Darstellung ganzer Zahlen als Bitvektoren

Jan Peleska Jan Bredereke

Universität Bremen, Fachbereich Informatik

Vers. 1.1

1 Darstellung natürlicher Zahlen $z \in \mathbb{N}_0$ und Addition

1.1 Dualzahlen dargestellt durch Bitvektoren

Ein Maschinenwort ist ein Bitvektor. Um mit einem Rechner Zahlen verarbeiten können, können wir eine Zahl als Dualzahl und diese durch einen Bitvektor darstellen:

$$(x_{n-1} \ \dots \ x_0) \quad \text{interpretiert als} \quad z = \sum_{i=0}^{n-1} 2^i \cdot x_i \qquad \text{mit } x_i \in \{0,1\} \tag{1}$$

Beispiel:

$$(1\ 0\ 1\ 0\ 1) \quad \text{interpretiert als} \quad 2^4 + 2^2 + 2^0 = 16 + 4 + 1 = 21$$

□

So dargestellte Dualzahlen $(x_{n-1} \ \dots \ x_0)$ sind Elemente aus $\underbrace{\mathbb{B} \times \ldots \times \mathbb{B}}_{n \text{ mal}} = \mathbb{B}^n$. Dualzahlen kennen nur zwei Ziffern:

Definition 1 (Menge der Binärziffern $\mathbb{B}$)

$\mathbb{B} = \{0, 1\}$

□

Formal beschreiben wir die Darstellung einer natürlichen Zahl $z \in \mathbb{N}_0$ als Bitvektor durch eine Repräsentationsfunktion r:

Definition 2 (Repräsentationsfunktion r)

$$\mathbb{N}_0 \xrightarrow{r} \mathbb{B}^n, \qquad z \mapsto (x_{n-1} \ \dots \ x_0) \qquad \textit{mit} \ \sum_{i=0}^{n-1} 2^i x_i = z$$

mit Definitionsbereich

$\underline{dom}\, r = \{z \in \mathbb{N}_0 \mid z < 2^n\}$

Das heißt, daß der Bitvektor $(x_{n-1} \ \dots \ x_0)$ einfach die Folge der Dualziffern von z ist. □

Den Beweis, daß die Repräsentation als Bitvektor wohldefiniert und eindeutig ist, lassen wir hier aus. Er beruht darauf, daß sich Zahlen in verschiedenen Zahlensystemen ineinander umwandeln lassen, und zwar eindeutig.

In der Programmiersprache C wird die Repräsentationsfunktion $r(z) = x$ implementiert durch die Funktion `scanf("%u",&x)` (wobei die Variable `x` definiert ist als `unsigned int x;`).

1.2 Abstrakte Interpretation

Die obige umgekehrte Interpretation eines Bitvektors als natürliche Zahl $z \in \mathbb{N}_0$ beschreiben wir formal durch eine Abstraktionsfunktion ρ:

Definition 3 (Abstraktionsfunktion ρ)

$$\mathbb{B}^n \xrightarrow{\rho} \mathbb{N}_0, \qquad (x_{n-1} \ \ldots \ x_0) \mapsto \rho(x_{n-1} \ \ldots \ x_0) =_{df} \sum_{i=0}^{n-1} 2^i x_i$$

□

In der Programmiersprache C wird die Abstraktionsfunktion $\rho(x)$ implementiert durch die Funktion `printf("%u",x)` (wobei die Variable `x` wiederum definiert ist als `unsigned int x;`).

1.3 Rechenoperationen auf Bits und Bitvektoren

Auf $\mathbb{B}$ definieren wir folgende Operationen:

Definition 4 (Multiplikation $\cdot$ auf $\mathbb{B}$)

$$\cdot\ : \mathbb{B} \times \mathbb{B} \to \mathbb{B}, \qquad \begin{aligned} (0,0) &\mapsto 0 \\ (0,1) &\mapsto 0 \\ (1,0) &\mapsto 0 \\ (1,1) &\mapsto 1 \end{aligned}$$

Diese Abbildung läßt sich auch durch eine Funktionstafel *darstellen:*

x	y	$x \cdot y$
0	0	0
0	1	0
1	0	0
1	1	1

□

Definition 5 (Addition $\oplus$ modulo 2 auf $\mathbb{B}$)

$\oplus : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$,

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

□

Definition 6 (Addition $+'$ auf $\mathbb{B}^n$)

$$
\begin{aligned}
&+' : \mathbb{B}^n \times \mathbb{B}^n \longrightarrow \mathbb{B}^{n+1},\\
&(x_{n-1} \ \ldots \ x_0) +' (y_{n-1} \ \ldots \ y_0) =_{df}\\
&\qquad (u_{n-1} \quad (x_{n-1} \oplus y_{n-1} \oplus u_{n-2}) \quad \ldots \quad (x_1 \oplus y_1 \oplus u_0) \quad (x_0 \oplus y_0))
\end{aligned}
$$

mit

$$
\begin{aligned}
u_0 &= x_0 \cdot y_0\\
u_l &= x_l \cdot y_l + (x_l \oplus y_l) \cdot u_{l-1} \qquad \text{für } 1 \leq l \leq n
\end{aligned}
$$

Dabei ist u_l der „von Stelle l erhaltene Übertrag“. □

Das hier verwendete Schema wurde früher praktisch in Hardware realisiert, und zwar als sogenannter Serienaddierer.

1.4 Eigenschaften

Satz 1 *Das folgende Diagramm* kommutiert*:*

$$
\begin{array}{ccc}
\mathbb{N}_0 \times \mathbb{N}_0 & \xrightarrow{+} & \mathbb{N}_0\\
\Big\uparrow{\scriptstyle \rho\times\rho} & & \Big\uparrow{\scriptstyle \rho}\\
\mathbb{B}^n \times \mathbb{B}^n & \xrightarrow{+'} & \mathbb{B}^{n+1}
\end{array}
$$

d.h.

$$
\begin{aligned}
&\forall n \in \mathbb{N}\ .\forall (x_{n-1} \ \ldots \ x_0), (y_{n-1} \ \ldots \ y_0) \in \mathbb{B}^n\ .\\
&\qquad \rho(x_{n-1} \ \ldots \ x_0) + \rho(y_{n-1} \ \ldots \ y_0) = \rho_{n+1}\big((x_{n-1} \ \ldots \ x_0) +' (y_{n-1} \ \ldots \ y_0)\big)
\end{aligned}
$$

Dabei ist ρ_{n+1} die aus Definition 3 bekannte Abstraktionsfunktion $\rho_{n+1} : \mathbb{B}^{n+1} \to \mathbb{N}_0$, $(x_n \ \ldots \ x_0) \mapsto \sum_{i=0}^{n} 2^i x_i$ für vorzeichenlose Zahlen, allerdings auf $\mathbb{B}^{n+1}$ statt auf $\mathbb{B}^n$. □

Beweis:

Zu zeigen ist: Für alle $n \in \mathbb{N}, (x_{n-1} \ \ldots \ x_0), (y_{n-1} \ \ldots \ y_0) \in \mathbb{B}^n$ gilt

$$
\underbrace{\sum_{i=0}^{n-1} 2^i (x_i + y_i)}_{=_{\mathrm{df}}\ L} \ = \ \underbrace{2^n u_{n-1} \ + \ \sum_{i=1}^{n-1} 2^i (x_i \oplus y_i \oplus u_{i-1}) \ + \ (x_0 \oplus y_0)}_{=_{\mathrm{df}}\ R} \tag{2}
$$

Beweis durch Induktion über $n \geq 2$.

Sonderfall $\boxed{n = 1}$, $(x_0), (y_0) \in \mathbb{B}^1$:

Zu zeigen ist:

$$\sum_{i=0}^{0} 2^i (x_i + y_i) \;=\; 2^1 u_0 \;+\; (x_0 \oplus y_0) \tag{3}$$

Dies ist gleichbedeutend mit

$$x_0 + y_0 \;=\; 2 \cdot x_0 \cdot y_0 \;+\; (x_0 \oplus y_0) \tag{4}$$

Der Beweis erfolgt durch Ausrechnen in einer Funktionstafel:

x_0	y_0	$x_0 + y_0$	$2x_0y_0 \;+\; (x_0 \oplus y_0)$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	2	2

Fall $\boxed{n = 2}$, Induktionsverankerung:

$$L \;=\; \sum_{i=0}^{1} 2^i (x_i + y_i) = 2(x_1 + y_1) + (x_0 + y_0) \tag{5}$$

$$R \;=\; 2^2 u_1 \;+\; \sum_{i=1}^{1} 2^i (x_i \oplus y_i \oplus u_{i-1}) \;+\; (x_0 \oplus y_0) \tag{6}$$

$$\;=\; 2^2 (x_1 y_1 + (x_1 \oplus y_1) u_0) \;+\; 2(x_1 \oplus y_1 \oplus u_0) \;+\; (x_0 \oplus y_0) \tag{7}$$

$$\;=\; 4x_1y_1 \;+\; 4(x_1 \oplus y_1) x_0 y_0 \;+\; 2(x_1 \oplus y_1 \oplus (x_0y_0)) \;+\; (x_0 \oplus y_0) \tag{8}$$

Der Beweis erfolgt wiederum über eine Funktionstafel:

x_1	y_1	$L = 2(x_1 + y_1) + (x_0 + y_0)$	$R = 4x_1y_1 + 4(x_1 \oplus y_1)x_0y_0$ $+ 2(x_1 \oplus y_1 \oplus (x_0y_0)) + (x_0 \oplus y_0)$	
0	0	$x_0 + y_0$	$2x_0y_0 + (x_0 \oplus y_0)$	$(*)$
0	1	(wie Fall 1/0 aus Symmetriegründen)		
1	0	$2 + x_0 + y_0$	$4x_0y_0 + 2(1 \oplus (x_0y_0)) + (x_0 \oplus y_0)$	$(\dagger)$
1	1	$4 + x_0 + y_0$	$4 + 2x_0y_0 + (x_0 + y_0)$	$(\ddagger)$

Die Gleichungen $(*)$ und $(\ddagger)$ sind wegen der oben bewiesenen Gleichung (4) erfüllt. Die Gleichung $(\dagger)$ weisen wir über eine weitere Funktionstafel nach:

x_0	y_0	$2 + (x_0 + y_0)$	$4x_0y_0 + 2(1 \oplus (x_0y_0)) + (x_0 \oplus y_0)$
0	0	2	2
0	1	3	$2 + 1$
1	0	3	$2 + 1$
1	1	4	4

Induktionsannahme:

Es gelte die Behauptung für $n = k, k \geq 2$:

$$\sum_{i=0}^{k-1} 2^i(x_i + y_i) \;=\; 2^k u_{k-1} \;+\; \sum_{i=1}^{k-1} 2^i(x_i \oplus y_i \oplus u_{i-1}) \;+\; (x_0 \oplus y_0) \tag{9}$$

Induktionsschritt: Fall $\boxed{n = k+1}$

Zu zeigen ist:

$$\underbrace{\sum_{i=0}^{k} 2^i(x_i + y_i)}_{L} \;=\; \underbrace{2^{k+1} u_k \;+\; \sum_{i=1}^{k} 2^i(x_i \oplus y_i \oplus u_{i-1}) \;+\; (x_0 \oplus y_0)}_{R} \tag{10}$$

Dies ist gleichbedeutend mit

$$L = 2^k(x_k + y_k) + \sum_{i=0}^{k-1} 2^i(x_i + y_i) \tag{11}$$

und

$$R = 2^{k+1}u_k \;+\; 2^k(x_k \oplus y_k \oplus u_{k-1}) \;+\; \sum_{i=1}^{k-1} 2^i(x_i \oplus y_i \oplus u_{i-1}) \;+\; (x_0 \oplus y_0) \tag{12}$$

$$= 2^{k+1}u_k \;+\; 2^k(x_k \oplus y_k) - 2^{k+1}(x_k \oplus y_k)u_{k-1} + 2^k u_{k-1} + \sum_{i=1}^{k-1} 2^i(x_i \oplus y_i \oplus u_{i-1}) \;+\; (x_0 \oplus y_0) \tag{13}$$

Der letzte Schritt (13) wird über die folgende Hilfsformel bewiesen:

$$2^k(x_k \oplus y_k) - 2^{k+1}(x_k \oplus y_k)u_{k-1} + 2^k u_{k-1} = 2^k(x_k \oplus y_k \oplus u_{k-1}) \tag{14}$$

Beweis der Hilfsformel (14):
Falls $u_{k-1} = 0$:

$$2^k(x_k \oplus y_k) - 0 + 0 = 2^k(x_k \oplus y_k \oplus 0) \tag{15}$$

Falls $u_{k-1} = 1$:

x_k	y_k	$2^k(x_k \oplus y_k) - 2^{k+1}(x_k \oplus y_k) + 2^k$	$2^k(x_k \oplus y_k \oplus 1)$
0	0	2^k	2^k
0	1	$2^k - 2^{k+1} + 2^k = 0$	0
1	0	(wie Fall 0/1 aus Symmetriegründen)	
1	1	2^k	2^k

(16)

Damit ist die Hilfsformel (14) bewiesen.

Nach Anwendung der Induktionsannahme (9) auf (11) und (13) bleibt zu zeigen:

$$\begin{aligned}
2^k(x_k + y_k) &\overset{!}{=} 2^{k+1}u_k \;+\; 2^k(x_k \oplus y_k) - 2^{k+1}(x_k \oplus y_k)u_{k-1} && (17)\\
\Leftrightarrow x_k + y_k &= 2u_k \;+\; (x_k \oplus y_k) - 2(x_k \oplus y_k)u_{k-1} \qquad \text{Division durch } 2^k && (18)\\
&= 2(x_k y_k + (x_k \oplus y_k)u_{k-1}) \;+\; (x_k \oplus y_k) - 2(x_k \oplus y_k)u_{k-1} && (19)\\
&= 2x_k y_k \;+\; (x_k \oplus y_k) && (20)
\end{aligned}$$

Die letzte Gleichung (20) ist gleichbedeutend mit (4) und bereits oben bewiesen worden. □

2 Darstellung negativer ganzer Zahlen und Subtraktion

2.1 Darstellung negativer ganzer Zahlen

Eine einfache, intuitive Methode zur Darstellung negativer ganzer Zahlen ist, ein Bit des Maschinenwortes bzw. Bitvektors für das Vorzeichen zu reservieren. Ist dieses Bit auf 1, so ist die Zahl negativ. Der Rest der Bits wird wie gewohnt als der Betrag $\sum_{i=0}^{n-1} 2^i x_i$ der Zahl interpretiert.

1		...	
n	n-1		0

Diese Darstellung wurde früher zum Teil tatsächlich verwendet, aber sie hat zwei wesentliche Nachteile:

1. Für die Null gibt es zwei verschiedene Repräsentationen:

0	0	...	0
n	n-1		0

   und

1	0	...	0
n	n-1		0

   Dies erfordert zusätzliche Umsicht und Maßnahmen bei der Überprüfung, ob zwei Zahlen gleich sind.

2. Der Rechner benötigt eine separate Implementierung der Subtraktion.

Eine bessere Darstellung ist die mit Hilfe des Zweierkomplements, welche wir im folgenden einführen werden. Die Darstellung der Null ist dabei eindeutig, und zusätzlich läßt sich die Subtraktion auf die Addition zurückführen, so daß dafür keine zusätzliche Hardware mehr benötigt wird.

Eine nicht-negative ganze Zahl kann eindeutig in der in Abschnitt 1.1 gezeigten Weise in einen vorzeichenlosen Bitvektor umgewandelt werden. Wir behandeln nun auch den Fall der negativen ganzen Zahlen, d.h. $z \in \mathbb{Z}$:

Definition 7 (Repräsentationsfunktion $\widetilde{r}$)

$$\widetilde{r} : \mathbb{Z} \to \mathbb{B}^{n+1}, \quad \widetilde{r}(z) = \begin{cases} (0 \;\; x_{n-1} \;\ldots\; x_0) & \mathit{mit} \; \sum\limits_{i=0}^{n-1} 2^i x_i = z \qquad \mathit{falls} \; 0 \le z < 2^n \\ (\underbrace{1}_{=w_n} \;\; w_{n-1} \;\ldots\; w_0) & \mathit{mit} \; \sum\limits_{i=0}^{n} 2^i w_i = z + 2^{n+1} \quad \mathit{falls} \; -2^n \le z < 0 \end{cases}$$

mit Definitionsbereich

$$\underline{dom}\,\widetilde{r} = \{z \in \mathbb{Z} \mid -2^n \leq z < 2^n\}$$

Außerdem sei, falls $-2^n \leq z < 0$,

$$w =_{df} \sum_{i=0}^{n} 2^i w_i = z + 2^{n+1}$$

□

Man sieht sofort, daß $2^n \leq w < 2^{n+1}$ gilt. Damit ist $w_n = 1$, d.h. das höchstwertige Bit ist für negative Zahlen immer gleich 1.

2.2 Abstrakte Interpretation

Definition 8 (Einerkomplement K_1)

$$K_1 : \mathbb{B}^{n+1} \to \mathbb{B}^{n+1}, \qquad \begin{aligned} K_1(x_n \ \dots \ x_0) &= ((x_n \oplus 1) \ \dots \ (x_0 \oplus 1)) \\ &= ((1 - x_n) \ \dots \ (1 - x_0)) \end{aligned}$$

□

Definition 9 (Projektion Π_{n+1} von $\mathbb{B}^{n+2}$ nach $\mathbb{B}^{n+1}$)

$$\Pi_{n+1} : \mathbb{B}^{n+2} \to \mathbb{B}^{n+1}, \qquad \Pi_{n+1}(x_{n+1}\, x_n \ \dots \ x_0) = (x_n \ \dots \ x_0)$$

□

Die Projektion schneidet einfach das am weitesten links stehende Bit ab.

Definition 10 (Zweierkomplement K_2)

$$K_2 : \mathbb{B}^{n+1} \to \mathbb{B}^{n+1},$$

$$K_2(x_n \ \dots \ x_0) = \begin{cases} \Pi_{n+1}\big(K_1(x_n \ \dots \ x_0) +' (0\,0 \ \dots \ 0\,1)\big) & \textit{für } (x_n \ \dots \ x_0) \neq (0 \ \dots \ 0) \\ (0\,0 \ \dots \ 0) & \textit{für } (x_n \ \dots \ x_0) = (0 \ \dots \ 0) \end{cases}$$

Dabei ist $+'$ die aus Definition 6 auf Seite 3 bekannte Addition $+'_{n+1}$ vorzeichenloser Bitvektoren, jetzt aber von $\mathbb{B}^{n+1} \times \mathbb{B}^{n+1}$ nach $\mathbb{B}^{n+2}$ statt von $\mathbb{B}^n \times \mathbb{B}^n$ nach $\mathbb{B}^{n+1}$. Deshalb schneiden wir mit Hilfe der Projektion Π_{n+1} die $(n+2)$-te Stelle ab, um einen Ergebnisvektor aus $\mathbb{B}^{n+1}$ zu bekommen. Dieses dürfen wir tun, weil bei obiger Verwendung gilt, daß diese Ziffer stets gleich 0 ist. □

Beispiel:

Wir arbeiten im $\mathbb{B}^{3+1}$.

$$
\begin{aligned}
K_1(0\ 0\ 0\ 0) &= (1\ 1\ 1\ 1) \rightsquigarrow K_2(0\ 0\ 0\ 0) = (0\ 0\ 0\ 0)\\
K_1(0\ 0\ 0\ 1) &= (1\ 1\ 1\ 0) \rightsquigarrow K_2(0\ 0\ 0\ 1) = (1\ 1\ 1\ 1)\\
K_1(0\ 1\ 1\ 1) &= (1\ 0\ 0\ 0) \rightsquigarrow K_2(0\ 1\ 1\ 1) = (1\ 0\ 0\ 1)\\
K_1(1\ 1\ 1\ 1) &= (0\ 0\ 0\ 0) \rightsquigarrow K_2(1\ 1\ 1\ 1) = (0\ 0\ 0\ 1)\\
K_1(1\ 0\ 0\ 0) &= (0\ 1\ 1\ 1) \rightsquigarrow K_2(1\ 0\ 0\ 0) = (1\ 0\ 0\ 0)\\
K_1(1\ 0\ 0\ 1) &= (0\ 1\ 1\ 0) \rightsquigarrow K_2(1\ 0\ 0\ 1) = (0\ 1\ 1\ 1)
\end{aligned}
$$

Definition 11 (Abstraktionsfunktion $\widetilde{\rho}$)

$$
\widetilde{\rho} : \mathbb{B}^{n+1} \to \mathbb{Z}, \qquad \widetilde{\rho}(x_n\ \ldots\ x_0) = \begin{cases} \rho(x_n\ \ldots\ x_0) = \sum\limits_{i=0}^{n-1} 2^i x_i & \textit{für } x_n = 0 \\ -\rho(K_2(x_n\ \ldots\ x_0)) & \textit{für } x_n = 1 \end{cases}
$$

Die Abstraktionsfunktion $\widetilde{\rho}$ wird auch Retrieve-Funktion genannt. □

Dabei ist ρ jetzt die aus Definition 3 auf Seite 2 bekannte Abstraktionsfunktion $\rho_{n+1} : \mathbb{B}^{n+1} \to \mathbb{N}_0, \quad (x_n\ \ldots\ x_0) \mapsto \sum_{i=0}^{n} 2^i x_i$ für vorzeichenlose Zahlen, allerdings auf $\mathbb{B}^{n+1}$ statt auf $\mathbb{B}^n$.

Beispiel:

Wir arbeiten im $\mathbb{B}^{3+1}$.

$$\widetilde{\rho}(0\ 0\ 0\ 0) = 0 \tag{21}$$

$$\widetilde{\rho}(1\ 0\ 0\ 0) = -\rho(K_2(1\ 0\ 0\ 0)) = -\rho(1\ 0\ 0\ 0) = -\sum_{i=0}^{3} 2^i x_i = -(1 \cdot 2^3) = -8 \tag{22}$$

$$\widetilde{\rho}(1\ 0\ 0\ 1) = -\rho(0\ 1\ 1\ 1) = -7 \tag{23}$$

$$\ldots$$

$$\widetilde{\rho}(1\ 1\ 1\ 1) = -\rho(0\ 0\ 0\ 1) = -1 \tag{24}$$

$$\widetilde{\rho}(0\ 1\ 1\ 1) = 7 \tag{25}$$

Man sieht an den Beispielen (22) und (25), daß bei der Zweierkomplementdarstellung der Betrag der kleinsten darstellbaren Zahl ungleich dem Betrag der größten darstellbaren Zahl ist.

2.3 Addition mit Vorzeichen

Definition 12 (Addition $\widetilde{+}$ auf $\mathbb{B}^{n+1}$)

$$\widetilde{+} : \mathbb{B}^{n+1} \times \mathbb{B}^{n+1} \nrightarrow \mathbb{B}^{n+1}$$

mit Definitionsbereich

$$\underline{dom}\ \widetilde{+} = \Big\{(x,y) \in \mathbb{B}^{n+1} \times \mathbb{B}^{n+1} \ \Big| \ \big(x_n \oplus y_n = 1\big) \vee \big(x_n = 0 \ \wedge \ y_n = 0 \ \wedge \ \rho_{n+2}((x_n \ \dots \ x_0) +' (y_n \ \dots \ y_0)) < 2^n\big) \vee \big(x_n = 1 \ \wedge \ y_n = 1 \ \wedge \ \rho((x_{n-1} \ \dots \ x_0) +' (y_{n-1} \ \dots \ y_0)) \geq 2^n\big)\Big\}$$

und

$$(x_n \ \dots \ x_0) \ \widetilde{+} \ (y_n \ \dots \ y_0) =_{df} \Pi_{n+1}((x_n \ \dots \ x_0) \ +' \ (y_n \ \dots \ y_0))$$

Dabei ist ρ_{n+2} die Abstraktionsfunktion $\rho_{n+2} : \mathbb{B}^{n+2} \to \mathbb{N}_0, \quad (x_{n+1} \ \dots \ x_0) \mapsto \sum_{i=0}^{n+1} 2^i x_i$ für vorzeichenlose Zahlen, jetzt allerdings auf $\mathbb{B}^{n+2}$ statt auf $\mathbb{B}^{n+1}$.

Und $\nrightarrow$ bezeichnet eine partielle Abbildung. □

2.4 Eigenschaften

Lemma 1

$$\widetilde{\rho}(x_n \ \dots \ x_0) = -(2^{n+1} - \underbrace{\rho(x_n \ \dots \ x_0)}_{=\sum_{i=0}^{n} 2^i x_i}) \qquad \text{für } x_n = 1$$

□

Beweis:

Sei $x_n = 1$.

$$\widetilde{\rho}(x_n \ \dots \ x_0) = -\rho(K_2(x_n \ \dots \ x_0)) \tag{26}$$

$$= -\rho\Big(\Pi_{n+1}\big(K_1(x_n \ \dots \ x_0) +' (0 \ \dots \ 0\ 1)\big)\Big) \tag{27}$$

$$= -\rho\Big(\Pi_{n+1}\big(((x_n \oplus 1) \ \dots \ (x_0 \oplus 1)) +' (0 \ \dots \ 0\ 1)\big)\Big) \tag{28}$$

$$= -\Big[\rho\Big((x_n \oplus 1) \ \dots \ (x_0 \oplus 1)\Big) + \rho(0 \ \dots \ 0\ 1)\Big] \qquad [x_n \oplus 1 = 0, \text{ Satz } 1] \tag{29}$$

$$= -\left[\sum_{i=0}^{n} 2^i (x_i \oplus 1) \quad + \quad 1\right] \tag{30}$$

$$= -\left[\sum_{i=0}^{n} 2^i (1 - x_i) \quad + \quad 1\right] \tag{31}$$

$$= -\left[\sum_{i=0}^{n} 2^i \quad - \quad \sum_{i=0}^{n} 2^i x_i \quad + \quad 1\right] \tag{32}$$

$$= -\left[\left(\sum_{i=0}^{n} 2^i \quad + \quad 1\right) \quad - \quad \sum_{i=0}^{n} 2^i x_i\right] \tag{33}$$

$$= -\big[2^{n+1} \quad - \quad \rho(x_n \ \dots \ x_0)\big] \tag{34}$$

□

Satz 2 *Die Abstraktionsfunktion $\widetilde{\rho}$ ist invers zur Repräsentationsfunktion $\widetilde{r}$, d.h.*

$$\widetilde{\rho}(\widetilde{r}(z)) = z \qquad \textit{für } z \in \underline{dom}\,\widetilde{r}$$

$\square$

Beweis:

Fall $\boxed{z \in \{0, \ldots, 2^n - 1\}}$ **:**

Dann gilt mit $\sum_{i=0}^{n-1} 2^i x_i = z$

$$\widetilde{r}(z) = (0\ x_{n-1}\ \ldots\ x_0) \tag{35}$$

und damit

$$\begin{aligned} \widetilde{\rho}(\widetilde{r}(z)) &= \widetilde{\rho}(0\ x_{n-1}\ \ldots\ x_0) && (36)\\ &= \rho(0\ x_{n-1}\ \ldots\ x_0) && (37)\\ &= \sum_{i=0}^{n-1} 2^i x_i && (38)\\ &= z && (39) \end{aligned}$$

Fall $\boxed{z \in \{-2^n, \ldots, -1\}}$ **:**

Dann gilt mit $\sum_{i=0}^{n} 2^i w_i = z + 2^{n+1}$ und $w_n = 1$

$$\widetilde{r}(z) = (1\ w_{n-1}\ \ldots\ x_0) \tag{40}$$

und damit

$$\begin{aligned} \widetilde{\rho}(\widetilde{r}(z)) &= \widetilde{\rho}(1\ w_{n-1}\ \ldots\ w_0) && (41)\\ &= \rho(1\ w_{n-1}\ \ldots\ w_0) - 2^{n+1} && \text{[Lemma 1]} \quad (42)\\ &= \left(\sum_{i=0}^{n} 2^i w_i\right) - 2^{n+1} && (43)\\ &= (z + 2^{n+1}) - 2^{n+1} && (44)\\ &= z && (45) \end{aligned}$$

$\square$

Satz 3 *Das folgende Diagramm* kommutiert*:*

$$\begin{array}{ccc} \mathbb{Z} \times \mathbb{Z} & \xrightarrow{+} & \mathbb{Z} \\ \uparrow{\scriptstyle \widetilde{\rho} \times \widetilde{\rho}} & & \uparrow{\scriptstyle \widetilde{\rho}} \\ \mathbb{B}^{n+1} \times \mathbb{B}^{n+1} & \xrightarrow{\widetilde{+}} & \mathbb{B}^{n+1} \end{array}$$

d.h.

$$\forall n \in \mathbb{N}\,.\forall\big((x_n\ \dots\ x_0),(y_n\ \dots\ y_0)\big) \in \underline{dom}\ \widetilde{+}\ .$$
$$\widetilde{\rho}(x_n\ \dots\ x_0)+\widetilde{\rho}(y_n\ \dots\ y_0)=\widetilde{\rho}((x_n\ \dots\ x_0)\ \widetilde{+}\ (y_n\ \dots\ y_0))$$

□

Beweis:

Sei $x=(x_n\ \dots\ x_0)$ und $y=(y_n\ \dots\ y_0)$.

Fall $\boxed{x_n=y_n=0}$ **und** $\rho_{n+2}((x_n\ \dots\ x_0)+'(y_n\ \dots\ y_0))<2^n$ **:**

Dann gilt

$$\begin{aligned}
\widetilde{\rho}(x_n\ \dots\ x_0)+\widetilde{\rho}(y_n\ \dots\ y_0) &= \rho(x_n\ \dots\ x_0)+\rho(y_n\ \dots\ y_0) && (46)\\
&= \rho_{n+2}((x_n\ \dots\ x_0)+'\rho(y_n\ \dots\ y_0)) \quad \text{[Satz 1]} && (47)\\
&= \rho((x_n\ \dots\ x_0)\,\widetilde{+}\,\rho(y_n\ \dots\ y_0)) \quad [\text{Def. } \widetilde{+},\ x_n=y_n=0] && (48)\\
&= \widetilde{\rho}((x_n\ \dots\ x_0)\,\widetilde{+}\,\rho(y_n\ \dots\ y_0)) && (49)
\end{aligned}$$

Der letzte Schritt gilt, da $x_n=y_n=0$ und da wegen $\rho_{n+2}((x_n\ \dots\ x_0)+'(y_n\ \dots\ y_0))<2^n$ auch kein Übertrag in das n-te Bit entsteht.

Fall $\boxed{x_n=1,y_n=0}$ **:**

Dann gilt

$$\begin{aligned}
\widetilde{\rho}(x)+\widetilde{\rho}(y) &= -2^{n+1}+\rho(x)+\rho(y) && (50)\\
&= -2^{n+1}+\rho_{n+2}(x+'y) && (51)
\end{aligned}$$

Da $x_n=1,y_n=0$ und damit $u_n=x_ny_n+(x_n\oplus y_n)u_{n-1}=u_{n-1}$, gilt

$$\begin{aligned}
x+'y &= (u_n\quad (x_n\oplus y_n\oplus u_{n-1})\quad \dots\quad) && (52)\\
&= (u_{n-1}\quad (1\oplus u_{n-1})\quad (x_{n-1}\oplus y_{n-1}\oplus u_{n-2})\quad \dots\quad) && (53)
\end{aligned}$$

Fall (a) $\boxed{u_{n-1}=0}$ **:**

Dann ist

$$x+'y=(\underbrace{0}_{n+1}\quad 1\quad (x_{n-1}\oplus y_{n-1}\oplus u_{n-2})\quad \dots\quad (x_0\oplus y_0)) \qquad (54)$$

und damit

$$\begin{aligned}
\Pi_{n+1}(x+'y) &= (1\quad (x_{n-1}\oplus y_{n-1}\oplus u_{n-2})\quad \dots\quad (x_0\oplus y_0)) && (55)\\
&= x\,\widetilde{+}\,y && (56)
\end{aligned}$$

Für Fall (a) gilt also:

$$\begin{aligned}
\widetilde{\rho}(x \mathbin{\widetilde{+}} y) &= \widetilde{\rho}(\Pi_{n+1}(x +' y)) && && (57)\\
&= -2^{n+1} + \rho(\Pi_{n+1}(x +' y)) && \text{[Lemma 1]} && (58)\\
&= -2^{n+1} + \rho(x) + \rho(y) && \text{[Satz 1, } u_{n-1} = 0\text{]} && (59)\\
&= -2^{n+1} + \rho(x) + \widetilde{\rho}(y) && && (60)\\
&= \widetilde{\rho}(x) + \widetilde{\rho}(y) && && (61)
\end{aligned}$$

Fall (b) $\boxed{u_{n-1} = 1}$ **:**

Dann ist

$$x +' y = (\underbrace{1}_{n+1} \quad 0 \quad (x_{n-1} \oplus y_{n-1} \oplus u_{n-2}) \quad \ldots \quad (x_0 \oplus y_0)) \tag{62}$$

Also folgt

$$x \mathbin{\widetilde{+}} y = (0 \quad (x_{n-1} \oplus y_{n-1} \oplus u_{n-2}) \quad \ldots \quad (x_0 \oplus y_0)) \tag{63}$$

und damit

$$\begin{aligned}
\widetilde{\rho}(x \mathbin{\widetilde{+}} y) &= \rho(x \mathbin{\widetilde{+}} y) && && (64)\\
&= \rho_{n+2}(x +' y) - 2^{n+1} && \text{[Korrektur des hinzugefügten 1-Bits } n+1\text{]} && (65)\\
&= \rho(x) + \rho(y) - 2^{n+1} && \text{[Satz 1]} && (66)\\
&= \rho(x) + \widetilde{\rho}(y) - 2^{n+1} && && (67)\\
&= \widetilde{\rho}(x) + \widetilde{\rho}(y) && \text{[Lemma 1]} && (68)
\end{aligned}$$

Fall $\boxed{x_n = 0, y_n = 1}$ **:**

Aus Symmetriegründen wie $x_n = 1, y_n = 0$.

Fall $\boxed{x_n = 1, y_n = 1}$ **:**

Dann ist

$$\begin{aligned}
x +' y &= (u_n \quad (x_n \oplus y_n \oplus u_{n-1}) \quad \ldots \quad (x_0 \oplus y_0)) && (69)\\
&= (\underbrace{(\underbrace{x_n y_n}_{=1} + \underbrace{(x_n \oplus y_n)}_{=0} u_{n-1})}_{=1} \quad \underbrace{(\underbrace{x_n \oplus y_n}_{=0} \oplus u_{n-1})}_{=u_{n-1}} \quad \ldots \quad (x_0 \oplus y_0)) && (70)\\
&= (\underbrace{1}_{n+1} \quad \underbrace{u_{n-1}}_{n} \quad (x_{n-1} \oplus y_{n-1} \oplus u_{n-2}) \quad \ldots \quad (x_0 \oplus y_0)) && (71)\\
&= (1 \quad 1 \quad (x_{n-1} \oplus y_{n-1} \oplus u_{n-2}) \quad \ldots \quad (x_0 \oplus y_0)) \qquad \text{[wegen Def. } \underline{\text{dom}}\ \widetilde{+}\text{, 3. Fall]} && (72)
\end{aligned}$$

Also folgt

$$x \mathbin{\widetilde{+}} y = (1 \quad (x_{n-1} \oplus y_{n-1} \oplus u_{n-2}) \quad \dots \quad (x_0 \oplus y_0)) \tag{73}$$

und damit

$$\begin{aligned}
\widetilde{\rho}(x \mathbin{\widetilde{+}} y) &= -2^{n+1} + \rho(x \mathbin{\widetilde{+}} y) && \text{[Lemma 1]} && (74)\\
&= -2^{n+1} + \rho_{n+2}(x +' y) - 2^{n+1} && \text{[Korrektur des hinzugefügten 1-Bits } n+1\text{]} && (75)\\
&= -2^{n+1} + \rho(x) + \rho(y) - 2^{n+1} && \text{[Satz 1]} && (76)\\
&= \widetilde{\rho}(x) + \widetilde{\rho}(y) && \text{[Lemma 1]} && (77)
\end{aligned}$$

□