IEEE-754-2008

Von Janette Kaspar und Alexander Linz

Die Entstehung von IEEE 754

- Die IEEE Gesellschaft
- Geschichte von IEEE bis 1985
- Erschaffung von IEEE-754-2008
  - Motivation
  - Hauptziele

Die IEEE Gesellschaft

- Institute of Electrical and Electronics Engineers
- Bereiche: Elektrotechnik und Informationstechnik
- Gegründet: 01.01.1963
  - Zusammenschluss der AIEE und IRE
  - Logo zeigt Korkenzieherregel und eine Raute

Geschichte von IEEE-754

- 1976:
  - Intel entwirft einen Floating Point Co-Prozessor
  - Verwendet für i8086/8 und i432 Microprocessor
- Dr. John Palmer wird zum Manager ernannt
  - Er rekrutiert Dr. William Kahan
  - Zusammen erarbeiten sie die Grundzüge des Standards
- Silicon Valley:
  - IT Firmen treffen sich und besprechen das Problem Floating Point
  - Daraus geht der IEEE p754 hervor → Vorläufer zu IEEE 754

Geschichte von IEEE-754

- Der Standard IEEE-754-1985 wurde 1985 festgesetzt und veröffentlicht
- Sehr erfolgreich
  - Wurde in den meisten Prozessoren und Programmiersprachen verwendet
- Doch mit wachsendem Markt kamen Probleme:
- Der IEEE-754-1985 Standard musste überarbeitet werden

Erschaffung von IEEE-754-2008

Motivation

- IEEE-754-1985 musste erneuert worden, als die Computer immer besser wurden
- Die Version von 1985
  - Einige Fehler und ließ Mehrdeutigkeit zu
  - Keine Dezimalen Formate
  - Spezialfälle nicht vorhanden

Erschaffung von IEEE-754-2008

Hauptziele

- Zusammenführung von IEEE-754-1985 und IEEE-854
- Entfernung von Mehrdeutigkeit und Fehlern
- Neue Genauigkeiten: 16 und 128 Bit neben 32 und 62 Bit
- Spezialfälle: ±0 und ±∞

Was wird durch IEEE-754-2008 definiert?

- Arithmetische Formate
  - Endliche Menge an Zahlen
  - $\pm\infty$
  - NaN (Not a Number)
- Rundungsregeln
- Rechenoperationen
- Exception Handling

Arithmetische Formate

Genauer definiert durch:

- Eine Basis b, entweder 2 oder 10
- Die Präzision p
- Der Exponentenbereich (*emin* bis *emax*)

Und besteht aus:

- s = das Vorzeichen (sign)
- c = der Koeffizient
- q = der Exponent
- Numerische Darstellung:

$$(-1)^s \times c \times b^q$$

- $\pm\infty$
- quiet NaN und signaling NaN

Arithmetische Formate

Genauer definiert durch:

- Eine Basis b, entweder 2 oder 10
- Die Präzision p
- Der Exponentenbereich (*emin* bis *emax*)

Und besteht aus:

- s = das Vorzeichen (sign)
- c = der Koeffizient
- q = der Exponent
- Numerische Darstellung:

$$(-1)^s \times c \times b^q$$

- $\pm\infty$
- quiet NaN und signaling NaN

Rundungsregeln

- Runde zum nächsten, bevorzuge gerade
- Runde zum nächsten, bevorzuge weiter weg von 0

- Runde in Richtung 0
- Runde in Richtung $+\infty$
- Runde in Richtung $-\infty$

Rundungsregeln

- Runde zum nächsten, bevorzuge gerade
- Runde zum nächsten, bevorzuge weiter weg von 0

- Runde in Richtung 0
- Runde in Richtung $+\infty$
- Runde in Richtung $-\infty$

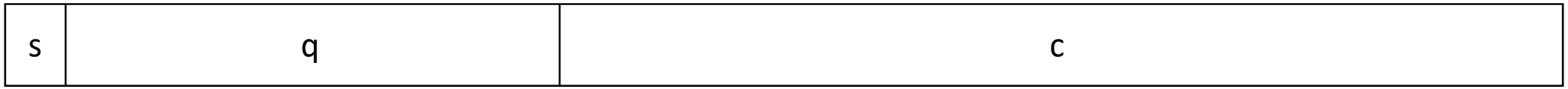
Rechenoperationen

- Arithmetische Operation
  - (Addition, Subtraktion, Multiplikation, Division, Potenz, Rest, …)
- Konvertierungen
  - (Zu/Von Strings und zwischen Formaten)
- Vorzeichen Manipulation
  - (Betrag, Negation, …)
- Vergleichen und totale Ordnung
- NaN Tests

Exception Handling

1. Invalid Operation -> qNaN
2. Division durch 0 -> ±∞
3. Overflow -> ±∞ oder gerundeter Wert
4. Underflow -> 0 oder subnormaler Wert
5. Nicht exakter Wert -> Gerundeter Wert

Binary32 – Ein Beispiel



s	Vorzeichen	1 bit
q	Exponent	8 bit
c	Koeffizient	23 bit

Formel zum Umwandeln:

$$(-1)^s \times 2^{q-127} \times (1.c_2)_{10}$$

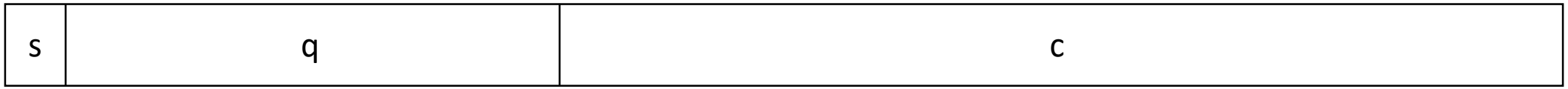
48

0 10000100 10000000000000000000000

0_2 10000100_2 $(1.)10000000000000000000000_2$

0_{10} 132_{10} 1.5_{10}

Binary32 – Ein Beispiel



s	Vorzeichen	1 bit
q	Exponent	8 bit
c	Koeffizient	23 bit

Formel zum Umwandeln:

$$(-1)^s \times 2^{q-127} \times (1.c_2)_{10}$$

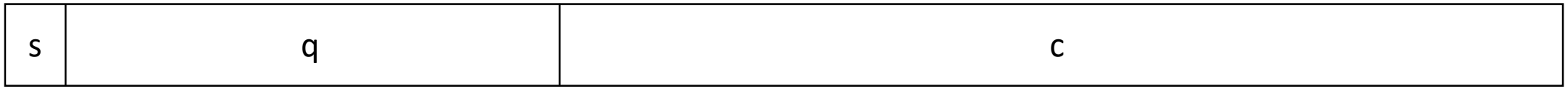
48

0 10000100 10000000000000000000000

0_2 10000100_2 $(1.)10000000000000000000000_2$

0_{10} 132_{10} 1.5_{10}

Binary32 – Ein Beispiel



s	Vorzeichen	1 bit
q	Exponent	8 bit
c	Koeffizient	23 bit

Formel zum Umwandeln:

$$(-1)^s \times 2^{q-127} \times (1.c_2)_{10}$$

48

0 10000100 10000000000000000000000

0_2	10000100_2	$(1.)10000000000000000000000_2$
0_{10}	132_{10}	1.5_{10}

Binary32 – Ein Beispiel

	48	
0_2	10000100_2	$(1.)10000000000000000000000_2$
0_{10}	132_{10}	1.5_{10}

$$(-1)^s \times 2^{q-127} \times (1.c_2)_{10}$$

$0^0 \times 2^{132-127} \times 1.5$

$1 \times 2^5 \times 1.5$

32×1.5

48

Binary32 – Ein Beispiel

48		
0_2	10000100_2	$(1.)10000000000000000000000_2$
0_{10}	132_{10}	1.5_{10}

$$(-1)^s \times 2^{q-127} \times (1.c_2)_{10}$$

$0^0 \times 2^{132-127} \times 1.5$

$1 \times 2^5 \times 1.5$

32×1.5

48

Binary32 – Ein Beispiel

	48	
0_2	10000100_2	$(1.)10000000000000000000000_2$
0_{10}	132_{10}	1.5_{10}

$$(-1)^s \times 2^{q-127} \times (1.c_2)_{10}$$

$0^0 \times 2^{132-127} \times 1.5$

$1 \times 2^5 \times 1.5$

32×1.5

48

Binary32 – Ein Beispiel

	48	
0_2	10000100_2	$(1.)10000000000000000000000_2$
0_{10}	132_{10}	1.5_{10}

$$(-1)^s \times 2^{q-127} \times (1.c_2)_{10}$$

$0^0 \times 2^{132-127} \times 1.5$

$1 \times 2^5 \times 1.5$

32×1.5

48

Binary32 – Ein Beispiel

	48	
0_2	10000100_2	$(1.)10000000000000000000000_2$
0_{10}	132_{10}	1.5_{10}

$$(-1)^s \times 2^{q-127} \times (1.c_2)_{10}$$

$0^0 \times 2^{132-127} \times 1.5$

$1 \times 2^5 \times 1.5$

32×1.5

48

Quellen

- IEEE Xplore Digital Library
  - https://ieeexplore.ieee.org/document/4610935/
- IBM, Decimal Arithmetic Encodings, Mike COWLISHAW, 2009
  - https://speleotrove.com/decimal/decbits.pdf/