

Laborübung 4 Rechnerarchitektur

Hinweise: Schreiben Sie die Lösungen, so weit es möglich ist, auf die Aufgabenblätter. Tragen Sie Namen, Matrikelnummer und Studiengang in die nachfolgende Tabelle ein und schreiben Sie auf jedes zusätzlich abgegebene Blatt ihre Matrikelnummer. Lassen Sie vorgeführte Experimente vom Betreuer gegenzeichnen. Für eine Bescheinigung der erfolgreichen Teilnahme sind in jeder bis auf einer Laborübung mindestens 60% der Punkte zu erreichen.

Name	Matrikelnummer	Studiengang	Punkte von 20	≥ 40%

Aufgabe 1:

Übersetzen Sie das nachfolgende Programm zur Division von zwei 8-Bit-Zahlen mit -O1 und untersuchen Sie das disassemblierte Programm:

```

uint8_t q, a=0xB3, b=0x7;
int main(void){
    q=a/b;
}
    
```

- a) In welchen Registern werden Divident und Divisor übergeben. Welches Register ist der Schleifenzähler und in welchem Register erfolgt die Rückgabe des Quotienten? 2P

	Schleifenzähler	a (Divident)	b (Divisor)	q (Quotient)
Register				

- b) Tragen Sie in den Tabellenkopf die Registernamen in der Reihenfolge der Tabelle zuvor und in die weiteren Spalten zusätzlich genutzte Register ein. Setzen Sie einen Unterbrechungspunkt vor dem bedingten Sprung am Schleifenende. Arbeiten Sie das Programm im Schrittbetrieb ab und tragen Sie für jeden Schleifendurchlauf die Registerwerte in die Tabelle ein. 5P

vor				
U1				
U2				
U3				
U4				
U5				
U6				
U7				
U8				
U9				

- c) Lesen Sie den berechneten Quotienten und den Rest nach dem Rücksprung ab und kontrollieren Sie den Wert mit dem Taschenrechner: 1P

Quotient	Rest	Ergebnis Taschenrechner

- d) Beschreiben Sie den Algorithmus und versuchen Sie zu zeigen, dass es sich bei diesem Algorithmus aus bedingter Subtraktion und Verschiebung tatsächlich um eine Division handelt¹. 2P

Aufgabe 2:

Gesucht ist ein korrekt arbeitendes C-Programm zur Berechnung von

$$I_D = K \cdot (U_{GS})^2$$

($K = 0,375$ – Konstante) mit Festkommaformaten für die Variablen:

	Bitanzahl	Nachkommabits	Wertebereich
I_D, U_{GS}, U_{GS}^2	16	8	-0x80 bis 0x7F,FF
K	16	14	-2 bis 0x1,FFFE

Das nachfolgende Beispiel enthält zusätzlich die korrekte Berechnung mit Gleitkommazahlen, aber bei den Festkommaberechnung wurde die Kommaverschiebungen gegenüber ganzzahligen Multiplikationen vergessen.

¹Heben Sie sich diese Teilaufgabe bis zum Schluss auf. Die ist anspruchsvoll! Sie müssen darauf achten, welche Operation vor den Rotationen jeweils das Carry-Bit setzt oder löscht und was das Carry-Bit dann repräsentiert.

```

11 float float_K = 0.375;
12 float float_ID, float_UGS=7.83;
13 int16_t fix_K, fix_ID, fix_UGS;
14
15 int main(void){
16     fix_K     = (int16_t)(float_K * (1<<14));
17     fix_UGS   = (int16_t)(float_UGS * (1<< 8));
18     float_UGS *= float_UGS;
19     fix_UGS   = fix_UGS * fix_UGS;
20     float_ID  = float_K * float_UGS;
21     fix_ID    = fix_K * fix_UGS;
22 }

```

a) Bereiten Sie folgende Testbeispiele vor²: 6P

Sollwertwert			Sollwert hex. mit 8 Nachkommabits		
U_{GS}	U_{GS}^2	I_D	U_{GS}	U_{GS}^2	I_D
1			0x100		
2,5					
7,83					

- b) Für welchen Wertebereich von U_{GS} kann das Programm mit dem gegebenen Wert für K maximal funktionieren, ohne dass der Wertebereich für I_D überläuft? 1P
- c) Korrigieren Sie die Berechnungen der Festkommawerte in den Zeilen 19 und 21 so, dass richtige Ergebnisse herauskommen. 3P

Hinweis: Sie müssen jeweils 32-Bit-Produkte bilden und daraus die richtigen 16 Bit auswählen. Das erfordert eine zusätzliche Variable vom Typ `int32_t` und die Konvertierung eines Faktors vor jeder der beiden Multiplikation.

²Sollwerte sind die Werte die herauskommen müssten, nicht die, die das Programm berechnet.