Rechnerstrukturen: Übungsblatt 4

Florian Ludewig (185722)

5. Juni 2020

Aufgabe 3

```
1  int odd_ones(unsigned x)
2  {
3    for (int i = 16; i >= 1; i /= 2)
4    {
5      int shifted = x >> i;
6      x = x ^ shifted;
7    }
8    return x << 31 >> 31;
9  }
```

Die Anfangs 32-stellige Bitsequencz wird jede Iteration der Schleife in zwei gleich große Abschnitte geteilt. Diese Abschnitte werden dann mit einem bitweisen `XOR` verglichen. Wenn `x` ursprünglich eine ungerade Anzahl an Einsen hatte, hat auch das neue Zwischenergebnis, welches nur noch halb so groß ist, wieder eine ungerade Anzahl an Einsen (gilt ebenfalls für gerade Anzahl an Einsen). In der letzten Iteration ist somit das einzige Bit welches übrig bleibt entweder `1` (ungerade Anzahl an Einsen) oder `0` (gerade Anzahl an Einsen).

Aufgabe 4 auf Seite 2

Aufgabe 4

```c
#include <stdio.h>

unsigned short compare(unsigned short x, unsigned short mask)
{
  return (x & mask) == mask;
}

int rows(unsigned short x)
{
  int top = 0b0000000111000000;
  int middle = 0b0000000000111000;
  int bottom = 0b0000000000000111;

  return (compare(x, top) || compare(x, middle) || compare(x, bottom));
}

int columns(unsigned short x)
{
  int left = 0b0000000001001001;
  int middle = 0b0000000010010010;
  int right = 0b0000000100100100;

  return (compare(x, left) || compare(x, middle) || compare(x, right));
}

int diagonals(unsigned short x)
{
  int bottom_left_to_top_right = 0b0000000100010001;
  int bottom_right_to_top_left = 0b0000000100010100;

  return (compare(x, bottom_left_to_top_right) || compare(x,
      bottom_right_to_top_left));
}

unsigned short winner_result(unsigned short x)
{
  int one_is_winner = (columns(x) || diagonals(x) || rows(x));
  int zero_is_winner = (columns(~x) || diagonals(~x) || rows(~x));
  if (one_is_winner && zero_is_winner)
    return (x & 0b0111111111111111);
  if (one_is_winner || zero_is_winner)
    return (x | 0b1000000000000000);
  return (x & 0b0111111111111111);
}
```

Die Funktion `winner_result` ist die Funktion die das nach Aufgabenstellung gewünschte Resultat zurückgibt.