BiROP
5.10. V.

Lit.: · Seifart: Digitale Schaltungen (zur Wiederholung DST)

· Pirsch: Architekturen der digitalen Signalverarbeitung

· Hwang: Computer Arithmetic
· Carvanagh: Digital Computer Arithmetic
· Waser, Flynn: Introduction to Arithmetic for digital Systems Designers

Allgemeine Zahlendarstellung mit Stellenwertsystem

dargestellter Wert Z (Größe eines Wertes) (typisch. dezimale Angabe)

Stellenwertsystem

ganzzahlige Stellenzahl $n+m$

ganzzahliger Stellenindex i mit $-m \leq i \leq n-1$

m: Anzahl Nachkommastellen

n: Anzahl Vorkommastellen

r: Basis

Ziffernwert (Digits) x_i (ganzzahlig)

Stellenwertigkeiten (Weighting Factors) r^i (Potenz d. Basis)

positive ganzzahlige Basis (Radix) r mit $r \geq 2$

additiver Beitrag einer Stelle

$$X_i = x_i \cdot r^i$$

$-m$: niederwertigste Stelle (least significant Digit = LSD)

$n-1$: höchstwertigste Stelle (most significant Digit = MSD)

Hinweis: Dualsystem mit $r=2$ ⇒ LSB / MSB

1. Nichtredundantes Stellenwertsystem

für jeden repräsentierbaren Wert gibt es nur <u>eine</u> <u>Darstellungsmöglichkeit</u>.

Bedingung nichtredundant: Ziffernwerte $x_i \in (0, 1, \dots, r-1)$
es gibt genau r unterschiedliche Ziffernwerte

1. Redundantes Stellenwertsystem

für die repräsentierbaren Werte können <u>mehrere</u> <u>Darstellungsmöglichkeiten</u> existieren.

Bedingung: <u>Anzahl unterschiedlicher Ziffernwerte</u>, Anzahl $\{x_i\} > r$
↳ größer als Basis

Bsp. Basis $r = 2$ Ziffernwerte $x_i \in (-1, 0, 1)$

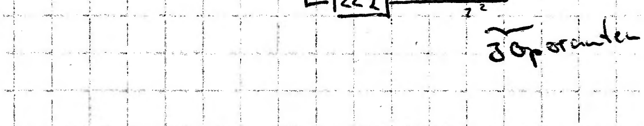
$n = 3$ $m = 0$

		2^2	2^1	2^0
Darstellung Zahl	$Z = 1$	0	0	1
	oder	0	1	-1
	oder	1	-1	-1

Anwendung: - Unterbrechung von Übertragsketten bei Addition und/oder Subtraktion
- Vereinfachung von Konstantenmultiplikation

Bsp. Dualsystem $a \cdot 7$

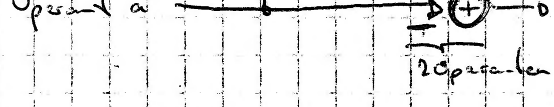
	2^2	2^1	2^0
$7 \hat{=}$	1	1	1



$\overbrace{3 \text{ Operanden}} \hat{=} 2$ Addierern

7 als Redundantes Stellenwerte mit $x_i \in (-1, 0, 1)$

	2^3	2^2	2^1	2^0
$7 \hat{=}$	1	0	0	-1

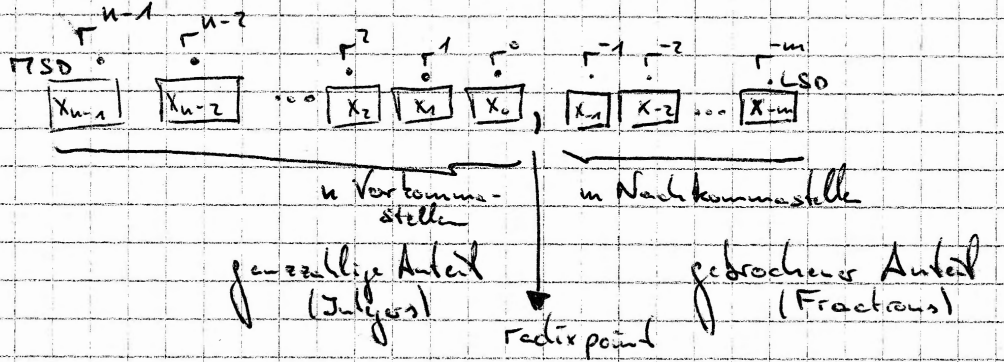


$\overbrace{2 \text{ Operanden}} \hat{=} 1$ Subtrahierer

Ermittlung d. dargestellten Wertes Z aus den Ziffern- und Stellenwerten (gilt für nichtredundantes und redundantes Stellen-wertsystem)

Z: dezimale Zahl (Angabe)

$$Z = \sum_{i=-m}^{n-1} X_i = \sum_{i=-m}^{n-1} x_i \cdot r^i$$



Dezimalsystem

$r = 10 \qquad z_i \in (0, 1, \ldots, 9)$

nichtredundantes Stellenwertsystem

negative Dezimalzahlen

Vereinbarung: Minus-Zeichen vor höchstwertig. Stelle

Kennzeichnung: $(\text{Ziffernfolge})_{10}$ Wert $Z = \sum_{i=-m}^{n-1} z_i \cdot 10^i$

Dualsystem oder natürliches Binärsystem

$r = 2 \qquad d_i \in (0, 1)$

nichtredundant Wert $Z = \sum_{i=-m}^{n-1} d_i \cdot 2^i$

Vorteil 1 Stelle = 1 Bit = 1 Leitung

$(\text{Ziffernfolge})_2$

Bsp. positive Dualzahl $(1110{,}011)_2$

$2^3 \quad 2^2 \quad 2^1 \quad 2^0 \; . \; 2^{-1} \quad 2^{-2} \quad 2^{-3}$

$n = 4$ $\qquad$ $m = 3$

$1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1$

dezimale Wert: $Z = 14{,}375$

BinROp
12.10. V

1) Umwandlung zwischen Stellenwertsystemen unterschiedlicher Basis

- getrennte Umwandlung von ganzzahligem Anteil (Vorkommastellen) und gebrochenen Anteil (Nachkommastellen).
- Umwandlung der Vorkommastellen zu den Vorkommastellen der neuen Basis:

→ fortlaufende Division durch den Wert der Basis des neuen Stellenwertsystems.

→ auftretende Reste stellen in der Reihenfolge von r^0 beginnend bis r^{n-1} (n: Anzahl Vorkommastellen des neuen Stellenwertsystems) die Ziffernwerte des neuen Stellenwertsystems dar.

→ letzte erforderliche Division ist die Division mit dem Divisionsergebnis von Null, der hierbei noch übrige Rest gibt den Ziffernwert der Stelle r^{n-1} an.

Bsp.: $(38)_{10} = (?)_2$ $r_{alt} = 10$ $r_{neu} = 2$

			Rest
2^0	38	: 2	→ 0
2^1	= 19		1
2^2	= 9		1
2^3	= 4		0
2^4	= 2		0
2^5	= 1		1
	= 0		

$(38)_{10} = (100110)_2$

Bsp: $(100110)_2 = (?)_{10}$

Stelle 10^0 →

$$
\begin{array}{r}
100110 : 1010 = 011 \\
-0000 \\
\hline
10011 \\
-1010 \\
\hline
010010 \\
-1010 \\
\hline
01000
\end{array}
\rightarrow \text{Rest } 8
$$

Stelle 10^1

$$
\begin{array}{r}
011 : 1010 = 0 \\
-000 \\
\hline
011
\end{array}
\rightarrow \text{Rest } 3
$$

$\Rightarrow 38_{10}$

- Umwandlung der Nachkommastellen zu den Nachkommastellen der neuen Basis
  - → fortlaufende Multiplikation mit dem Wert der Basis des neuen Stellenwertsystems
  - → auftretende Überträge (Vorkommastellen der Multiplikationsergebnisse) stellen in der Reihenfolge der Stellenwerte von r^{-1} beginnend bis r^{-m} (m: Anzahl Nachkommastellen im neuen Stellenwertsystem) die Ziffernwerte des neuen Stellenwertsystems dar.
  - → fortlaufende Multiplikation wird immer ohne die eventuell auftretenden Überträge ausgeführt.
  - → Abbruch der fortlaufenden Multiplikation bei einem Multiplikationsergebnis mit einem Wert der Nachkommastellen von Null oder bei einer ausreichenden Stellenzahl bestimmt durch die geforderte Genauigkeit.

Bsp: $(0,375)_{10} = (?)_2$ $r_{alt} = 10$ $r_{neu} = 2$ Mult. mit 2

$0,375 \cdot 2 \rightarrow 0,750 \cdot 2 \rightarrow 1,50 \rightarrow 0,5 \cdot 2 \rightarrow 1,0$

[redacted]

$0,\ldots \rightarrow 0$ $1,\ldots \rightarrow 1$ $1,\ldots \rightarrow 1$ Abbruch

$\rightarrow ,011$

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

Bsp $(,011)_2 = (?)_{10}$ $r_{alt} = 2$ $r_{neu} = 10$ Mult. mit 1010

```
0,011 · 1010        0,110 · 1010        0,1 · 1010
      ,000                 ,000               ,0
     0,11                 1,10               1,
    00,10                00,0               00,
   011,                 110,               100,
   11,110              111,100             101,0
```

10^{-1} Übertrag 3_{dez} 10^{-2} 7_{dez} 10^{-3} 5_{dez}

$0,375_{10}$

- II Konvertierung von Zahlen mit ganzzahlig- und gebrochen Anteil

getrennte Umwandlung von Vor- und Nachkommastellen und anschließend Addition (Zusammenfügen)

Bsp $(38,375)_{10} = (?)_2$

$(38)_{10} \Rightarrow (100110)_2$ $(0,375)_{10} = (0,011)_2$

$(38,375)_{10} = (100110,011)_2$

II. Konvertierung in Dezimalsystem

$(100110,011)_2$

$= 2^5 + 2^2 + 2^1 + 2^{-2} + 2^{-3} = 38,375_{10}$

#

II Oktalsystem

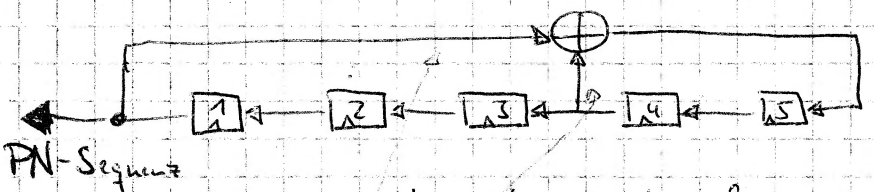
Basis: = 8   Ziffernwert $o_j \in (0,1,\dots,7)$

nichtredundantes Stellenwertsystem

Wert $Z = \sum_{j=-m}^{n-1} o_j \cdot 8^j$

Anwendung: Pseudo Binär-Zufallsfolgen-Generator

⊕EXOR



(PN = Pseudo Noise) → Anwendung: Testsignal

Kennzeichnung der Rückführungen:

	Reg	Reg	Reg	Reg	Reg
	5	4	3	2	1
		↓			↓
0	0	1	0	0	1

$\underbrace{0\ 0\ 1}\ \underbrace{0\ 0\ 1}$

11 okt

Aufbau: 3 Dualziffern werden zu einer Oktalzahl zusammengeführt

Dualsystem: $2^5 \; 2^4 \; 2^3 \; 2^2 \; 2^1 \; 2^0 \; , \; 2^{-1} \; 2^{-2} \; 2^{-3}$

$\underbrace{d_5 \; d_4 \; d_3} \; \underbrace{d_2 \; d_1 \; d_0} \; , \; d_{-1} \; d_{-2} \; d_{-3}$

Oktalsystem: $o_1 \quad o_0 \quad , \quad o_{-1}$

$8^1 \quad 8^0 \quad , \quad 8^{-1}$

d_{3j+2}	d_{3j+1}	d_{3j}	o_j
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Bsp: $(1110,01)_2 = (?)_8$

$001 \; 110 \; , \; 010$

$(16,2)_8$

Bsp $(16,2)_8 \to (?)_{10}$

$\overset{\uparrow}{8^1} \; \overset{\uparrow}{8^0} , \overset{\uparrow}{8^{-1}} \Rightarrow 14,25_{10}$

B=Rop
19.10.✓

Bsp.: $(14,25)_{10} = (?)_8$ $\quad r_{alt} = 10 \quad r_{neu} = 8$

$14 : 8 \overset{Rest}{\Rightarrow} 6 \quad \hat{=} \; 8^0$

$= 1 : 8 \; \text{Rest} \; 1 \quad \hat{=} \; 8^1$

$= 0$

$0,25 \cdot 8 = 2,0$

$\overset{\uparrow}{8^{-1}} \quad \hat{=}$ Abbruch

16,2

Hexadezimalsystem

Basis $r = 16$

Ziffernwerte: $h_j \in (0, 1, \ldots, 9, \overset{a}{A}, \overset{b}{B}, \overset{c}{C}, \overset{d}{D}, \overset{e}{E}, \overset{f}{F})$

$h'_j \in (0, 1, \ldots, 9, 10 \ldots \ldots 15$

nichtredundantes Stellenwertsystem

Wert: $Z = \sum_{j=-m}^{n-1} h'_j \cdot 16^j$

$d_i \in (0, 1)$

Dualsystem $2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0,\ 2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}$

$$\underbrace{d_7\ d_6\ d_5\ d_4}_{h'_1}\ \underbrace{d_3\ d_2\ d_1\ d_0}_{h'_0},\ \underbrace{d_{-1}\ d_{-2}\ d_{-3}\ d_{-4}}_{h'_{-1}}$$

$\Updownarrow$

Hexadezimalsystem $16^1\ \ 16^0,\ 16^{-1}$

d_{4j+3}	d_{4j+2}	d_{4j+1}	d_{4j+0}	h'_j	h_j
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	10	A
1	0	1	1	11	B
1	1	0	0	12	C
1	1	0	1	13	D
1	1	1	0	14	E
1	1	1	1	15	F

Kennzeichnung:

$(\ldots)_{16}$ üblicher ..H ..h ..hex

0x..

Bsp: positives Dualzahlen-format

$(11110010101101,11101)_2$

|00 | | | || | 000|

ges: hexadez. Darstellung

$(3\ C\ A\ D,\ E\ 8)_{16}$

ges dezim. Darstellung

$3 \cdot 16^3 + 12 \cdot 16^2 + 10 \cdot 16^1 + 13 \cdot 16^0,\ 14 \cdot 16^{-1} + 8 \cdot 16^{-2}$

$(15533,90625)_{10}$

Bsp: $(15533,90625)_{10} = (?)_{16}$

$15533 : 16$

	Rest	
= 970	13	← 16^0
60	10	
3	12	
0	3	← 16^3

$0,90625 \cdot 16$

$= 14,5$

16^{-1} →

$0,5 \cdot 16 =$

$= 8$

16^{-2} →

3 12 10 13 , 14 8

$(3\ C\ A\ D,\ E\ 8)_H$

Wertebereich und Quantisierung im Dualsystem mit positiven Werten

Format:

$$\underbrace{\boxed{2^{n-1}}\boxed{2^{n-2}}\ldots\boxed{2^{1}}\boxed{2^{0}}}_{n \text{ Vorkommastellen}},\underbrace{\boxed{2^{-1}}\boxed{2^{-2}}\ldots\boxed{2^{-m}}}_{m \text{ Nachkommastellen}}$$

Wertebereich: minimal $Z_{min} = 0$

maximal $Z_{max} = 2^n - 2^{-m}$

Quantisierung bzw. Quantisierungsstufenhöhe

↳ $q = 2^{-m}$

max. Quantisierungsfehler nach einer durchgeführten Rundung $= \pm \frac{2^{-m}}{2}$

Stufe 2^{-m}

max Quantisierungsfehler $\frac{\text{Stufe}}{2} \Rightarrow \pm \frac{2^{-m}}{2} = \pm \frac{1}{2}$ LSB

Bsp.: geg. pos. Dualzahlformat $d \in (0,1)$

$|d_5\ d_4\ d_3\ d_2\ d_1\ d_0\ ,\ d_{-1}\ d_{-2}\ d_{-3}|$

ges. max darstellbarer Wert Z_{max} = Dezimalangabe

Quantisierungsstufenhöhe q in Dezimalangabe

Lsg.) $Z_{max} = 63,875$

$q = 0,125$

II Bestimmung der Anzahl der zur Wertdarstellung erforderlichen Nachkommastellen im Dualsystem

→ Genauigkeit soll durch die Umwandlung nicht verringert werden.

→ wird vom gebrochenen Anteil bestimmt

Dualsystem

$$,\underbrace{\boxed{2^{-1}}\ \ldots\ \boxed{2^{-m+1}}\boxed{2^{-m}}}_{m \text{ Nachkommastellen}}$$

$$\text{Quantisierungsstufenhöhe} = q = 2^{-m}$$

Dezimalsystem

$$,\underbrace{\boxed{10^{-1}}\ \ldots\ \boxed{10^{-z+1}}\boxed{10^{-z}}}_{z \text{ Nachkommastellen}}$$

$$\text{Genauigkeit} = 1 \cdot 10^{-z} = \frac{1}{10^{z}}$$

(= Quantisierungsstufenhöhe im Dezimalsystem)

Bedingung: zum Erhalt der Genauigkeit bei Umwandlung Dezimalsystem nach Dualsystem:

$$\underset{\text{im Dualsystem}}{\text{Quantisierungsstufenhöhe}} \leq \underset{\text{im Dezimalsystem}}{\text{Genauigkeit}}$$

↷

$$2^{-m} \leq 10^{-z}$$

$$-m \leq \log_2 10^{-z}$$

$$m \geq -\log_2(10^{-z})$$

$$m \geq -\frac{\log_{10}(10^{-z})}{\log_{10}(2)} = -\frac{-z}{\log_{10} 2} = \frac{z}{\log_{10} 2} \approx 3{,}322 \cdot z$$

Aufrunden auf nächst größere ganze Zahl im Falle das
es noch keine ... [illegible] (ceiling Fkt)

$$m = \text{ceil}\left[\frac{z}{\log_{10} 2}\right] \approx \text{ceil}[3,322 \cdot z]$$

Tabelle

Genauigkeit im Dezimalsystem 10^{-z}	Anzahl Nachkommast. Dualsys. m	Quantisierungsstufenhöhe im Dualsys. $2^{-m} = q$
$10^0 = 1$ (ohne Nkstellen)	0	$2^0 = 1$
$10^{-1} = 0,1$	4	$2^{-4} = 0,0625$
$10^{-2} = 0,01$	7	$2^{-7} = 0,0078125$
$10^{-3} = 0,001$	10	$2^{-10} = 0,0009765625$
⋮	⋮	⋮

1) Bestimmung d. Anzahl notwendiger Vorkommastellen

$$\underbrace{\boxed{2^{n-1}}\boxed{2^{n-2}}\dots\boxed{2^1}\boxed{2^0}}_{n \text{ Vorkommastellen}},$$

max. darstellbarer Wert $= 2^n - 1$

Bedingung für Darstellbarkeit

$$2^n - 1 \geq \text{Wert} \leftarrow \text{Wert ohne Nachkommastelle}$$

$$2^n \geq \text{Wert} + 1$$

$$n \geq \log_2(\text{Wert}+1) = \frac{\log_{10}(\text{Wert}+1)}{\log_{10} 2} \approx 3,322 \cdot \log_{10}(\text{Wert}+1)$$

Anzahl Vorkommastellen

$$n = \text{ceil}\left[\frac{\log_{10}(\text{Wert}+1)}{\log_{10}(2)}\right] \approx \text{ceil}[3,322 \cdot \log_{10}(\text{Wert}+1))$$

BinRep
19.10.(Ü)
2. Teil
✓

Bsp: $19876543{,}21)_{10}$

ges a) Erforderliche Anzahl Vorkommazahlen

b) –"– –"– Nachkommazahlen

für Wandlung in Dualsystem

a) $n \geq \log_2 (\text{Wert} + 1) = 23{,}2356$

$n = 24$

b) $m \geq -\log_2 (10^{-2}) = 6{,}644$

$m = 7$

Addition

(wichtigste Rechenoperation)

→ nicht benötigt bei Multiplikation u. Div. mit Zweierpotenzen

Additionsregeln einer einzelnen Stelle

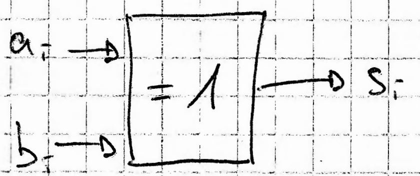
$a_i ; b_i$: Summandenstellen

s_i : Summenstelle

c_{i+1} : Ausgangsübertragstelle

Stellenwert 2^i	2^i	2^i	2^{i+1}
a_i	b_i	s_i	c_{i+1}
Eingänge		Ausgänge	
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

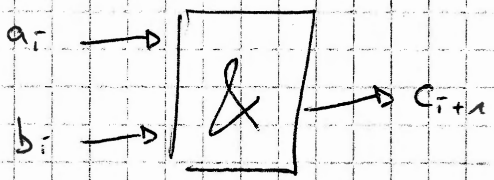
1 logische Realisierung Summenstelle



Exklusiv ODER

$$s_i = a_i \oplus b_i$$

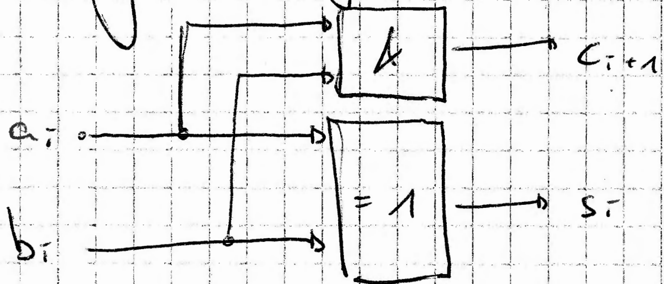
1 logische Realisierung Ausgangsstelle



$$c_{i+1} = a_i \cdot b_i$$

$$c_{i+1} = a_i \wedge b_i$$

Gesamtlogik



Halbaddierer

- Additionsregeln mit Eingangsübertrag:

Volladdierer

Stellenwert	Stellenwert	Stellenwert	Stellenwert	Stellenwert
2^i	2^i	2^i	2^i	2^{i+1}
c_i	a_i	b_i	s_i	c_{i+1}
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

Si-20D

15 40

✓

$$\boxed{s_i = (a_i \oplus b_i) \oplus c_i}$$

$$a_i \oplus b_i = a_i \cdot \bar{b}_i + \bar{a}_i \cdot b_i$$

$$s_i = (a_i \cdot \bar{b}_i + \bar{a}_i \cdot b_i) \oplus c_i$$

$$= (a_i \cdot \bar{b}_i + \bar{a}_i \cdot b_i) \cdot \bar{c}_i + \overline{(a_i \cdot \bar{b}_i + \bar{a}_i \cdot b_i)} \cdot c_i$$

$$\ldots + ((\bar{a}_i + b_i) \cdot (a_i + \bar{b}_i) \cdot \ldots$$

$$s_i = a_i \cdot \bar{b}_i \cdot \bar{c}_i + \bar{a}_i b_i \cdot \bar{c}_i + [\underbrace{\bar{a}_i a_i}_{0} + \bar{a}_i \bar{b}_i + b_i a_i + \underbrace{b_i \bar{b}_i}_{0}] \cdot c_i$$

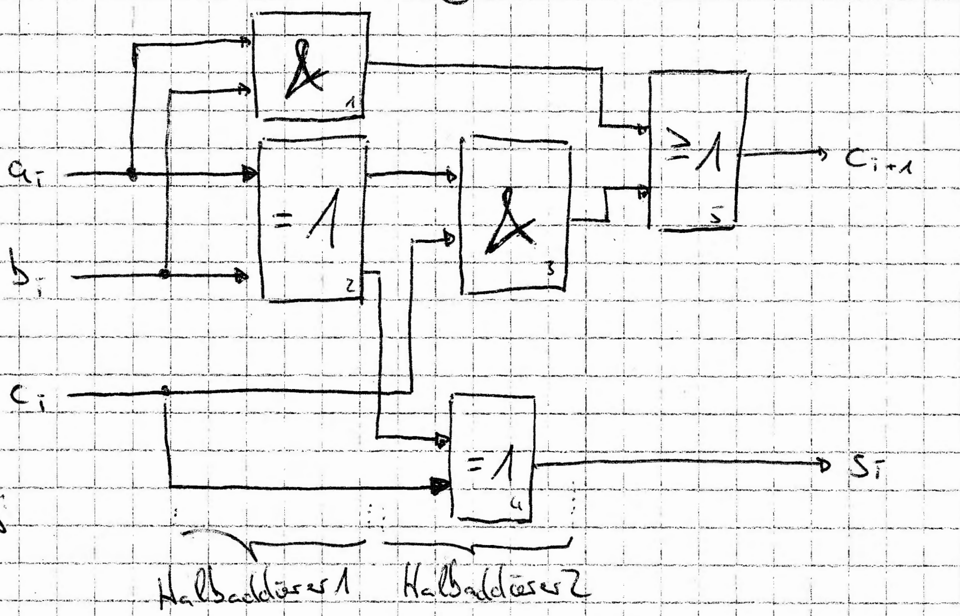
$$s_i = a_i \cdot \bar{b}_i \cdot \bar{c}_i + \bar{a}_i b_i \cdot \bar{c}_i + \bar{a}_i \bar{b}_i \cdot c_i + a_i b_i c_i$$

$$c_i + 1 = \underbrace{a_i b_i \bar{c}_i}_{a_i b_i \text{ ausklam}} + \underbrace{\bar{a}_i b_i c_i + a_i \bar{b}_i c_i}_{c_i \text{ ausklammern}} + \underbrace{a_i b_i c_i}_{a_i b_i \text{ auskl.}}$$

$$c_{i+1} = (\underbrace{\bar{a}_i b_i + a_i \bar{b}_i}_{a_i \oplus b_i}) \cdot c_i + a_i b_i \cdot (\underbrace{c_i + \bar{c}_i}_{1})$$

$$\boxed{c_i + 1 = (a_i \oplus b_i)\, c_i + a_i b_i}$$

Schaltung Volladdierer



Summenbildung:
Blöcke 2 u. 4

Übertragsbildung:
Blöcke 1, 2, 3 u. 5

Einführung Einheitsverzögerung Δ

jeder Block verzögert um ein Δ

Verzögerung Summenstelle s_i

a_i oder $b_i \to s_i \rightsquigarrow 2\Delta$

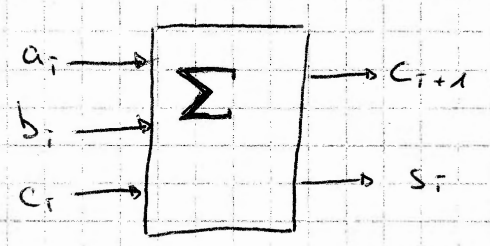
$c_i \to s_i \rightsquigarrow 1\Delta$

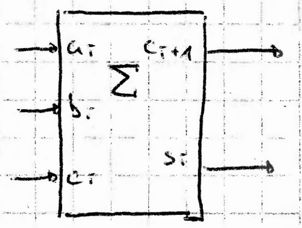
Verzögerung Übertragsstelle c_{i+1}

a_i oder $b_i \to c_{i+1} \rightsquigarrow 3\Delta$

$c_i \to c_{i+1} \rightsquigarrow 2\Delta$

Symbol einer Volladdiererstelle

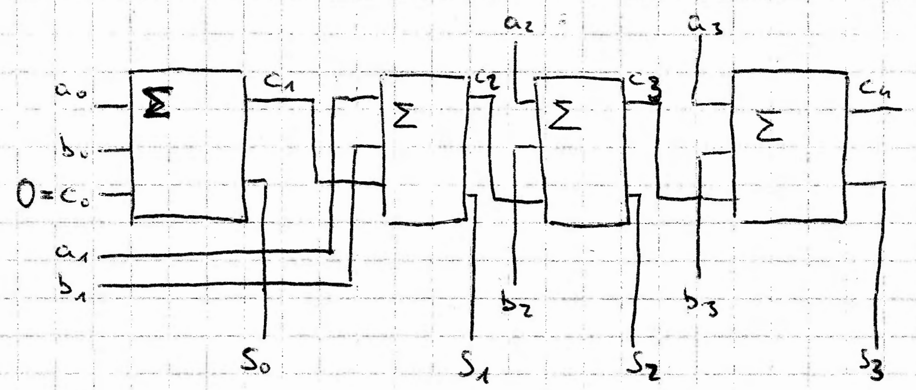




Bsp. Verschaltung von 4 Stellen

Serielle Übertragsgenerierung Δ

(Ripple Carry Adder)



längster Weg a_0 (oder b_0) $\to c_1 \to c_2 \to c_3 \to s_3 \Rightarrow 8\Delta$

$3\Delta \quad 2\Delta \quad 2\Delta \quad 1\Delta$

Bitbreite n (Stellenzahl n) $\Rightarrow \underbrace{3\Delta + 1\Delta}_{\text{1. u. letzte}} + (n-2)\cdot 2\Delta =$

= allg. für Bitbreite n: Verzögerung $n \cdot 2\Delta$

Addierer mit paralleler Übertragslogik

Carry-Look-Ahead-Adder   CLA-Adder

Prinzip: an jeder Stelle wird das Signal „Generate" g_{i+1} und „Propagate" p_{i+1} generiert.

Bedeutung:

(erzeugen) „Generate" g_{i+1}: kennzeichnet einen durch die beiden Summanden der Stelle bei Kombination $a_i=1$ und $b_i=1$ erzeugten Übertrag.

(weiterleiten) „Propagate" p_{i+1}: kennzeichnet einen von der niederwertigen Stelle $i-1$ weiterzuleitenden Übertrag an Stelle $i+1$, im Falle dass der Übertrag nicht an der aktuellen Stelle kompensiert werden kann a_i oder $b_i=1$

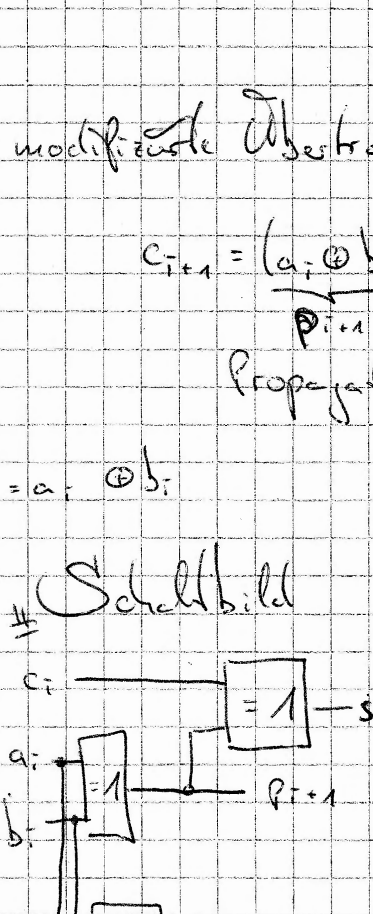
„modifizierte Übertragsgleichung des Volladdierers

$$c_{i+1} = \underbrace{(a_i \oplus b_i)\cdot c_i}_{p_{i+1}} + \underbrace{a_i \cdot b_i}_{g_{i+1}}$$

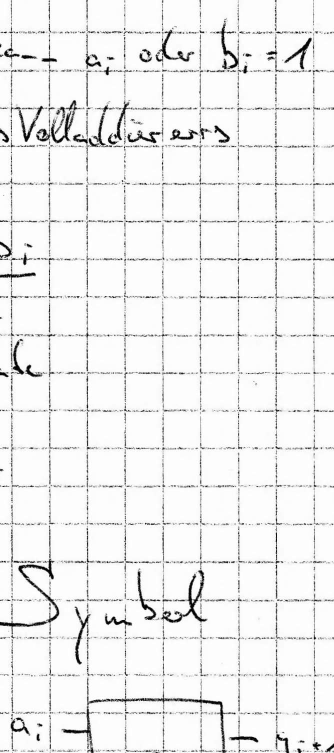
Propagate   Generate

$p_{i+1} = a_i \oplus b_i$ $g_{i+1} = a_i \cdot b_i$

Schaltbild



Symbol



B_ROP

1.11.

- Ripple-Carry-Adder (mit serieller Übertr.-kette)

Verzögerung von $2n\,\Delta$

n = Bitbreite

- Carry-Look-Ahead-Adder (mit paralleler Übertrags-kette)

generate / propagate

parallele Übertragslogik für 4 Stellen

wichtig: Als Eingangsgrößen dürfen nur p_{i+1}, g_{i+1} und Übergangsübertrag der niederwert. Stelle (z.B. c_0) auftreten.

Eingangsübertrag des Addierers

LSB
Stelle: 2^0 c_0 beliebig

$c_1 = g_1 + c_0 \cdot p_1$ (3 Eingänge zur Generierung von c_1)

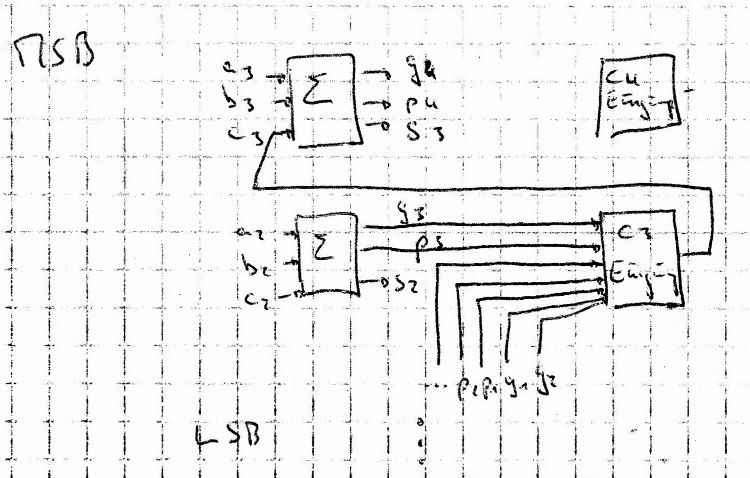
2^1 $c_2 = g_2 + c_1 p_2$ (5 Eingänge)

$c_2 = g_2 + (g_1 + c_0 p_1) p_2 = g_2 + g_1 p_2 + c_0 p_1 p_2$

2^2 $c_3 = g_3 + c_2 p_3$ (7 Eingänge)

$c_3 = g_3 + (g_2 + g_1 p_2 + c_0 p_1 p_2) p_3 = g_3 + g_2 p_3 + g_1 p_2 p_3 + c_0 p_1 p_2 p_3$

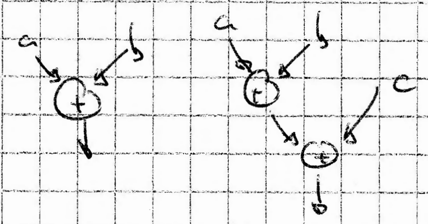
2^3 $c_4 = g_4 + c_3 p_4$ (9 Eingänge)



Längster Pfad:

$1\Delta + 2\Delta + 1\Delta = 4\Delta$

4 Kombinatorische Multioperandenaddition



Anwendung: Addition Teilprodukte bei Multiplikation

Bsp: $\overset{3}{0011} \cdot \overset{7}{0111}$

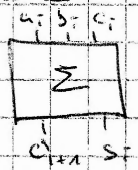
0011
+ 0011
+ 0011
10101

21

Bsp: Addition von 4 Operanden mit 4 Dualstellen

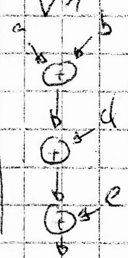
Variante: Kettenschaltung mit Ripple Carry Addierer

grundelement

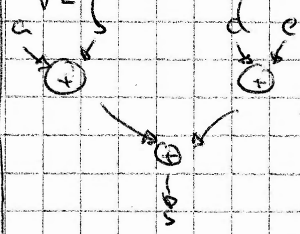


a_3 a_2 a_1 a_0
b_3 b_2 b_1 b_0
d_3 d_2 d_1 d_0
e_3 e_2 e_1 e_0

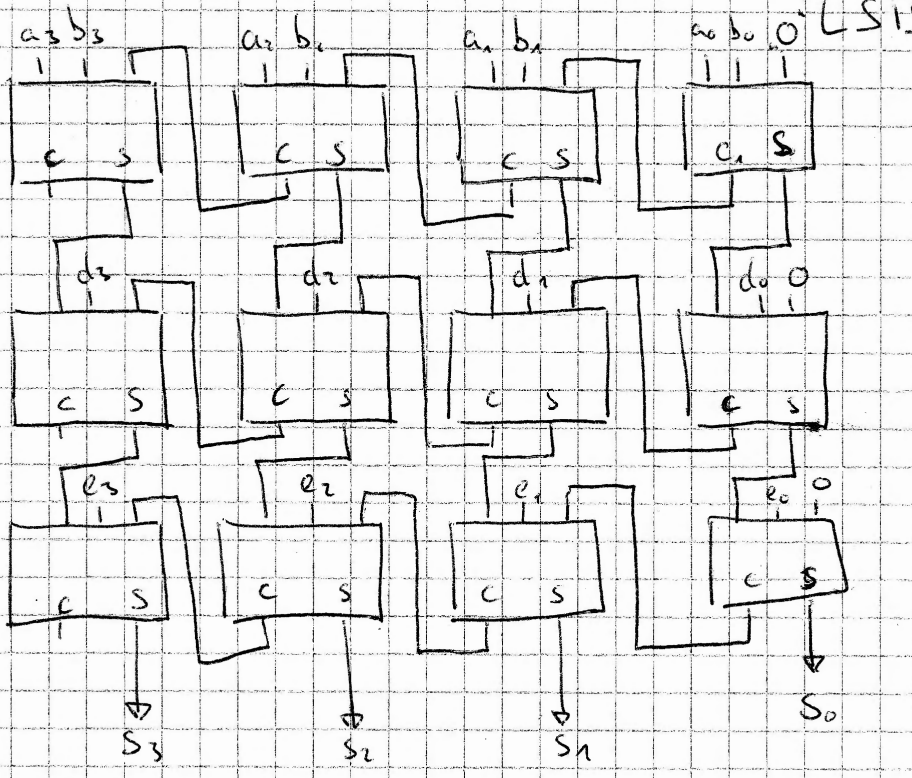
V1



V2



V1 MSB



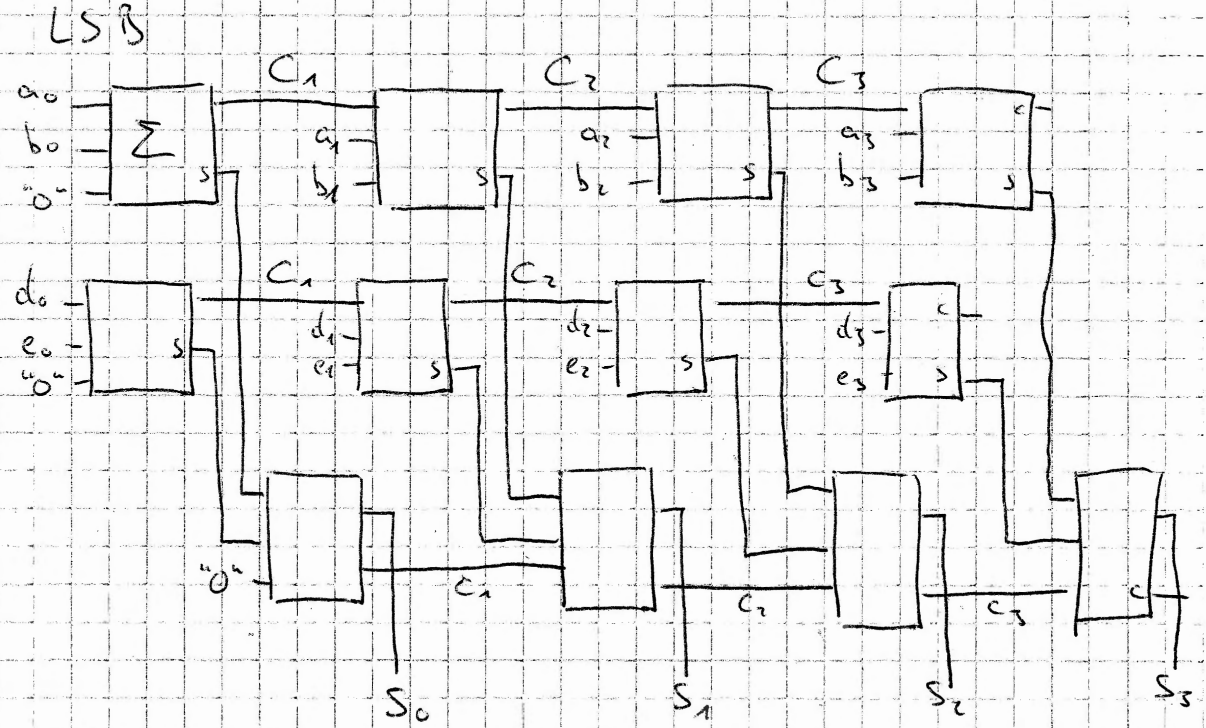
längster Weg

$a_i ; b_i \to s_i$ 2Δ
$a_i ; b_i \to c_{i+1}$ 3
$c_i \to c_{i+1}$ 2Δ
$c_i \to s_i$ 1Δ

2Δ + 2Δ + 3
+ 2Δ + 2Δ + 1
= 12Δ

2.11.
B._ROR

V2



Verzögerung Volladdierer

$a_i; b_i \Rightarrow s_i \Rightarrow 2\Delta$

$a_i; b_i \Rightarrow c_{i+1}$ 3Δ

$c_i \Rightarrow s_i$ 1Δ

$c_i \Rightarrow c_{i+1}$ 2Δ

V3 Carry-Save-Addition

Zahlendarstellung der Zwischenergebnisse:

„Stored-carry"-Darstellung bzw SC-Darstellung

Kennzeichen SC-Darstellung:

Basis $r = 2$

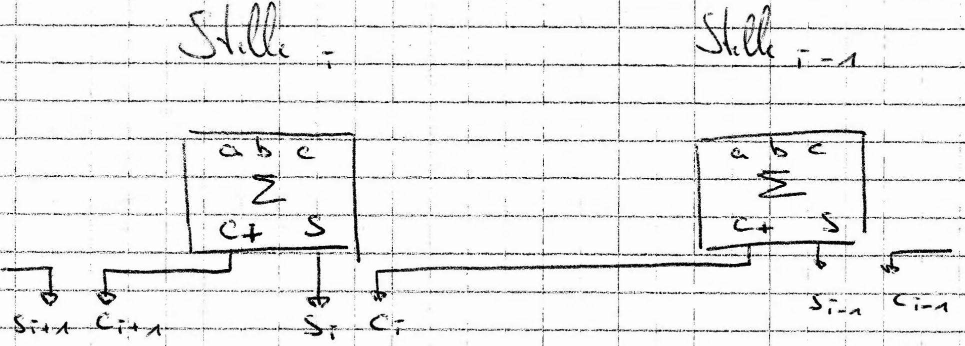
Ziffernwerte $s_i^{sc} \in \{0, 1, 2\}$

Ziffernanzahl > Basis ⇒ redundante Zahlendarstellung

binäre Codierung d. Ziffernwerte

s_i^{sc}	s_i^s	$s_i^c = c_i$
0	0	0
1	0	1
	1	0
2	1	1

z.B.: zur Codierung der Stelle



Rückwandlung eines SC-Wertes in das Dualzahlformat

→ normale Dualzahladdition von s_i^s und $s_i^c = c_i$

für $i = -m \dots n-1$

$$
\begin{array}{rcccc}
 & s_3^s & s_2^s & s_1^s & s_0^s \\
+ & c_3 = s_3^c & c_2 = s_2^c & c_1 = s_1^c & c_0 = s_0^c \\
\hline
 & s_3 & s_2 & s_1 & s_0
\end{array}
$$

Vorteil der SC-Darstellung

→ Übertragskette der Addierer entfällt

außer bei Rückwandlungsaddition (noch vermeidbar durch CLA-Addierer

Bsp redundante Darstellung

$[0101]_{sc} = [0021]_{sc} = [1221]_{sc} = [2101]_{sc} = [2021]_{sc}$

$= 5_{10}$

		2^3	2^2	2^1	2^0	
$[0101]_{sc}$	s^s	0	1	0	1	↕ vertauschbar
	+ s^c	0	0	0	0	
	=	0	1	0	1	

$[0021]_{sc}$	s^s	0	0	1	1
	+ s^c	0	0	1	0
	=	0	1	0	1

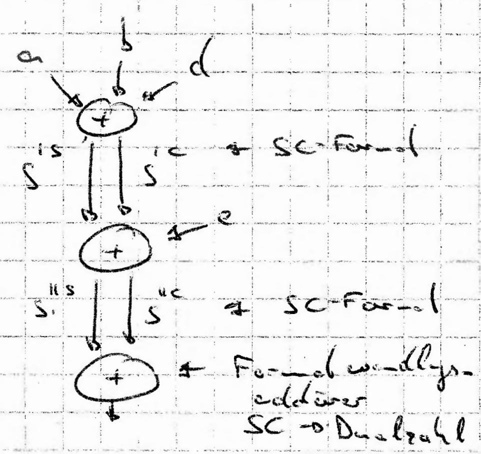
$[1221]_{sc}$	s^s	1	1	1	1
	+ s^c	0	1	1	0
	=	0	1	0	1

Übertrag wird ignoriert

Anwendung: 4 Operanden Addition mit Carry-Save

Besonderheit in der obersten Stufe oder mehreren oberen Stufen bei Verwendung der Baumstruktur kann d. Übertragseingang ~~mit~~ mit als Operandeneingang genutzt werden.

4 Operanden: a b d e



Aufgabe: Add von 4 Operanden

$a_4 \ldots a_0$
$b_4 \ldots b_0$
$d_4 \ldots d_4$
$e_4 \ldots e_4$

Grundelement Addierer