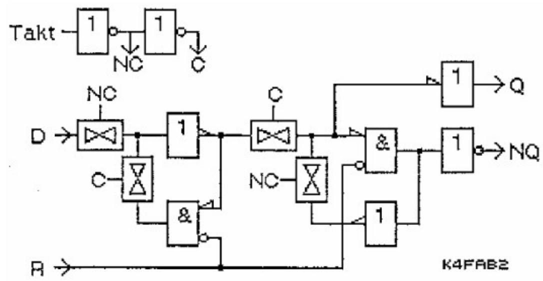


5.2.5 D-Latch



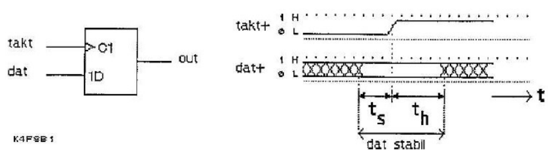
D	C	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	0	1
1	0	Q	$\overline{Q}$
1	1	0	1

5.2.6 D-Flipflop mit Reset



5.2.7 Setup- und Holdtime

- t_s= setup time → Minimale Zeitspanne, während der ein Datensignal vor einer aktiven Clockflanke stabil sein muss, um zuverlässig eingelesen zu werden.
- t_H= hold time → Minimale Zeitspanne, während der ein Datensignal nach einer aktiven Clockflanke noch stabil bleiben muss, damit der Einlesevorgang des Datensignals erfolgreich abgeschlossen werden kann.



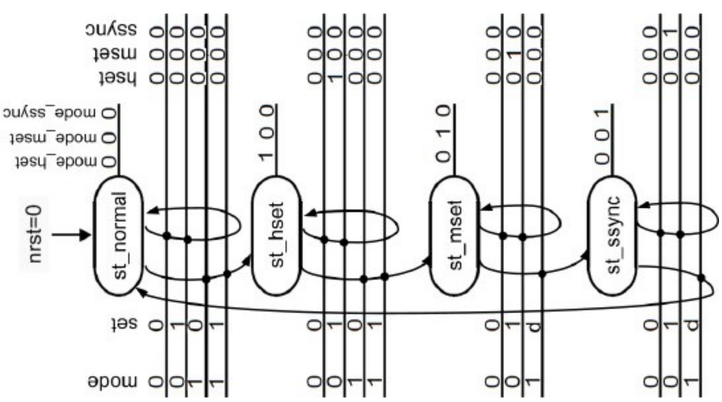
5.3 Beschreibung sequentieller Systeme

- S: Menge der Zustände mit Zustandsaktionen
- I⊆S: Initalzustände
- T: Kombinatorische Übergangsrelation
- E: Eingangssignale
- A: Ausgangssignale

5.3.1 Tabellarische Beschreibung

Aktzeller Zustand	Eingangs-var.		Folge-Zustand	Ausgangsvariablen					
state_name	mode	set	state_name	mod_hset	mod_mset	mod_ssync	hset	mset	ssync
st_normal	0	0	st_normal	0	0	0	0	0	0
	0	1	st_normal				0	0	0
	1	0	st_hset				0	0	0
	1	1	st_hset				0	0	0
st_hset	0	0	st_hset	1	0	0	0	0	0
	0	1	st_hset				1	0	0
	1	0	st_mset				0	0	0
	1	1	st_mset				0	0	0
st_mset	0	0	st_mset	0	1	0	0	0	0
	0	1	st_mset				0	1	0
	1	d	st_ssync				0	0	0
st_ssync	0	0	st_ssync	0	0	1	0	0	0
	0	1	st_ssync				0	0	1
	1	d	st_normal				0	0	0

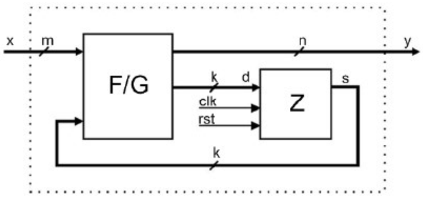
5.3.2 Grafische Beschreibung (Zustandsdiagramm)



Diese Beispiele visualisieren das sequentielle System des Watch-Controllers.

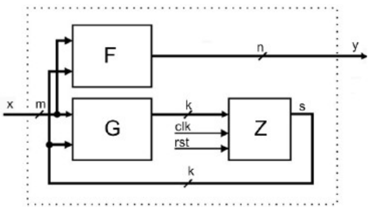
5.4 Strukturen der Finite State Machine

5.4.1 Grundstruktur



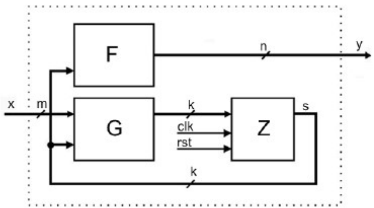
- s: Zustand, Zustandsvektor
- x: Primäre Eingänge, Eingangsvektor
- y: Primäre Ausgänge, Ausgangsvektor
- d: Speicheransteuerung, Folgezustand
- m: Anzahl Eingänge
- n: Anzahl Ausgänge
- k: Anzahl Speicherstellen
- F: Funktion für die Ausgänge
- G: Funktion für die Speicheransteuerung
- Z: Zustandsspeicher
- I: Index der aktuellen Taktflanke (Taktflankennummer)

5.4.2 Mealy-System



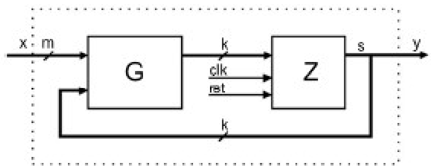
Ausgänge hängen vom momentanen Zustand und den aktuellen Eingängen ab

5.4.3 Moore-System



Ausgänge hängen nur vom momentanen Zustand ab und ändern mit der Clock Flanke

5.4.4 Medwedjew-System



Die primären Ausgänge entsprechen dem Zustandsvektor

5.5 Zustandscodierung

- Binär: Alle Zustände werden der Reihe nach durchnummeriert.
- ONE-HOT: Nur eine Speicherstelle im Code hat jeweils den Wert 1. Alle anderen besitzen den Wert 0 (z.B. 001, 010, 100)
- ONE-COLD: Nur eine Speicherstelle im Code hat jeweils den Wert 0. Alle anderen besitzen den Wert 1 (z.B. 110, 101, 011)

5.6 Synthese von Zustandsmaschinen

1. Zustandsdiagramm aufstellen
2. Zustandskodierung zuweisen
3. Zustandstabelle nach festen Regeln aufstellen
4. Speicheransteuer-Funktionen bestimmen
5. Ausgangs-Funktionen bestimmen