1. KOMBINATORISCHE LOGIK

Einfache logische Operationen ohne Speicher.

Die Ausgänge ändern sich nur in Abhängigkeit von den Eingängen. Jeder Ausgang y_i lässt sich durch eine boolesche Funktion f_i aller Eingänge beschreiben: $y_i = f_i(x_0, x_2, \cdots, x_n)$

Für N Eingänge gibt es 2^N Eingangskombinationen.

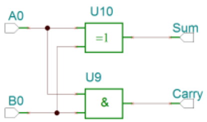
Logische Operationen

Function	Boolean Algebra [1]	IEC 60617-12 since 1997
AND	A & B	&
OR	A # B	≥ 1
Buffer	A	1
XOR	A $ B	= 1
NOT	!A	1
NAND	!(A & B)	&
NOR	!(A # B)	≥ 1
XNOR	!(A $ B)	= 1

A	B	$!A$	$A\,\&\,B$	$A\,\#\,B$	$A\,\$\,B$
0	0	1	0	0	1
0	1	1	0	1	0
1	0	0	0	1	0
1	1	0	1	1	1

1-Bit Halb-Addierer (HA)

Bildet die Summe und Carry von zwei Bits.



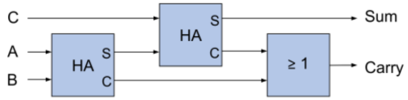
A0	B0	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Sum = *A0 $ B0*

Carry = *A0 & B0*

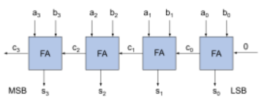
1-Bit Voll-Addierer (FA)

Bildet die Summe und Carry von zwei Bits und vorherigem Carry.



C	A	B	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

4-Bit Addierer



2. SEQUENTIELLE LOGIK

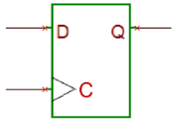
Sequentielle Logik hat gegenüber der Kombinatorischen Logik mehrere Zustände und enthält Speicher. Grundelement dafür sind D-Flip-Flops.

D-Flip-Flop

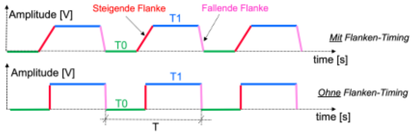
Wert am Eingang D wird gespeichert und an den Ausgang Q übertragen, wenn C von 0 auf 1 wechselt.

Ein Flip-Flop hat 2 Zuständ.

N Flip-Flops haben 2^N Zustände.



Clock Signal



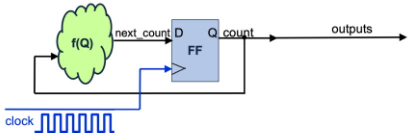
Periode $T = T_0 + T_1 [s]$

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

Duty Cycle $\frac{T_1}{T}$

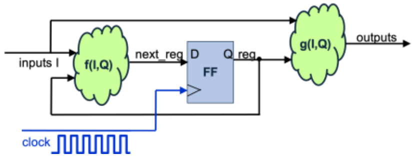
Zähler (Counter)

Reihenfolge der Zustände und Zustand vom Ausgang hängt vom internen Zustand/Logik ab.



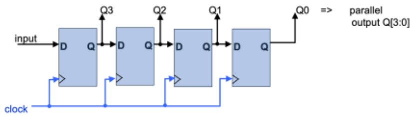
Zustandsautomaten (Finite State Machine / FSM)

Der Ausgang ist abhängig vom Input und dem Status des Speichers. Der FF-Ausgangswert Q entspricht dem Zustand des Automaten.



Schieberegister

Jedes FF verzögert den Input um einen Takt. Schieberegister können paralell oder in Serie sein. Kann Rückkopplung enthalten



Zustandsdiagramm (Bsp. Ampel)

Um ein Zustandsautomat von einem Zustandsdiagramm zu bauen, muss man die Funktion $f(Q)$ für den nächsten Zustand und die Funktion $g(Q)$ für den Output ermitteln.

Zustandslogik / Ausgangslogik

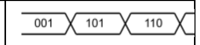
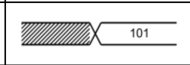
Q1	Q0	D1	D0
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

state		outputs		
Q1	Q0	red_on	yellow_on	green_on
0	0	1	0	0
0	1	1	1	0
1	0	0	0	1
1	1	0	1	0

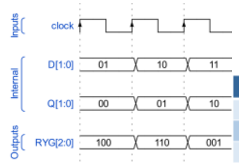
D0 = !D0, D1 = Q0 $ Q1

red = !Q1, yellow = Q0, green = !Q0 & Q1

Zeitverlaufsdiagramme / Vektoren

Signal wechselt von 0 nach 1	
Signal wechselt von 1 nach 0	
Vektor (= Bus oder Signalgruppe) wechselt den Wert (MSB-LSB)	
Vektor wechselt von unbekanntem in definierten Wert	
Vektor wechselt von bekanntem in undefinierten Wert	

Bsp.:

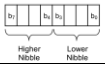


Q1	Q0	D1	D0
0	0	0	1
0	1	1	0
1	0	1	1

3. ZAHLENSYSTEME

4 Bit -> Nibble

8 Bit -> Byte (Octet)



Dec	Bin	Hex	Oct
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

Stellenwertsysteme

Um in Zahlen in dezimal umzurechnen, kann folgende formel verwendet werden, wobei i die Stelle (begonnen bei 0), b die Basis und a_i die Ziffer an Stelle i:

$$Z = \sum a_i \cdot b^i$$

Das Schieben um s Stellen wird wie folgt berechnet:

$$Z_{neu} = Z \cdot \lfloor b^s \rceil$$

Addition

Erster Summand	1 1 0 1. 0
Zweiter Summand	+ 1 0 1 1. 1
Carry	1 1 1 1
Sum	1 1 0 0 0. 1

Subtraktion

Minuend	1 1 0 1. 0
Subtrahend	- 1 0 1 1. 1
Carry	1 1 1.
Differenz	0 0 0 1. 1

Multiplikation

Faktoren	1 0 1 x 1 1 1 0
	1 1 1 0
	+ 0 0 0 0
	+ 1 1 1 0
Carry	1 1
Produkt	1 0 0 0 1 1 0

Division

Divisor ÷ Divisor 1 1 0 1 1 0 ÷ 1 0 1 0 = 1 0 1

```
  1 1 0 1 1 0 ÷ 1 0 1 0 = 1 0 1
- 1 0 1 0
  0 0 1 1 1
  - 0 0 0 0
    0 1 1 1 0
    - 1 0 1 0
      0 1 0 0
```

0111 < 1010
=> 0 in Quotient

Rest 0 1 0 0

Darstellung negativer Zahlen

Binär	Dezimal	Sign+Magn.	Einerkomp.	Zweierkomp.	Exzess-8
1111	15	−7	−0	−1	+7
1110	14	−6	−1	−2	+6
1101	13	−5	−2	−3	+5
1100	12	−4	−3	−4	+4
1011	11	−3	−4	−5	+3
1010	10	−2	−5	−6	+2
1001	9	−1	−6	−7	+1
1000	8	−0	−7	−8	0
0111	7	+7	+7	+7	−1
0110	6	+6	+6	+6	−2
0101	5	+5	+5	+5	−3
0100	4	+4	+4	+4	−4
0011	3	+3	+3	+3	−5
0010	2	+2	+2	+2	−6
0001	1	+1	+1	+1	−7
0000	0	+0	+0	0	−8

2er-Komplementbildung (Vorzeichenwechsel)

- **Verfahren:**

	+ 2 → -2	- 2 → +2
	0 0 0 0 0 0 1 0	1 1 1 1 1 1 1 0
• invertieren:	1 1 1 1 1 1 0 1	0 0 0 0 0 0 0 1
• 1 addieren:	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1
	1 1 1 1 1 1 1 0	0 0 0 0 0 0 1 0

- **Spezialfälle:**

	0 → 0	- 128 → Überlauf
	0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0
1er-Komplement	1 1 1 1 1 1 1 1	0 1 1 1 1 1 1 1
	0 0 0 0 0 0 0 1	0 0 0 0 0 0 0 1
	0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0

Verarbeitungsbreite

Register	Bezeichnung	Max uint	Max int
4 Bit	Nibble	0...15	-8...7
8 Bit	Byte	0...255	-128...127
16 Bit	Word	0...65'353	-32'768...32'767
32 Bit	Double Word	$0...4.29 \cdot 10^9$	$-2.15 \cdot 10^9 ... 2.15 \cdot 10^9$
64 Bit	Logn Word	$0...1.84 \cdot 10^{19}$	$-9.22 \cdot 10^{18} ... 9.22 \cdot 10^{18}$

4. INFORMATIONSTHEORIE

Discrete Memoryless Source (DMS)

Eine diskrete gedächtnislose Quelle (DMS) ist ein Modell, das eine Informationsquelle beschreibt, die eine endliche Menge von Symbolen $S = \{s_1, s_2, \ldots, s_m\}$ mit bestimmten Wahrscheinlichkeiten $P = \{p(s_1), p(s_2), \ldots, p(s_m)\}$ erzeugt, wobei jedes Symbol unabhängig von den vorherigen Symbolen ausgewählt wird.

Binary Memoryless Source (BMS)

Eine binäre gedächtnislose Quelle (BMS) ist ein Spezialfall der DMS, bei dem die Symbolmenge auf zwei Symbole $S = \{0, 1\}$ beschränkt ist. Die Wahrscheinlichkeiten für die beiden Symbole sind $P = \{p(0), p(1)\}$, wobei $p(0) + p(1) = 1$ gilt.

Auftretenswahrscheinlichkeit

Die Auftretenswahrscheinlichkeit eines Symbols s_i in einer Nachricht wird durch die relative Häufigkeit

$$p(s_i) = \frac{N_i}{N}$$

bestimmt, wobei N_i die Anzahl der Vorkommen des Symbols s_i und N die Gesamtanzahl der Symbole in der Nachricht ist.

Informationsgehalt

Der Informationsgehalt $I(s_i)$ eines Symbols s_i wird durch die Formel

$$I(s_i) = -\log_2(p(s_i)) \text{ [Bit]}$$

definiert. Er gibt an, wie viel Information das Symbol liefert, wobei seltenere Symbole mehr Information enthalten.

Mittlere Informationsgehalt (Entropie)

Der mittlere Informationsgehalt $H(X)$ einer diskreten Zufallsvariable X mit den möglichen Symbolen $S = \{s_1, s_2, \ldots, s_m\}$ und den Wahrscheinlichkeiten $P = \{p(s_1), p(s_2), \ldots, p(s_m)\}$ wird durch die Formel

$$H(X) = -\sum_{i=1}^{N} p(s_i) \cdot \log_2(p(s_i)) \left[\frac{\text{Bit}}{\text{Symbol}}\right]$$

definiert. Er gibt den durchschnittlichen Informationsgehalt pro Symbol an.

Wenn alle Symbole die gleiche Wahrscheinlichkeit haben, vereinfacht sich die Formel zu:

$$H(X) = \log_2(N) \left[\frac{\text{Bit}}{\text{Symbol}}\right]$$

By Justin Iven Müller (2025)
github.com/JustinIven/zhaw-cheatsheets