

	1	2	3	Σ
Benke Hargitai 5370932				
Lukas Seyfried 5343019				

Aufgabenblatt 04

Abgabe: 18.11.2022

Aufgabe 1

Prozessorzeit:

$$z = 1000 \text{ Taktzyklen}$$

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

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

Festplattezeit:

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

$$w = 32 \text{ bit} = 8 \text{ Byte} \quad (\text{Wortlänge})$$

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

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

Relativer Anteil:

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

Aufgabe 2

Jedes mal, wenn der Interrupt-Controller über das Signal **INT** einen Interrupt meldet, zählen wir mit dem **up** hoch, und wenn der Prozessor die Abarbeitung eines Interrupts 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.

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

