Bits and bytes

Stellenwertsysteme

Steinzeit IIII III

Römer MMXXIII

Dezimalsystem $2023 = 2 \cdot 1000 + 0 \cdot 100 + 2 \cdot 10 + 3 \cdot 1$

decem = 10 $= 2 \cdot 10^3 + 0 \cdot 10^2 + 2 \cdot 10^1 + 3 \cdot 10^0$

Stellenwertsystem zur Basis 10: zehn Ziffern 0, 1, 2, 3, ..., 9

↳ Wert einer Ziffer hängt von ihrer Stelle ab.

- Nullen sind wichtig: $2023 \neq 223$
- Warum 10er-Potenzen: 10 Finger

Fünfersystem = das Stellenwertsystem zur Basis 5

fünf Ziffern 0, 1, 2, 3, 4.

$$\boxed{4}\boxed{1}\boxed{0}\boxed{2}\boxed{3} = \underbrace{4 \cdot 625}_{2500} + 1 \cdot 125 + 0 \cdot 25 + 2 \cdot 5 + 3 \cdot 1$$

$$(41023)_5 = (2638)_{10}$$

Aufgabe: $(2023)_{10} = (31043)_5$ 263

Ausprobieren: $2023 = \underbrace{3 \cdot 625}_{1875,\ \text{d.h. es fehlen}} + 1 \cdot 125 + 0 \cdot 25 + 4 \cdot 5 + 3 \cdot 1$

↳ fehlen noch 23 fehlen 3

noch $2023 - 1875 = 148$

Systematisches Verfahren / Algorithmus

Idee: Im 10er-System ist die letzte Ziffer der Rest bei Division durch 10.

$2023 : 5 = 404$ Rest 3

$404 : 5 = 80$ Rest 4

$80 : 5 = 16$ Rest 0

$16 : 5 = 3$ Rest 1

$3 : 5 = 0$ Rest 3 (fertig)

$$(3\,1\,0\,4\,3)_5 = (2023)_{10}$$

Warum funktioniert das Verfahren?

$$\begin{aligned}
2023 &= 3 + 5 \cdot 404 \\
&= 3 + 5 \cdot (4 + 5 \cdot 80) \\
&= 3 + 5 \cdot (4 + 5 \cdot (0 + 5 \cdot 16)) \\
&= 3 + 5 \cdot (4 + 5 \cdot (0 + 5 \cdot (1 + 5 \cdot 3))) \\
&= 3 + 5 \cdot 4 + 5^2 \cdot 0 + 5^3 \cdot 1 + 5^4 \cdot 3 \\
&= 3 \cdot 5^4 + 1 \cdot 5^3 + 0 \cdot 5^2 + 4 \cdot 5 + 3 \\
&= (3\ 1\ 0\ 4\ 3)_5
\end{aligned}$$

Kurzform:

		Rest
2023		3
↓ durch 5		
404	→	4
80	→	0
16	→	1
3	→	3
0		

Aufgabe: 4321 in's Fünfersystem umwandeln.

4321	1
864	4
172	2
34	4
6	1
1	1
0	0
0	0

$\boxed{:5} = \boxed{\cdot \frac{1}{5}} = \boxed{\cdot \frac{2}{10}}$

$\cancel{00}(11424 1)_5 = (4321)_{10}$

Primarschule

Zähle im 5er-System von 0 bis $(32)_{10}$:

$(0)_5$	$(1)_5$	$(2)_5$	3	4
10	11	12	$(13)_5$	14
20	21	22	23	24
30	31	32	33	34
40	41	42	43	44
100	101	102	103	104
110	111	112		

$1 \cdot 5 + 3 \cdot 1 = 8$

$1 \cdot 5^2 + 1 \cdot 5 + 2 \cdot 1 = 32$

Kleines Einsplus eins und Einmaleins

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	10
2	2	3	4	10	11
3	3	4	10	11	12
4	4	10	11	12	13

·	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	11	13
3	0	3	11	14	22
4	0	4	13	22	31

Schriftliche Addition

$$\begin{array}{r} 43341 \\ +\ 41402 \\ \hline 140243 \end{array}$$

Schriftliche Multiplikation

$$123 \cdot 304$$

$$\begin{array}{r} 492 \\ 0 \\ 1102 \\ \hline 44002 \end{array}$$

Wiederholung: Stellenwertsysteme

Prüfung 4. April

Bsp: Fünfersystem

2	6	3

keine Zahl im Fünfersystem, da nur die Ziffern 0, 1, 2, 3, 4 erlaubt

2	4	1

$= 2 \cdot 25 + 4 \cdot 5 + 1 \cdot 1$

$= 71$

wie viele 1er $1 = 5^0$

wie viele 5er $5 = 5^1$

wie viele 25er $25 = 5^2$

$(241)_5 = (71)_{10}$

$(600)_{10} = (\quad)_5$

naiv: $(4400)_5 = \underbrace{4 \cdot 125}_{500} + \underbrace{4 \cdot 25}_{100} + 0 \cdot 5 + 0 \cdot 1 = 600$

600	Rest	0
:5 ↓		
120	Rest	0
:5 ↓		
24	Rest	4
:5 ↓		
4	Rest	4
:5 ↓		
0		

→ 4400

$(1000)_{10} = (13000)_5$

1000	0
200	0
40	0
8	3
1	1
0	

13 000

Binärsystem (= Dualsystem = Zweiersystem)

‖ ist in der Informatik am wichtigsten

Stellenwertsystem zur Basis 2, Ziffern 0, 1

‖ ‖

Aus An

$(1\,00\,11)_2 = 1\cdot 2^4 + 0\cdot 2^3 + 0\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0$

$= 16 \qquad + 2 + 1 = 19$

$(11\,000\,1)_2 = 32 + 16 + 1 = 49$

$(11\,001)_2 = 16 + 8 + 1 = 25 = (25)_{10}$

$(2023)_{10} = (\underbrace{1111\,11}_{6}\,00\,111)_2$

2023	1
1011	1
505	1
252	0
126	0
63	1
31	1
15	1
7	1
3	1
1	1
0	

256 128 64 32 16 8 4 2 1

$$(101101110)_2 = (366)_{10}$$

$$256 + 64 + 32 + 8 + 4 + 2 = 366$$

Kleine Einspluseins

+	0	1
0	0	1
1	1	10

Kleine Einmaleins

·	0	1
0	0	0
1	0	1

```
  1 1 1 1 1 1 0 0 1 1
+ 1 0 0 1 1 1 0 0 1 0
---------------------
```

```
1011 · 1101
-----------
  1011000
   101100
    00000
+ 1011
-------
 10001111
```

Zähle im Binärsystem von 0 bis 40

```
  0
  1
 10
 11
100
101
110
111
1000
1001
1010
```

— $1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 10$

schriftliche Addition:

binär	dezimal
1010	10
0111	7
11	
10001	17

Ziel: Bringe dem Computer schriftliches Addieren bei

Wie Computer addieren

Bit = binary digit = Binärziffer, d.h. 0 oder 1.

Byte = Folge von 8 Bit = 8-stellige Binärzahl, z.B. 0100 1101

Wieviele verschiedene Werte kann ein Byte annehmen?

An jeder der 8 Positionen gibt es zwei Möglichkeiten,

also $2 \cdot 2 \cdot 2 \cdots 2 = 2^8 = 256$ Werte:

alle Binärzahlen von 0000 0000 bis $(1111\ 1111)_2 = (255)_{10}$

Logische Schaltungen / Verknüpfungen

AND	logische Und	Konjunktion
OR	logisches Oder	Disjunktion
NOT	logische Verneinung	Negation

Erinnerung:

AND	falsch	wahr
falsch	falsch	falsch
wahr	falsch	wahr

IS jetzt schreibe 0 für falsch (oder AUS)
1 für wahr (oder AN)
↑ Spannung / Strom.

a	b	a AND b	a OR b
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

Wahrheitstafel

a	NOT a
0	1
1	0

logische Gatter

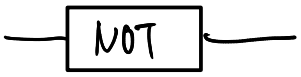
in Logisim-Software

AND-Gatter

zwei Eingänge Ausgang

OR-Gatter

NOT-Gatter



Video „Exploring how computers work" bis 4:15

---

a AND b mathematische Schreibweise $a \wedge b$ (^ stilisiertes A)

a OR b $a \vee b$ (v von lateinisch vel = oder)

NOT a $\overline{a}$

Aufgabe:

Hilfsspalten

a	b	$\overline{a \wedge b}$	$\overline{a \vee b}$	$\overline{a} \wedge \overline{b}$	$\overline{a} \vee \overline{b}$	a∧b	a∨b	$\overline{a}$	$\overline{b}$
0	0	1	1	1	1	0	0	1	1
0	1	1	0	0	1	0	1	1	0
1	0	1	0	0	1	0	1	0	1
1	1	0	0	0	0	1	1	0	0

Beobachtungen:

- erste und vierte Spalte stimmen überein, d.h.

$$\overline{a \wedge b} = \bar{a} \vee \bar{b}$$

- zweite und dritte Spalte gleich, d.h.

$$\overline{a \vee b} = \bar{a} \wedge \bar{b}$$

} für alle Belegungen von a und b

} die Morgansche Gesetze

Es gibt viele andere solche Gesetze, die man genauso beweisen kann,

siehe Wikipedia, Boolesche Algebra.

z.B. Assoziativgesetze

$$a \wedge (b \wedge c) = (a \wedge b) \wedge c$$
$$a \vee (b \vee c) = (a \vee b) \vee c$$

} kann Klammern weglassen

$a \wedge b \wedge c$

$a \vee b \vee c$

unabhängig von Setzung der Klammern.

Aufgabe zur Siebensegmentanzeige

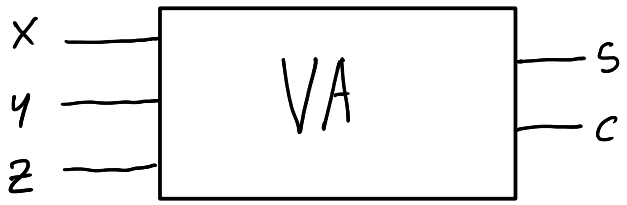
	a	b	c
	0	0	0
	0	0	1
	0	1	0
	0	1	1
	1	0	0
	1	0	1
	1	1	0
	1	1	1

evtl. Video $\overline{\bar{a} \wedge \bar{b}} = \bar{\bar{a}} \vee \bar{\bar{b}} = a \vee b$

20.02.2023 Video bis 9:20 (unser Ziel erwichard um 7:29)

Disjunktive Normalform und Volladdierer (VA)

Volladdierer addiert 3 Bits



x	y	z	c	s		$\bar{x} \wedge \bar{y} \wedge z$
0	0	0	0	0		0
0	0	1	0	1	$\bar{x} \wedge \bar{y} \wedge z$	1
0	1	0	0	1	$\bar{x} \wedge y \wedge \bar{z}$	0
0	1	1	1	0		0
1	0	0	0	1	$x \wedge \bar{y} \wedge \bar{z}$	0
1	0	1	1	0		0
1	1	0	1	0		0
1	1	1	1	1	$x \wedge y \wedge z$	0

Die disjunktive Normalform (DNF) liefert einen logischen Ausdruck für jede Output-Spalte einer beliebigen Wahrheitstabelle.

Gesucht: logischer Ausdruck für s-Spalte.

Wann wird $\bar{x} \wedge \bar{y} \wedge z$ wahr?

Nur dann, wenn alle drei Eingänge/Argumente $\bar{x}$, $\bar{y}$ und z wahr sind,

d.h. wenn x falsch, y falsch und z wahr sind

d.h. im Fall $x = 0,\ y = 0,\ z = 1$.

Der rote Ausdruck wird nur in der „roten Zeile" wahr = 1.

Dasselbe gilt für jede Farbe.

Bildet man die Ver-Oderung / Disjunktion der farbigen Ausdrücke, so wird dieser Ausdruck genau in den farbigen Zeilen wahr = 1,

d.h. $S = (\bar{x} \wedge \bar{y} \wedge z) \vee (\bar{x} \wedge y \wedge \bar{z}) \vee (x \wedge \bar{y} \wedge \bar{z}) \vee (x \wedge y \wedge z)$

disjunktive Normalform

für S-Spalte

Analog für C (freiwillige Aufgabe)

$\rightsquigarrow$ kann Volladdierer per logisim bauen.

Alternativ kann man den Volladdierer aus bereits konstruierten Bauteilen bauen:

S ist genau dann 1, wenn ungeradzahlig viele Eingänge 1.

$S = (x \text{ XOR } y) \text{ XOR } z$

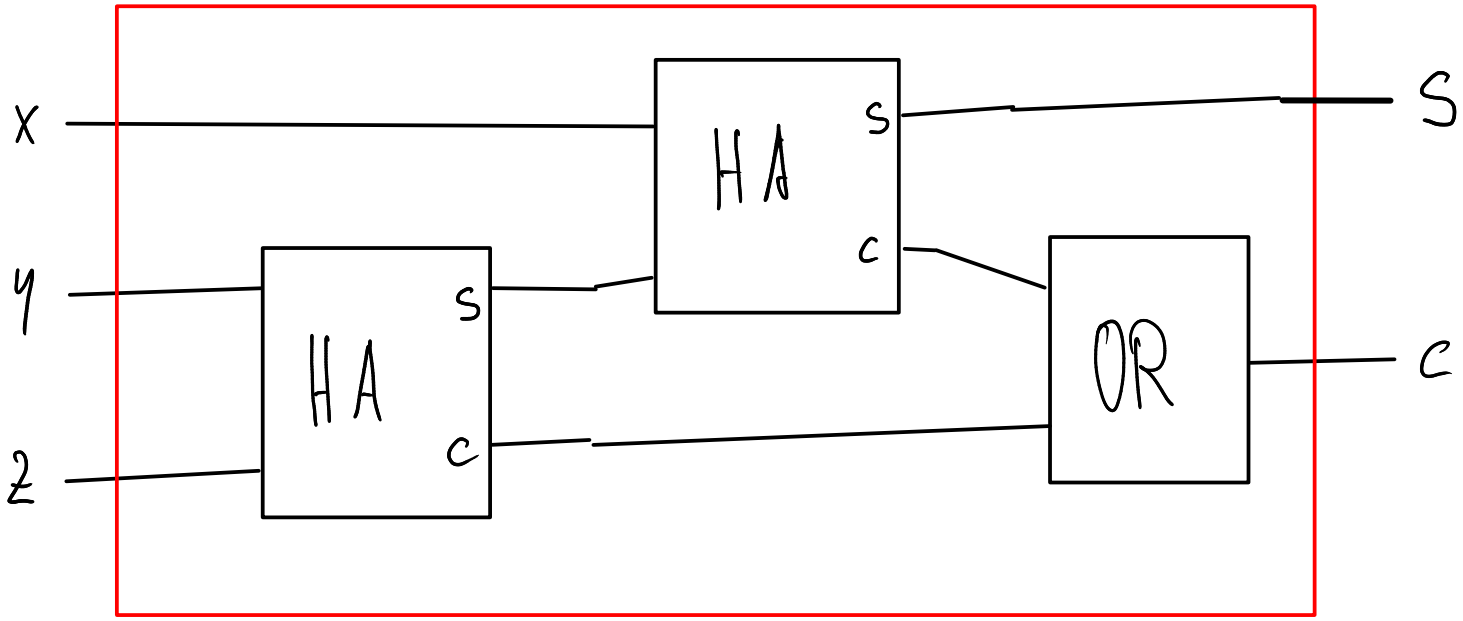
↑ bitte prüfen

C ist genau dann 1, wenn mindestens zwei Eingänge 1 sind.

$$c = (y \wedge z) \vee (x \wedge z) \vee (x \wedge y)$$

---

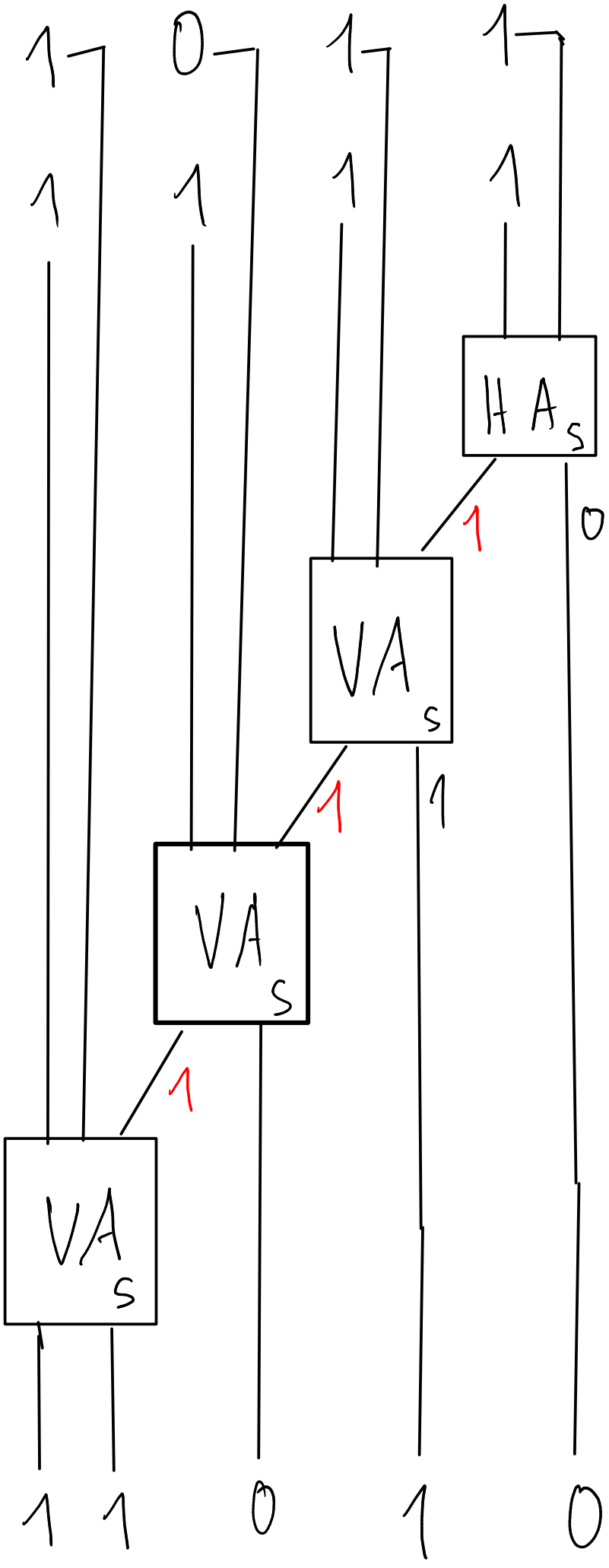
dritte Möglichkeit, VA zu bauen:



Volladdierer

4-Bit-Addierer

4-Bit-Addierer



$$
\begin{array}{cccc}
1 & 0 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & \\
\hline
1 & 1 & 0 & 1 & 0
\end{array}
$$