Moritz Hahn

1	2	3	4	Σ

Übungsblatt Nr. 1
(Abgabetermin 19.10.2024)

Aufgabe 2

(a)

Binärdarstellung von 93:

$$93 \div 2 = 46 \text{ Rest } 1 \quad \text{(niederwertigstes Bit)}$$
$$46 \div 2 = 23 \text{ Rest } 0$$
$$23 \div 2 = 11 \text{ Rest } 1$$
$$11 \div 2 = 5 \text{ Rest } 1$$
$$5 \div 2 = 2 \text{ Rest } 1$$
$$2 \div 2 = 1 \text{ Rest } 0$$
$$1 \div 2 = 0 \text{ Rest } 1$$

Wenn man die Reste von unten nach oben liest, erhält man:

$$93_{10} = 1011101_2$$

Binärdarstellung von 162:

$$162 \div 2 = 81 \text{ Rest } 0$$
$$81 \div 2 = 40 \text{ Rest } 1$$
$$40 \div 2 = 20 \text{ Rest } 0$$
$$20 \div 2 = 10 \text{ Rest } 0$$
$$10 \div 2 = 5 \text{ Rest } 0$$
$$5 \div 2 = 2 \text{ Rest } 1$$
$$2 \div 2 = 1 \text{ Rest } 0$$
$$1 \div 2 = 0 \text{ Rest } 1$$

Die Binärdarstellung von 162 ist also:

$$162_{10} = 10100010_2$$

(b)

Hexadezimaldarstellung von 93 (1011101_2):

- Zunächst ergänzen wir auf volle 4er-Blöcke, indem wir vorne Nullen hinzufügen: 0101 1101
- $0101_2 = 5_{16}$
- $1101_2 = D_{16}$

Damit ist:

$$93_{10} = 5D_{16}$$

Hexadezimaldarstellung von 162 (10100010_2):

- Die 4er-Blöcke sind bereits vollständig: 1010 0010
- $1010_2 = A_{16}$
- $0010_2 = 2_{16}$

Damit ist:

$$162_{10} = A2_{16}$$

(c)

Allgemeine Beobachtung: Wenn $a + b = 2^n - 1$, dann kann man feststellen, dass die Binärdarstellungen von a und b so beschaffen sind, dass sie sich in keinem Bit "überschneiden", also in keinem Bit beide eine 1 haben. In der Summe ergibt sich dann für jedes Bit a_i und b_i, dass $a_i + b_i = 1$. Das bedeutet, dass die Binärdarstellungen von a und b genau komplementär zueinander sind, also dort, wo a eine 1 hat, hat b eine 0 und umgekehrt.

Beispiel mit $n = 8$, $a = 93$ und $b = 162$:

$$a = 1011101_2$$

$$b = 10100010_2$$

Die Summe $a + b$ ist:

$$1011101_2 + 10100010_2 = 11111111_2 = 255_{10} = 2^8 - 1$$

Es gilt also, dass a und b in ihren Binärdarstellungen komplementär zueinander sind.

Aufgabe 2

(a)

$$\begin{array}{cccccc} / & 1 & 0 & 0 & 1 & 1 \\ + & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{array}$$

Das Ergebnis der Addition im Binärsystem ist also:

$$19_2 + 14_2 = 100001_2$$

Im Dezimalsystem ergibt dies:

$$100001_2 = 33_{10}$$

(b) Schriftliche Multiplikation von 19 und 14

Dezimalsystem:

$$19_{10} \times 14_{10} = 266_{10}$$

Binärsystem:

$$19_{10} = [10011]_2$$

$$14_{10} = [1110]_2 = [01110]_2$$

Nun führen wir die schriftliche Multiplikation im Binärsystem durch:

$$\begin{array}{r}
10011_2 \\
\times\ 01110_2 \\
\hline
00000 \quad \text{(10011 mal 0)} \\
10011 \quad \text{(10011 mal 1, um 1 Stelle nach links verschoben)} \\
10011 \quad \text{(10011 mal 1, um 2 Stellen nach links verschoben)} \\
10011 \quad \text{(10011 mal 1, um 3 Stellen nach links verschoben)} \\
\hline
100001110 \quad \text{(Ergebnis der Addition der Teilergebnisse)}
\end{array}$$

Das Ergebnis der Multiplikation im Binärsystem ist also:

$$10011_2 \times 1110_2 = 100001110_2$$

Im Dezimalsystem ergibt dies:

$$100001110_2 = 266_{10}$$

Aufgabe 3

(a)

Die Binärdarstellung lautet $[1011wxyz]_2$, wobei w, x, y und z beliebige Binärziffern (0 oder 1) sind. Das bedeutet, dass die ersten vier Stellen festgelegt sind: 1011 (was 11 im Dezimalsystem entspricht) und die letzten vier Stellen beliebig sein können. Damit gibt es $2^4 = 16$ mögliche Kombinationen für $wxyz$.

Für jede dieser Kombinationen können wir die Zahl berechnen:

$$[1011wxyz]_2 = 11 \times 16 + (wxyz)_2$$

Damit liegen die Dezimalwerte zwischen:

$$11 \times 16 + 0 = 176 \quad \text{und} \quad 11 \times 16 + 15 = 191$$

Also sind alle Dezimalzahlen zwischen 176 und 191.

(b)

Hier sind die vier Ziffern xy beliebig. Die festgelegten Ziffern sind 101000, was 40 im Dezimalsystem entspricht. Die ersten beiden Stellen xy können wieder beliebig sein, was $2^2 = 4$ mögliche Kombinationen ergibt.

Für jede dieser Kombinationen können wir die Zahl berechnen:

$$[xy101000]_2 = (xy)_2 \times 64 + 40$$

Die möglichen Werte für $(xy)_2$ sind $00, 01, 10, 11$, also:

$$0 \times 64 + 40 = 40$$

$$1 \times 64 + 40 = 104$$

$$2 \times 64 + 40 = 168$$

$$3 \times 64 + 40 = 232$$

Also sind die Dezimalzahlen 40, 104, 168 und 232.

(c)

Wenn in der Binärdarstellung kein 01 vorkommen darf, dann müssen die Zahlen ausschließlich aus Folgen von Einsen oder Nullen bestehen. Die zulässigen Muster sind also $00000000, 11111111, 11111110, 11111100, \dots$ und umgekehrt mit Nullen.

Die Zahlen, die dieses Muster erfüllen, sind:

$$0, 1, 3, 7, 15, 31, 63, 127, 255$$

(d)

Die Zahl $[11110111]_2$ entspricht 247_{10}. Nun suchen wir Zahlen, die sich in genau einer Stelle von dieser Binärzahl unterscheiden. Das bedeutet, dass wir eine der 8 Stellen von $[11110111]_2$ umdrehen können, um die gesuchten Zahlen zu erhalten.

$$11110111 \quad (247)$$

$$01110111 = 119, \quad 10110111 = 183, \quad 11010111 = 215$$

$$11100111 = 231, \quad 11111111 = 255, \quad 11110011 = 243, \quad 11110101 = 245$$

Die gesuchten Zahlen sind also: $119, 183, 215, 231, 255, 243, 245$.

(e)

Die Binärdarstellung $[v10wx1yz]_2$ hat 5 feste Bits (10 und 1 sind festgelegt), und die restlichen Bits v, w, x, y, z sind frei wählbar. Das ergibt $2^5 = 32$ mögliche Kombinationen.

Also gibt es 32 verschiedene Dezimalzahlen.

(f)

Die Zahl $[11110111]_2$ entspricht 247_{10}. Um zwei Stellen zu ändern, wählen wir zwei der 8 möglichen Stellen aus, die wir umdrehen können. Die Anzahl der möglichen Kombinationen, zwei Stellen auszuwählen, ist:

$$\binom{8}{2} = \frac{8 \times 7}{2} = 28$$

Also gibt es 28 verschiedene Dezimalzahlen.

4

(a)

Eine Subnetzmaske besteht aus einer Folge von Einsen gefolgt von Nullen in der binären Darstellung. Entscheiden wir, welche der beiden Zahlenfolgen erlaubte Subnetzmasken sind:

- **i.** 255.255.200.0:
  Die dezimale Darstellung lautet:

  $$255 = 11111111_2, \quad 255 = 11111111_2, \quad 200 = 11001000_2, \quad 0 = 00000000_2$$

  Diese Zahlenfolge ist keine gültige Subnetzmaske, da die Einsen in der binären Darstellung von 200 nicht fortlaufend sind (es gibt eine Null zwischen den Einsen).

- **ii.** 252.0.0.0:
  Die dezimale Darstellung lautet:

  $$252 = 11111100_2, \quad 0 = 00000000_2, \quad 0 = 00000000_2, \quad 0 = 00000000_2$$

  Diese Zahlenfolge ist eine gültige Subnetzmaske, da die Einsen fortlaufend sind, gefolgt von Nullen.

(b)

Die dezimalen Zahlen in einer IPv4-Subnetzmaske sind die jeweiligen Werte der 8-Bit-Blöcke in der binären Darstellung. Da jeder Block 8 Bits lang ist, können die dezimalen Werte zwischen 0 und 255 liegen. Diese Zahlen müssen jedoch fortlaufende Einsen und dann Nullen darstellen, was bedeutet, dass nur bestimmte Werte zulässig sind. Diese Werte sind:

$$0, 128, 192, 224, 240, 248, 252, 254, 255$$

(c)

Die Subnetzmaske 255.255.224.0 hat folgende binäre Darstellung:

$$255 = 11111111_2, \quad 255 = 11111111_2, \quad 224 = 11100000_2, \quad 0 = 00000000_2$$

Also lautet die vollständige binäre Darstellung:

$$11111111.11111111.11100000.00000000$$

Nun überprüfen wir, ob sich die beiden IP-Adressen jeweils im selben Subnetz befinden können, indem wir die IP-Adressen mit der Netzmaske logisch UND verknüpfen.

i. IP-Adressen: 10.10.138.10 und 10.10.158.248

Die binäre Darstellung der beiden IP-Adressen lautet:

$$10.10.138.10 = 00001010.00001010.10001010.00001010$$

$$10.10.158.248 = 00001010.00001010.10011110.11111000$$

Nun wenden wir die Subnetzmaske an:

$$00001010.00001010.10001010.00001010 \wedge 11111111.11111111.11100000.00000000 =$$

$$00001010.00001010.10000000.00000000$$

$$00001010.00001010.10011110.11111000 \wedge 11111111.11111111.11100000.00000000 =$$

$$00001010.00001010.10000000.00000000$$

Da beide IP-Adressen nach der Anwendung der Subnetzmaske denselben Wert haben, befinden sie sich im selben Subnetz.

ii. IP-Adressen: 120.210.68.0 und 120.210.58.0

Die binäre Darstellung der beiden IP-Adressen lautet:

$$120.210.68.0 = 01111000.11010010.01000100.00000000$$

$$120.210.58.0 = 01111000.11010010.00111010.00000000$$

Nun wenden wir die Subnetzmaske an:

$$01111000.11010010.01000100.00000000 \wedge 11111111.11111111.11100000.00000000 =$$

$$01111000.11010010.01000000.00000000$$

$$01111000.11010010.00111010.00000000 \wedge 11111111.11111111.11100000.00000000 =$$

$$01111000.11010010.00100000.00000000$$

Da die beiden IP-Adressen nach der Anwendung der Subnetzmaske unterschiedliche Werte haben, befinden sie sich nicht im selben Subnetz.

(d)

Eine IPv4-Adresse hat 32 Bit, was bedeutet, dass es insgesamt 2^{32} mögliche IPv4-Adressen gibt:

$$2^{32} = 4,294,967,296$$

also ist es nicht möglich, jedem Smartphone eine eindeutige IPv4-Adresse zuzuweisen.

Eine IPv6-Adresse besteht aus 128 Bit. Die Anzahl der möglichen IPv6-Adressen ist daher:

$$2^{128} = 3.4 \times 10^{38}$$

Wenn wir diese Anzahl durch 7 Milliarden teilen, erhalten wir die Anzahl der IPv6-Adressen pro Smartphone:

$$\frac{2^{128}}{7 \times 10^9} \approx 4.9 \times 10^{28}$$

Also könnte theoretisch jedes Smartphone etwa 4.9×10^{28} IPv6-Adressen erhalten.