

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 Bitsequenz 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

```
1  // TODO
```