

Technische Universität
 Clausthal Institut für Informatik
 Prof. G. Kemnitz

9. Januar 2017

Rechnerarchitektur: Aufgabenblatt 1

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.

Name	Matrikelnummer	Studiengang	Punkte von 17	≥ 40% (7P)

Aufgabe 1.1:

- a) Wie viele Bits werden zur Unterscheidung von 20 Befehlen benötigt? 1P
- b) Wie kann ein Prozessor mit einem 32 Bit-Befehlszähler einen Hauptspeicher größer 4 GByte (> 2³²Bytes) adressieren? 1P
- c) Wie viele Befehle benötigt ein 8 Bit-RISC-Prozessor mindestens, um zwei in sein Register geladene 32 Bit-Zahlen zu addieren? 1P
- d) Warum kann ein 32 Bit-RISC-Prozessor nicht mit einem Befehl eine 32 Bit-Konstante laden? 1P

Aufgabe 1.2: Ergänzen Sie in den nachfolgenden Rechnungen die fehlenden Binär- und Hexadezimalziffern. 8P

$\begin{array}{r} a \ \square\square\square\square \ 0110 \ (0x\square\square) \\ b \ 1101 \ 1010 \ (0x\square\square) \\ \hline a \wedge b \ 1100 \ \square\square\square\square \ (0x\square\square) \end{array}$	$\begin{array}{r} a \ 1\square01 \ 0011 \ (0x\square\square) \\ b \ \square100 \ 0110 \ (0x\square\square) \\ \hline a \vee b \ 11\square\square\square\square \ (0x\square\square) \end{array}$
$\begin{array}{r} a \ 1001 \ 1110 \ (0x\square\square) \\ b \ \square\square\square\square \ 0011 \ (0x\square\square) \\ \hline a \oplus b \ 0001 \ \square\square\square\square \ (0x\square\square) \end{array}$	$\begin{array}{r} b \ \square\square\square\square \ 0001 \ (0x\square\square) \\ \hline \bar{b} \ 0110 \ \square\square\square\square \ (0x\square\square) \end{array}$

Aufgabe 1.3: Ergänzen Sie in dem nachfolgenden Ausschnitt aus einem C-Programm die den Variablen zugewiesenen Werte binär und hexadezimal. 5P

```

uint8_t a=0xA1, b=0xCB, c = 0x15, d, e, f, g;
...
d = a & b; // d= 0b . . . : . . . = 0x .
e = d | c; // e= 0b . . . : . . . = 0x .
f = e >> 3; // f= 0b . . . : . . . = 0x .
g = f << 5; // g= 0b . . . : . . . = 0x .
h = f | g; // h= 0b . . . : . . . = 0x .
    
```