

Benke Hargitai 5370932
Lukas Seyfried 5343019

1	2	3	Σ
1.5/7	6.75/7	6/6	14.25/20

Aufgabenblatt 04

Abgabe: 18.11.2022

Aufgabe 1

Prozessorzeit:

$z = 1000$ Taktzyklen

$f = 8 \text{ MHz} = 8 \cdot 10^6 \text{ 1/s}$

$$T_p = T_{\text{prozessor}} = z/f = \frac{1000}{8 \cdot 10^6} \text{ s} = 1.25 \cdot 10^{-4} \text{ s}$$

solange braucht der Prozessor für den Overhead

Festplattezeit:

$r = 8 \text{ MB/s} = 8 \cdot 2^{20} \text{ Byte/s}$ (Datenübertragungsrate)

$w = 32 \text{ bit} = 4 \text{ Byte}$ und $8 \cdot 4 \text{ B} = 28 \text{ B}$ (Wortlänge)

$$T_{\text{raw}} = w/r = \frac{8}{8 \cdot 2^{20}} \text{ s} = 9.5 \cdot 10^{-7} \text{ s}$$

die MHz sind schon Mega und nicht Mibi. Das ist nur bei Bytes so, dass man meist Mibi meint.

$$T_f = T_{\text{festplatte}} = \frac{T_{\text{raw}}}{5\%} = \frac{9.5 \cdot 10^{-7}}{0.05} = 1.9 \cdot 10^{-5} \text{ s}$$

Warum geteilt? -0.5
-1 Aber da ist die CPU ja garnicht miteinbezogen.

Die 5% sind auf die CPU bezogen, man muss die CPU, die schneller als die Festplatte ist noch gegen diese matchen

Relativer Anteil:

$$\frac{T_f}{T_p} = \frac{1.9 \cdot 10^{-5}}{1.25 \cdot 10^{-4}} = 0.152 = 15.2\%$$

Folgefehler: falsche Zeiten in Relation zueinander gesetzt

leider nein $> < 1.5 / 3$ für teilweise richtige Schlussfolgerungen im Rechenweg

Aufgabe 2

was ist mit \reset? -0.25

Jedes mal, wenn der Interrupt-Controller über das Signal **INT** einen Interupt meldet, zählen wir mit dem **up** hoch, und wenn der Prozessor die Abarbeitung eines Interupts mittels **/INTA** signalisiert, zählen wir mit **down** runter. Somit ist der Zähler also genau dann 0, wenn keine ISR auf dem Prozessor läuft.

Diese Methode reicht nicht aus, um den Interrupt-Controller korrekt zu implementieren, weil die Priorität der aktuell laufenden ISR nicht ermittelt werden kann. Wenn der Prozessor also die Abarbeitung einer ISR meldet und dann einen unterbrochenen Interrupt fortsetzt,

dann kann uns der Zähler nur verraten, dass gerade eine ISR im Prozessor läuft, aber nicht mit welcher Priorität und somit kann der Interrupt-Controller nicht bestimmen, ob ein neuer Interrupt signalisiert werden darf.

also in
einem
Stack *

Als Lösung speichern wir die Prioritäten der gemeldeten Interrupts in der ersten freien Speicherzelle ab. Der Zähler aus Aufgabe a) gibt hierbei die Adresse der ersten freien Speicherzelle an.

Aufgabe 3

